A Neurocontrol Paradigm for Intelligent Process Control using Evolutionary Reinforcement Learning

by

Alex van Eck Conradie

Dissertation presented for the Degree

of

DOCTOR OF PHILOSOPHY IN ENGINEERING
(Chemical Engineering)

in the Department of Chemical Engineering
at the University of Stellenbosch

Professor C. Aldrich

STELLENBOSCH
December 2004

DECLARATION

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.

Alex van Eck Conradie December 2004

Now to Him who is able to keep
us from stumbling,
And to bring us faultless
Before the presence of His glory
with exceeding joy,
To God our Saviour,
Who alone is wise,
Be glory and majesty,
Dominion and power,
Both now and forever.
Amen.

Jude 25

ACKNOWLEDGEMENTS

My Father in heaven for refusing to let me go.

My supervisors, Professor Chris Aldrich and Professor Izak Nieuwoudt, for their mentorship, encouragement and support.

Professor Risto Miikkulainen, for his immeasurable guidance at the University of Texas at Austin.

Friends and family, for their love, kindness and patience.

Loving thanks to my wife, Anje, who put up with "Alex world" for long enough.

SYNOPSIS

A Neurocontrol Paradigm for Intelligent Process Control using Evolutionary Reinforcement Learning

Balancing multiple business and operational objectives within a comprehensive control strategy is a complex configuration task. Non-linearities and complex multiple process interactions combine as formidable cause-effect interrelationships. A clear understanding of these relationships is often instrumental to meeting the process control objectives. However, such control system configurations are generally conceived in a qualitative manner and with pronounced reliance on past effective configurations (Foss, 1973). Thirty years after Foss' critique, control system configuration remains a largely heuristic affair.

Biological methods of processing information are fundamentally different from the methods used in conventional control techniques. Biological neural mechanisms (i.e., intelligent systems) are based on partial models, largely devoid of the system's underlying natural laws. Neural control strategies are carried out without a pure mathematical formulation of the task or the environment. Rather, biological systems rely on knowledge of cause-effect interactions, creating robust control strategies from ill-defined dynamic systems.

Dynamic modelling may be either phenomenological or empirical. Phenomenological models are derived from first principles and typically consist of algebraic and differential equations. First principles modelling is both time consuming and expensive. Vast data warehouses of historical plant data make empirical modelling attractive. Singular spectrum analysis (SSA) is a rapid model development technique for identifying dominant state variables from historical plant time series data. Since time series data invariably covers a limited region of the state space, SSA models are almost necessarily partial models.

Interpreting and learning causal relationships from dynamic models requires sufficient feedback of the environment's state. Systemisation of the learning task is imperative. Reinforcement learning is a computational approach to understanding and automating goal-directed learning. This thesis aimed to establish a neurocontrol paradigm for non-linear, high dimensional processes within an evolutionary reinforcement learning (ERL) framework. Symbiotic memetic neuro-evolution (SMNE) is an ERL algorithm developed for global tuning of neurocontroller weights. SMNE is comprised of a symbiotic evolutionary algorithm and local particle swarm optimisation. Implicit fitness sharing ensures a global search and the synergy between global and local search speeds convergence.

Several simulation studies have been undertaken, viz. a highly non-linear bioreactor, a rigorous ball mill grinding circuit and the Tennessee Eastman control challenge. Pseudo-empirical modelling of an industrial fed-batch fermentation shows the application of SSA for developing partial models. Using SSA, state estimation is forthcoming without resorting to fundamental models. A dynamic model of a multi-effect batch distillation (MEBAD) pilot plant was fashioned using SSA. Thereafter, SMNE developed a neurocontroller for on-line implementation using the SSA model of the MEBAD pilot plant.

Both simulated and experimental studies confirmed the robust performance of ERL neurocontrollers. Coordinated flow sheet design, steady state optimisation and non-linear controller development encompass a comprehensive methodology. Effective selection of controlled variables and pairing of process and manipulated variables were implicit to the SMNE methodology. High economic performance was attained in highly non-linear regions of the state space. SMNE imparted significant generalisation in the face of process uncertainty. Nevertheless, changing process conditions may necessitate neurocontroller adaptation. Adaptive neural swarming (ANS) allows for adaptation to drifting process conditions and tracking of the economic optimum on-line. Additionally, SMNE allows for control strategy design beyond single unit operations. SMNE is equally applicable to processes with high dimensionality, developing plant-wide control strategies. Many of the difficulties in conventional plant-wide control may be circumvented in the biologically motivated approach of the SMNE algorithm. Future work will focus on refinements to both SMNE and SSA.

SMNE and SSA thus offer a non-heuristic, quantitative approach that requires minimal engineering judgement or knowledge, making the methodology free of subjective design input. Evolutionary reinforcement learning offers significant advantages for developing high performance control strategies for the chemical, mineral and metallurgical industries. Symbiotic memetic neuro-evolution (SMNE), adaptive neural swarming (ANS) and singular spectrum analysis (SSA) present a response to Foss' critique.

OPSOMMING

'n Neurobeheer paradigma vir intelligente prosesbeheer deur die gebruik van evolusionêre versterkingsleer

Dit is 'n komplekse ontwikkelingstaak om menigte besigheids en operasionele doelwitte in 'n omvattende beheerstrategie te vereenselwig. Nie-liniêriteite en menigte komplekse prosesinteraksies verenig as gedugte aksie-reaksie verhoudings. Dit is dikwels noodsaaklik om hierdie interaksies omvattend te verstaan, voodat prosesbeheer doelwitte doeltreffend bereik kan word. Tog word sulke beheerstelsels dikwels saamgestel op grond van kwalitatiewe kriteria en word ook dikwels staatgemaak op historiese benaderings wat voorheen effektief was (Foss, 1973). Dertig jaar na Foss se kritiek, bly prosesbeheer stelsel ontwerp 'n heuristiese affere.

Die biologiese prosesering van informasie is fundamenteel verskillend van methodes wat gebruik word in konvensionele beheertegnieke. Biologiese neurale meganismes (d.i., intelligente stelsels) word gebaseer op gedeeltelike modelle, wat grootendeels verwyderd is van die onderskrywende natuur wette. Neurobeheer strategieë word toegepas sonder suiwer wiskundige formulering van die taak of die omgewing. Biologiese stelsels maak eerder staat op kennis van aksie-reaksie verhoudings en skep robuuste beheerstrategieë van swak gedefineerde dinamiese stelsels.

Dinamiese modelle is of fundamenteel of empiries. Fundamentele modelle word saamgestel vanaf eerste beginsels en word tipies uit algebraïese en differentiële vergelykings saamgestel. Modelering vanaf eerste beginsels is beide tydrowend en duur. Groot databasisse van historiese aanlegdata maak empiriese modelering aantreklik. Singulêre spektrum analiese (SSA) maak die spoedige ontwerp van empiriese modelle moontlik, waardeur dominante veranderlikes vanaf historiese aanleg tydreekse onttrek word. Aangesien tydreeks data slegs 'n gedeelte van die prosesomgewing verteenwoordig, is SSA modelle noodwendig gedeeltelike modelle.

Die interpretasie en aanleer van kousale verhoudings vanaf dinamiese modele vereis voldoende terugvoer van die omgewingstoestand. Die leer taak moet sistimaties uitgevoer word. Versterkingsleer is 'n raamingsbenadering tot 'n doelwit-gedrewe leerproses. Hierdie tesis bewerkstellig 'n neurobeheer paradigma vir nie-liniêre prosesse met hoë dimensies binne 'n evolusionêre versterkingsleer (EVL) raamwerk. Simbiotiese, memetiese neuro-evolusie (SMNE) is 'n EVL algoritme wat ontwikkel is vir globale verstelling van neurobeheerder gewigte. SMNE is saamgestel uit 'n simbiotiese evolusionêre algorithme en 'n lokale partikel swerm algoritme. Implisiete fiksheidsdeling verseker 'n globale soektog en die sinergie tussen globale en lokale soektogte bespoedig konvergensie.

Verskeie simulasie studies is onderneem, o.a. 'n hoogs nie-liniêre bioreactor, 'n balmeul aanleg en die Tennessee Eastman beheer uitdaging. Empiriese modelering van 'n industriële enkelladings fermentasie demonstreer die aanwending van SSA vir die ontwikkeling van gedeeltelike modelle. SSA benader die toestand van 'n dinamiese stelsel sonder die aanwending van fundamentele modelering. 'n Dinamiese model van 'n multi-effek enkelladings distillasie (MEBAD) proefaanleg is bewerkstellig deur die gebruik van SSA. Daarna is SMNE gebruik om 'n neurobeheerder te skep vanaf die SSA model vir die beheer van die MEBAD proefaanleg.

Beide simulasie en experimentele studies het die robuuste aanwending van EVL neurobeheerders bevestig. Gekoordineerde vloeidiagram ontwerp, gestadigde toestand optimering en nie-liniêre beheerder ontwikkeling bewerkstellig 'n omvattende metodologie. Beheer veranderlikes en die paar van proses en uitvoer veranderlikes is implisiet en effektief. Maksimale ekonomies aanwins was moontlik in hoogs nie-liniêre dele van die toestandsruimte. SMNE het besondere veralgemening toegevoeg tot neurobeheerder strategieë ten spyte van proses onsekerhede. Nie te min, veranderende proses toestande mag neurobeheerder aanpassing genoodsaak. Aanpasbare neurale swerm (ANS) pas neurobeheerders aan tydens veranderende proses kondisies en volg die ekonomiese optimum terwyl die beheerder die proses beheer. SMNE bewerkstellig ook die ontwikkeling van beheerstrategieë vir prosesse met meer as een eenheidsoperasie. SMNE skaal na prosesse met hoë dimensionaliteit vir die ontwikkeling van aanleg-wye beheerstrategieë. Talle kwelvrae in konvensionele aanleg-wye prosesbeheer word deur die biologies-gemotiveerde benadering van die SMNE algorithme uit die weg geruim. Toekomstige werk sal fokus op die refyning van beide SMNE en SSA.

SMNE en SSA bied 'n nie-heuristiese, kwantitatiewe benadering wat minimale ingenieurskennis of oordeel vereis. Die metodologie is dus vry van subjektiewe ontwerpsoordeel. Evolusionêre versterkingsleer bied talle voordele vir 'n ontwikkeling van effektiewe beheerstrategieë vir die chemiese, mineraal en metallurgiese industrieë. Simbiotiese memetiese neuro-evolusie (SMNE), aanpasbare neurale swerm (ANS) en singulêre spektrum analiese (SSA) gee antwoord op Foss se kritiek.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

OBJECTIVES OF CHAPTER 1
• Highlight the challenges of process control.
• Contrast an algorithmic and biologically motivated approach to controller design.
• Motivate the need for non-linear process control.
• Causality in process control.

1.1 FOSS' CHALLENGE

Simplistically, chemical process control challenges entail the regulation of complex, frequently poorly understood physico-chemical phenomena in response to numerous unknown and uncharacterised disturbances. Regulatory control typically implies maintaining certain process states (or time-averages thereof) at optimal (mostly constant) conditions. The dynamic characteristics of a process may complicate or facilitate such regulatory control. The process' dynamic responses are an inherent function of the plant design, extending the realm of the control engineer to process design. Foss (1973) emphasised that a plant's control may be enhanced significantly through modification to existing process equipment or plant layout. Even though a process is well understood, the dynamic cause-effect relationships are typically difficult to unravel. The behaviour of a single state variable is governed by many other state and physical property variables through countless physical and chemical interactions. For example, the extent of catalyst deactivation is likely unknown and drifting. Clearly, no modelling exercise, regardless of its rigour, may account for all process uncertainties. Nevertheless, the control engineer needs a robust control system able to accommodate a variety of uncharacterised disturbances (Foss, 1973).

Foss (1973) cited the coupling between process variables as the primary reason for "the near inscrutable complexity of dynamic processes". Foss (1973) recognised that the coupling between most process variables is non-linear, affirming that an effective control system design often requires explicit treatment of non-linear dependencies. Non-linearities and complex multiple process interactions combine as formidable cause-effect interrelationships. A clear understanding of these relationships is often instrumental to meeting the process control objectives. Such objectives must relate to a quantitative performance measure. A single performance index may not be sufficient, with each criterion matching a different control strategy design. Balancing multiple business and operational objectives within a comprehensive control strategy remains a complex configuration task.

The crucial steps in control system configuration involve selecting a subset of the available process variables as controlled variables, pairing these controlled variables with suitable manipulated variables and determining the governing control laws. Generally, such control system configurations are conceived in a qualitative manner and with pronounced reliance on past effective configurations (Foss, 1973). Thirty years after Foss' critique, control system configuration remains a largely heuristic affair. Control system design requires a combination of steady state and dynamic process information. Designs based solely on steady state information prove inadequate, as dynamic information is necessary for proper pairing and tuning. Fortunately, a complete dynamic description of the process proves unnecessary. Only dominant dynamic elements need to be accounted for in process models. Dynamic models may range from rigorous non-linear differential equations to linear transfer functions derived from plant step change data. However, conventional control design methodologies require that dynamic models adhere to a particular model structure (e.g., linear state space or Laplace transform), though process dynamics are confined rarely to parameterisation within these model structures. To avoid trial-and-error control system configuration, dynamic modelling (in one form or another) remains a critical component of control system configuration (Foss, 1973).

As opposed to control design considerations, control system operation presents its own set of challenges. Drifting process conditions affect a fixed control system adversely, demanding adaptation of the control system parameters in lieu of the current process conditions. The high cost of maintaining an adequate dynamic model also proves challenging and on-line adaptation may prove unreliable. High dimensionality, interactive behaviour, poorly understood dynamics, limited measurement of the underlying state variables and undetermined control structures present unique challenges (Foss, 1973).

No single process control theory is able to deal with the wide range of challenges posed by process control design. However, complementary theories that address key issues in process design, dynamic modelling and control system configuration should be sought. Chemical engineers have produced scores of journal articles of simulated control studies, which assume that the controller has access to all the state variables without measurement error. Control configuration synthesis using only sparse measurements and limited process knowledge remains elusive though "desperately needed in the process industries" (Foss, 1973). A practical method of state estimation, using available measurements or historical process data, is a key to process control progress. The inadequacies of conventional control methods insist on an extension of existing methods. Particularly, the synthesis of control systems in plants with high dimensionality remains a central issue. Heuristic methods to this problem are prone to flounder once confronted by the curse of dimensionality.

A broadly applicable solution to control structure design cannot be based on heuristic approaches that rely primarily on engineering judgement and personal preference. The method that addresses a wide scope of process control challenges, should be founded on broadly applicable dynamic modelling techniques and relate economic objectives to process control objectives directly within the realm of controllability (i.e., highly non-linear process may prevent effective control at the economic optimum). Such a method should cope with sparse and poor measurements and imprecise process knowledge. The control structure design task should thus involve an optimisation task, free from subjective design input (Foss, 1973).

The representation of process dynamics remains a major task. Phenomenological modelling allows for a consistent model structure that requires identification of several model parameters. However, it is not sensible to construct accurate non-linear models, when these non-linearities must be excluded from the control structure design. Conventional control design methods frequently require that the process' dynamics conform to a suitable model structure (e.g. linear), resulting in poor utilisation of existing non-linear models. Rigorous simulation with powerful simulation tools, such as ASPEN and HYSYS, offers limited benefit to conventional control theory for codifying process knowledge into the control law (Foss, 1973). Non-linear model predictive control is one exception.

Though published in 1973, Foss' article serves as a distant mirror in which the current state of process control may still be recognised. Finally, Foss (1973) closed his critique on classical control theory, "There is more than a suspicion that a work of genius is needed here, for without it the control configuration problem will likely remain in a primitive, hazily stated, and wholly unmanageable form". Though melodramatic, these words are echoed in a publication by Hecht-Nielsen (1988), who suggested a different approach.

1.2 CONVENTIONAL AND NEURAL DESIGN METHODOLOGIES

1.2.1 Algorithmic and biological approaches to information processing

To program a computer to perform a given control task, a person needs to first comprehend the control task, and second develop an algorithm for executing the automated task. For complex functions, such as computed axial tomography, devising an algorithm awaits the birth of geniuses. Likewise, many real world applications exist for which it is difficult or virtually impossible to communicate in an algorithmic format, a logical sequence that will lead to the successful execution of the task. Although an effective algorithm may be difficult to establish or may not exist, in several cases, the task may be specified exactly by means of a problem statement. Many such tasks exist, for example, robust algorithmic software that allows a robot arm to "pick-up-an-object" in a changing environment is difficult to develop; yet the problem statement is simplistic (Hecht-Nielsen, 1988).

The design of controllers that allow increased independence from human interaction is desirable. However, autonomous operation is dependent on the controller's ability to maintain robust performance, despite a variety of unexpected occurrences in an unstructured or uncertain operating environment. The success of biological systems at performing numerous such complex tasks, may be regarded as a significant motivator and framework for the design of adaptation algorithms and robust learning techniques (Gupta & Rao, 1994).

Biological methods of processing information are fundamentally different from the methods used in conventional control techniques. In biological systems, the plan to execute a task is formulated on the conscious level. In order to “pick-up-an-object”, it is necessary to gauge the relative position of the hand to the object, and compute directional vectors. In biological systems, these computations are entirely performed at the subconscious level - the control task is not evaluated consciously in terms of muscle coordination or joint angles. In contrast, an algorithmic design technique for the control of a robot arm needs to explicitly compute and coordinate the angles of different robot joints to produce a desired trajectory. Such an algorithmic control law may fail completely should the desired task or the environment change. Even though seemingly complex to describe such a task mathematically, biological systems are able to learn new tasks and adapt to a changing environment with relative ease (Gupta & Rao, 1994).

Procedures for designing conventional controllers, such as PID controllers and model reference adaptive controllers, are model based. These design methods generally involve constructing an explicit fundamental model using physical and chemical laws that govern the dynamic system (Gupta & Rao, 1994). Biological neuronal control mechanisms (i.e., intelligent systems) are based on partial models, largely devoid of the underlying natural laws of the system. For example, swallows have no understanding of the complex mathematics that describe aerodynamics, yet are capable of extraordinary aerobatics. Biological systems are able to determine the major cause-effect relationships from partial models, giving rise to robust control strategies successful at dealing with uncertainty and unstructured environments. Neural control mechanisms are carried out without a pure mathematical formulation of the task and the environment. A task is executed without solving integral, differential or other mathematical equations. Biological systems rather rely on knowledge of cause-effect interactions. Interpreting and learning such causal relationships requires sufficient feedback of the state of the environment (Gupta & Rao, 1994).

The differences between algorithmic control and neural control (i.e., as a sub-class of intelligent control) are highlighted in the following two sections.

1.2.2 Algorithmic Control System Design

The conventional design methods for control systems involve constructing a mathematical model of the system's dynamics and the utilisation of analytical techniques (classical or modern) for the derivation of a control law. Such mathematical models comprise sets of linear/non-linear differential equations, which are usually derived with a degree of simplification or approximation. The modelling of physical systems for feedback control generally involves a balance between model accuracy and model simplicity (Gupta & Rao, 1994).

Conventional controller development techniques are less useful, should a representative mathematical model be difficult to obtain. Also, even though an accurate model may be produced, the underlying nature of the model may make its utilisation using conventional control design difficult (Gupta & Rao, 1994). In order to achieve satisfactory controller performance in the event of poorly understood dynamics, adaptive control or robust control approaches are typically employed.

Adaptive control may be utilised as a solution to the control of complex plants and in the presence of greater uncertainty. The controller parameters in the fixed control structure are adapted by an adaptive algorithm that ensures that the desired performance level is maintained. An adaptive control system monitors a performance index using the current dynamic state of the plant under control. By comparing the actual performance to the desired performance index, the adaptation algorithm modifies the parameters of the controller to track the desired plant performance.

Although techniques such as model reference adaptive control (MRAC) and automatic tuning regulators have been used widely, the application of these techniques to many realistic problems has proven problematic (Gupta & Rao, 1994).

The theoretical basis of linear system theory is a mature science. Should it be possible to approximate the actual physical plant as a member of a class of systems (e.g. having first or second order linear dynamics), the use of robust controllers has general application. The implementation of robust controllers is relatively simple when system stability and reasonable performance is the only concern.

Further, the use of robust linear and adaptive controllers may prove effective, should a non-linear plant be operated in a narrow range of process conditions (i.e., locally linearised models are appropriate); but frequently prove inadequate for controlling operations over a broad range of operating conditions (Gupta & Rao, 1994).

In addressing the shortcomings of robust and adaptive control techniques, the design of general, practical non-linear controllers remains largely outside the reach of the available algorithmic theories. Between robust and adaptive linear control and non-linear control development, linear multi-input multi-output techniques have found niche applications, such as Dynamic Matrix Control (DMC) in the petroleum industry. Likewise, a number of conventional methods exist for developing non-linear controllers for specific classes of non-linear systems. These methods are system specific. A given design methodology may not be universally applied to all classes of non-linear systems. The wide inappropriate use of multi-loop linear controllers or adaptive linear controllers in non-linear systems, simply emphasises the need for effective methodologies for non-linear controller design (Stephanopoulos & Han, 1996).

1.2.3 Neural Controller Design

Stephanopoulos & Han (1996) claimed that the growth in "intelligent" control has been a reaction to the realisation that non-linear control theory is not able to offer practical solutions to today's control challenges. The evolution in the control paradigm has been fuelled by the need to deal with increasingly complex systems, and the need to accomplish increasingly demanding design requirements (i.e., operation closer to the economic optimum) with less precise knowledge of the dynamic system (Gupta & Rao, 1994).

In order to deal with uncertain plant dynamics a controller needs to estimate or allow for the existence of unmeasured information during operation. Should the strategy for dealing with unmeasured states be effective, the controller may be considered optimal (Gupta & Rao, 1994).

A control system is deemed a learning control system, as opposed to an adaptive system, when a control strategy has been developed based on past experience or performance. Adaptive systems differ from learning systems in that adaptive systems regard the current plant state as novel, whereas learning systems reconcile past learned plant operating regimes with the current state and act accordingly. Adaptive systems typically only adjust the parameters in a particular control structure. Learning systems may change the type and structure of a controller, or vary the parameters of the controller, after observing that the current controller is not performing satisfactorily (Gupta & Rao, 1994).

By endowing the controller with learning ability, the operating regime of the controller may be effectively increased. Learning occurs typically through the evaluation of examples or trials. The learning feature precludes the existence of a fundamental plant model, as plant history is utilised to develop empirical dynamic models used during learning. The control system is consequently able to compensate for a greater variety of changes in the plant conditions. Also, a learning system may have the desired ability for ever-improving future performance, based on information the controller has gained in the past through dynamic learning. Learning methodologies may be implemented in either on-line or off-line paradigms (Gupta & Rao, 1994).

Neural networks, with their ability to learn and their inherent massive parallel processing, introduce numerous opportunities for developing superior control structures for complex systems. A neurocontroller generally takes the form of a multi-layer neural network, where the neuron weights are the adaptable parameters during learning. Neural networks are able to arbitrarily approximate non-linear functions, which allows for the effective synthesis of non-linear controllers (Hornik et al., 1989). Neurocontrollers solve control problems by mapping process variables sensor readings into calculated actions for manipulated variable outputs. A neural network's parallel processing capability also makes its use for multivariate control systems attractive. The ability of a neural network, inherent to its structure, to robustly generalise to regions of the state space not encountered during learning or training, is of particular value in controller implementation to real world problems (Gupta & Rao, 1994). Such non-linear control would prove highly effective in controlling highly non-linear process plants.

1.3 MOTIVATION FOR NON-LINEAR CHEMICAL PROCESS CONTROL

In spite of the extensive research in adaptive (i.e., self-tuning) controllers and model reference control, there are many control problems in the chemical processing industries for which current control design techniques are inadequate. Many of the limitations of current adaptive controllers arise when controlling poorly modelled non-linear systems (Ungar, 1991).

Many chemical processes are highly non-linear. Non-linearities may be intrinsic to the physics or chemistry of a process such as in supercritical extraction, in which complex phase behaviour leads to a sensitive dependence on operating conditions. Non-linearities may also arise through the close coupling of simpler processes. For example, when heat integration is used to save energy, the process becomes tightly coupled, more multivariate and more difficult to control (Ungar, 1991).

Well-established process technologies also present challenges. Distillation is a highly non-linear process, and therefore remains one of the most widely studied control problems. In aluminium casting, different rates of cooling lead to different regimes of operation and produce aluminium with different properties. Such processes offer challenges in that no satisfactory first-principles model of the process exists and there are too many control parameters to experiment with randomly (Ungar, 1991).

Extensive theoretical and experimental studies have likewise been made of both batch and continuous stirred tank reactors (CSTRs). Although such reactors may be described (approximately) by simple equations, they often exhibit complex behaviour such as multiple steady states, periodic and chaotic behaviour. Bioreactors also pose a complex control problem in that the dynamics of the system may vary significantly depending on the current process conditions. Model-based control techniques for such reactors require the derivation of effective control laws, despite incomplete process models (Ungar, 1991).

Currently, new process routes are designed with an increasing number of non-linear elements. Despite greater non-linearity, methodologies for process design analysis and control system synthesis typically disregard the inherent non-linearities (Seider et al., 1990). Although the percentage of units requiring some form of advanced control is small, such key process units (as described above), may account for up to 40% of the gross revenue generated in the USA process industries (Harris & Palazoglu, 1998).

The need for non-linear control methodologies has been driven by a number of factors. The process industries' greater emphasis on optimal process design, safe plant operation, increased product quality and higher profitability call for improvements to existing control systems. Over wide operating regions, a linear controller, designed through local linearisation of the non-linear model equations, may not compensate effectively for the system non-linearities. Non-linear controllers deal effectively with operation over a wide operating region. Numerous non-linear processes exhibit complex dynamic behaviour, such as non-minimum phase behaviour, dead time and hysteresis, resulting in poor responses by linear controllers. Non-linear controllers are expected to deal more effectively with process model uncertainties (i.e., plant/model mismatch), owing to more efficient generalisation. Increased product specifications and environmental regulations also make linear controllers less feasible.

The increased availability of computer processing power has aided a shift to non-linear control techniques, which are more computationally intensive than linear control algorithms (Kuttisupakorn et al., 2001). Increased computer processing power has also aided data mining techniques, whereby causal relationships may be learned from historical plant data.

1.4 CAUSALITY - THE LANGUAGE OF EVOLUTION AND LIFETIME LEARNING

A biological system (i.e., an organism) responds to its environment by adapting on both genetic and lifetime learning time scales. Central to this adaptation is causality, i.e. cause-effect relationships, that is vital to develop human and animal understanding of any given environmental occurrence. Causality is essential in decision-making, providing a foundation for selecting actions that are likely to have a desired result. Life depends on information. More importantly, causal knowledge must be generated from such informational input. The basic blueprint of an organism's behaviour is encoded in DNA. This blueprint is like a fixed set of good ideas and sensible expectations. On the phenotypical level, the types of sensory organs and output structures that an organism is born with, is a kind of genetically encoded knowledge. For example; being born with hooves, hands, or flippers is an explicit prediction of the kind of world that an organism will need to survive in. Also, DNA builds a network of neural or nervous system connections that are aimed in a general manner to basic tasks like eating, reproducing, resting and surviving which may be improved upon during lifetime learning. Should a human being be born to live underwater, no degree of evolutionary or lifetime learning will prevent the death of the infant. This genetic legacy has been determined by evolution on a time scale of possibly thousands of generations (McCrone, 2002).

Once an organism is born in an expected environment, the organism learns from cause-effect relationships. A prehistoric human became more efficient at evading predators during a lifetime of causal interaction with predators. Similarly, all lifetime response pathways reflect the threats and opportunities that an individual organism has learnt to deal with during its lifetime. Every environmental occurrence presents a set of challenges and the brain's neural pathways must present an equally precise mental response. The general understanding and reactions coded by the genes and a lifetime of experiences must be focused to zero in on the meaning of each moment in time, following an ever-narrowing cone of adaptive activity. The neural landscape is continuously sculpted by cause-effect experience, though the brain draws significantly from past causal understanding. The massive, parallel structure of the millions of neurons in the brain allows for an untold ability to generalise behaviour to novel environmental circumstances. The brain is able to anticipate future events by predicting that whatever it knew to be true about the world a moment ago, remains true in the current moment, creating a state of expectancy. Even though a completely unexpected event may occur, the brain assumes that numerous expectancies remain valid (e.g., the Earth's gravitation remains the same) (McCrone, 2002).

Although a great deal may be learnt from correlated circumstances and there are similarities between correlation and causality, causality is far more difficult to underpin than correlation. Correlation may be determined directly from a single uncontrolled experiment. Causal conclusions require controlled experiments or causal assumptions that are difficult to assess in a single study. However, children somehow learn cause-effect relationships without performing controlled experiments. Pearl (1997) stated that the word "cause" is not in the language of probability theory and used the following example. Probability theory cannot produce a conclusion such as "mud causes rain". Probability theory only proves that the two states (i.e., mud and rain) are mutually correlated or dependent, implying that if one appears, the other may also be expected to appear. Formally, the more logical statement "rain causes mud" also presents difficulty, as probability theory cannot exclude the reverse. Ignoring the fact that the appearance of mud could be caused by almost limitless other possibilities, correlation clearly does not imply causality. Correlation may thus also represent a non-causal relationship.

The fundamental difference between correlation and causality is illustrated by the inclusion of covariates to a regression equation. The effect of one variable X on another Y may be adjusted by including variations of another variable Z in the regression analysis. The variable Z partitions the elements in X and Y into homogenous groups relative to Z. The relationship between X and Y is assessed in each Z-group and the results are averaged. Frequently, this brings about Simpson's paradox, which states that any statistical relationship between two variables may be negated or reversed by including additional variables in the analysis. For example, data may reveal that students who smoke obtain higher marks than those who do not smoke, but including age as a covariate the result reveals that smokers obtain lower marks than non-smokers in every age group. Covariate selection remains difficult and based on intuition (Pearl, 1997). In the processing industries controlled plant tests (i.e., with Z fixed) are difficult to obtain since process conditions vary, making true causal relationships, used for control structure design, difficult to assess through standard statistical techniques.

Causality also presents many other challenges. Chaining refers to a temporal chain of events that are described by the sequential events $[A_1, A_2, ..., A_{n-1}]$, which terminates in A_n. The question remains as to which event from A_1 to A_{n-1} caused A_n. For example, a man phones his friend, A_1, for help. The friend jumps in his car and speeds to the man's house, A_2. At an intersection, the friend is struck by a car, A_{n-1}. The friend dies (A_n). Who caused the friend's death? Was it the man who phoned, the friend himself or the driver of the car? Conjunction refers to a confluence of events, $[A_1, A_2, ..., A_n]$ with the resulting event, B. For example, a raincoat manufacturer wished to increase sales, A_1. She motivates her sales staff, A_2. She increases the advertising budget by 100%, $\mathrm{A}_{\mathrm{n-1}}$. Rainfall during the month is slightly higher than normal, A_n. Sales increase by 20%, B. The extent to which the increase in advertising budget, $A_{n-1,}$ contributed to the increase in sales is uncertain. A collection of events culminated in the increase in sales (Zadeh, 2001).

Pearl (1997) suggested a graphical language for determining the validity of causal relationships. Similarly, Johansson and Hägglund (2000) used a graphical technique for determining the control structure of single-input single-output (SISO) controllers with additional measurements. The graphical technique suggested improvements to the basic single-input single-output (SISO) feedback controller by incorporating relevant additional measurements in a cascade or feedforward control scheme.

In contrast, evolutionary reinforcement learning is a probabilistic method that initially assumes arbitrary cause-effect relationships between measured variables and manipulated variables. True causal relationships are confirmed by propagation of particular relationships (i.e., interactions) into subsequent generations. The efficiency of evolutionary reinforcement learning in exploiting causal relationships in a dynamic environment suggests that the language of evolution and lifetime learning contributes to an understanding of causality.

1.5 OBJECTIVES OF THIS STUDY

The objectives of this study have been to establish a neurocontrol design paradigm for the process industries within an evolutionary reinforcement learning framework. The neurocontrol design methodology needed to address the following shortcomings in conventional control design methodologies by having the following characteristics:

- Non-linear control structure thereby offering superior control for highly non-linear processes (Economou & Morari, 1986).
- Multi-input multi-output (MIMO) control structure and able to deal with severe process interaction through optimal pairing of process variables with manipulated variables.
- The design methodology must be able to use all possible process model structures for controller development.
- Robust non-linear model development must be possible from historical and experimental plant data. State estimation must be forthcoming without resorting to fundamental models. Controller development must be possible from partial models.
- Non-heuristic approach that requires no engineering judgement or knowledge, making the methodology free of subjective design input.
- The methodology must not be confined to set point regulation, but incorporate any control objective such as economic objectives.
- Allow the development of plant-wide control structures for processes with high dimensionality.
- The control structure must be able to adapt to process changes or shifting process objectives on-line, providing robust performance in the face of significant process uncertainty and unmeasured disturbances.

A novel evolutionary reinforcement learning algorithm, Symbiotic Memetic Neuro-Evolution (SMNE), has been developed to achieve these objectives within an evolutionary reinforcement learning framework. Adaptive Neural Swarming (ANS) is an add-on to SMNE, allowing on-line adaptation of neurocontrol structures. Simulation studies verified the methodology using several simulated processes obtained from the biotechnology, mineral processing and chemical industries. A pilot plant of a novel batch distillation configuration, Multi-Effect Batch Distillation (MEBAD), verified the neurocontrol methodology in a real-world application. Plant-wide control was established for the Tennessee Eastman control challenge.

Finally, Gupta and Rao (1994) paraphrase the goals of this thesis:

"It is our hypothesis that if the fundamental principles of neural computation used by biological control systems are understood, it seems likely that an entirely new generation of control methodologies can be developed that are more robust and intelligent, far beyond the capabilities of the present techniques based on explicit mathematical modelling."

1.6 CONCLUDING REMARKS

Intelligent control techniques offer a competitive edge in an ever-competitive economic environment. Although numerous applications of intelligent process control have been reported, few progress past simulation studies and only a small percentage progress to laboratory prototypes. Even fewer become products in the process control marketplace (Chiu, 1997). Narenda (1996) surveyed the literature published from 1990 to 1995 and found that 9955 articles were published in the engineering literature with "neural networks" in the title. Of these, more than 8000 related to function approximation and pattern recognition (i.e., static systems). One thousand nine hundred and sixty papers dealt with control using neural networks, of which only 353 represented applications. Of these 353 articles, 45% pertained to theory, with 28% dedicated to simulation studies and 23% represented laboratory experiments. A mere 4% were dedicated to real-world application, though this low percentage could be ascribed to industrial proprietary considerations (Narenda, 1996).

Despite the enormous potential of intelligent control systems, there is minimal incentive to switch from cost-effective PID control, unless the application truly demands non-linear control. PID control has proven satisfactory performance in set point regulation for a vast number of applications. Other than providing non-linear control through state feed back, intelligent control provides solutions that analytical control methodologies do not address. These include tracking changing economics brought on by changing raw material prices, energy costs and disturbances. The return on investment for intelligent control over conventional techniques must be justified, also taking into account the greater risk of implementing largely unproven technologies (Chiu, 1997).

2 PROCESS DESIGN AND PLANT-WIDE CONTROL

OBJECTIVES OF CHAPTER 2

- Emphasise the link between process control and process design.
- Describe the prevailing approach to plant-wide control both mathematical and heuristic, thereby providing a platform from which to highlight the differences and similarities to the evolutionary reinforcement learning approach in this thesis.

2.1 COORDINATED PROCESS DESIGN AND CONTROL OPTIMISATION

Before the 1970's, process designs relied on minimal heat integration and large inventories (i.e., extensive storage of intermediate products) that dampen process upsets between unit operations. Also, recycle streams were avoided to prevent disturbances in downstream unit operations from propagating to upstream unit operations. Isolating unit operations, by avoiding energy and material feedback, simplified the control system design, albeit at higher operating and capital costs. Control strategies evolved into a unit operation approach, which over several decades culminated into highly effective control strategies for individual unit operations (Luyben et al., 1997).

However, the 1970's energy crisis necessitated energy integration. Pressure increased to reduce capital and operating costs, to improve plant safety and to address environmental issues. Process design engineers responded by introducing heat integration (e.g., pinch technology), reducing surge vessel installations and relying on recycle streams. Improved heat integration and higher product yields through recycle, are economically viable on the steady-state flow sheet, but present process control challenges (Luyben et al., 1997).

Also, new process routes amplify the challenges to process control, as new process technologies are frequently complex with poorly understood dynamics. Despite a lack of adequate process information, new process routes need to be exploited sooner to ensure a positive net present value. Thereby, the extensive use of process specific operating experience and expert knowledge is not possible. Furthermore, process designs continue to become more non-linear, more interactive and consequently more difficult to control. A plant-wide control perspective is imperative, which looks beyond individual operating units.

Considering that numerous processing routes may exist to produce a given product, the design degrees of freedom approach the order of thousands. The production of penicillin has taken both chemical and biotechnology routes. The recovery of metals (e.g. copper) may take either a pyrometallurgical or hydrometallurgical route. Likewise, the delivery of pure oxygen may be via air separation units, pressure swing adsorption and membrane technology. The choice of process route may be a complex one, dictated to by production volumes, raw material quality and quantity and access to patented processes that may provide significant technology leverage. Economic considerations include not only the capital and operating costs, but also market forecasts, project life, stability of governments, currency fluctuations and the availability of trained human resources. Several of these considerations are difficult to quantify mathematically and rely on engineering and business judgment.

Typically, the design degrees of freedom are therefore limited to the capital cost of an installation and the unit cost or contribution for producing the product, which are easier to quantify mathematically. The process equipment largely determines the capital cost, while the operating region dictates the operating cost. The design degrees of freedom include all parameters related to the size of the equipment. For example, design degrees of freedom pertain to reactor volumes, heat exchange area and the number of stages in mass transfer contacting equipment. Once a short-list of process routes has been selected, each one is subjected to a capital and operating cost analysis.

Process design tasks are formulated with one or several objectives that require steady state optimisation of the non-linear programming definition, so that:

$$\min_{x,p,u} f\left(\bar{x},\bar{p},\bar{u}\right) \tag{2-1}$$

$$g\left(\bar{x},\dot{\bar{x}},\bar{p},\bar{u}\right)=0 \tag{2-2}$$

$$h\left(\bar{x},\bar{p},\bar{u}\right)\leq 0 \tag{2-3}$$

where f is the objective function, g is the system of differential and algebraic equations and h is the set of design constraints. In equations 2-1 to 2-3 $\bar{x}$ is the state variable vector, $\bar{p}$ is the vector of design parameters and $\bar{u}$ is the manipulated variable vector. Seider et al. (1990) indicated that when the steady state equations (Equation 2-2) have multiple solutions and are highly non-linear, conventional optimisation routines (e.g., successive quadratic programming) frequently do not converge to the global optimum. Steady state flow sheet design tools (e.g., Pro II) are typically not equipped with suitable optimisation algorithms to locate economic optima in complex operating regions. For design optimisation tasks, high dimensional state spaces also limit the viability of generalised design optimisation tools (Seider et al., 1990). Operation at the global economic optimum is critical.

2.1.1 Process design and plant-wide control

Process design and controller development are typically matched discontinuously as shown in Figure 2-1. Once the optimal economic set points have been determined, these set points are carried forward to the control development phase. An appropriate control structure and set of tuning parameters are devised for the steady state optimum, fixed with little regard for changing economic factors and disturbances. For example, the current market conditions (e.g., the cost of a raw material) or a disturbance (e.g., catalyst decay) may require a change in the plant's operating point to maintain operation at the economic optimum. Plant-wide control addresses this decoupling between the steady-state economic optimum and the control structure of the plant as shown in Figure 2-2.

Generally, the concept of plant-wide control refers to the control strategy, with an emphasis on structural design, rather than on the tuning of the control law parameters. At the highest level, plant-wide control is only concerned with the interactions between a process and the forces (i.e. controllers, external disturbances) that care to impact upon it. A plant-wide control system is typically divided into an economic objective, supervisory/predictive and a regulatory layer. The economic objectives based on current market condition may be updated as frequently as on an hourly basis (Larsson & Skogestad, 2000). For example, real-time pricing of electrical power benefits both the electricity supplier and the consumer, provided the consumer is in a position to react to periods of high overall electrical demand. Should the consumer turn down the plant's electrical demand during peak overall electrical demand (i.e., high unit cost of electricity), the electricity supplier may have no need to start-up an additional power plant. The consumer benefits by being able to take advantage of low demand periods, when the cost of electricity is extremely low. The supervisory (predictive) layer re-computes the steady state economic optimum based on the current market conditions (i.e., an updated cost function) and the current disturbances, given a steady state model of the process. The new steady state optimum propagates to the regulatory layer as set points, where feedback control stabilises the plant at the new operating condition. The regulatory control layer is typically incorporated with a decentralised control system using multi-loop PID controllers.

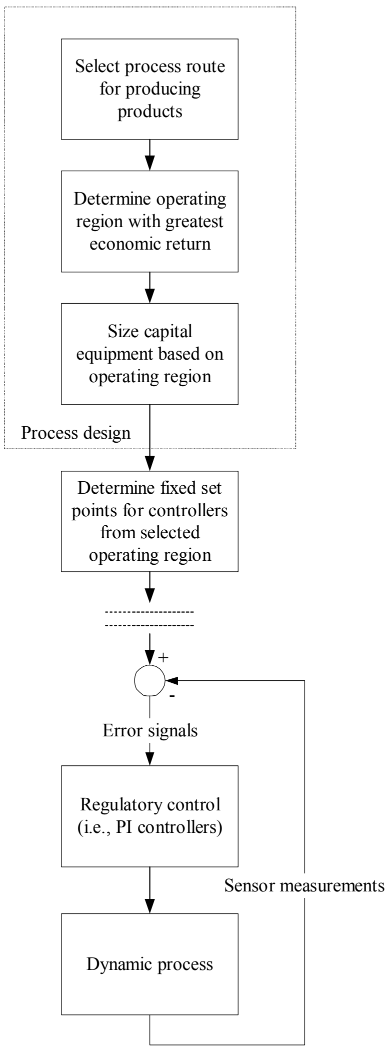

Select process route for producing products
Determine operating region with greatest economic return
Size capital equipment based on operating region
Process design
Determine fixed set points for controllers from selected operating region
+
-
Error signals
Regulatory control (i.e., PI controllers)
Sensor measurements
Dynamic process

Figure 2-1 - Typical design & control hierarchy.

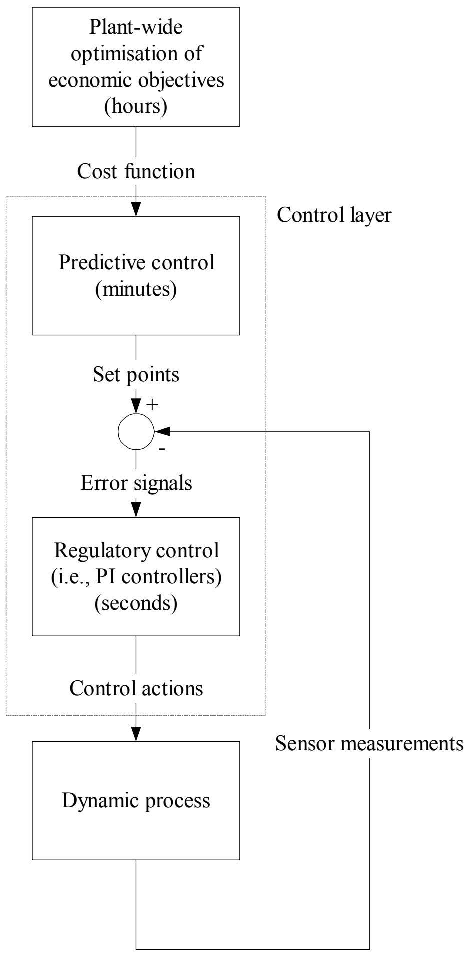

Plant-wide optimisation of economic objectives (hours)
Cost function
Control layer
Predictive control (minutes)
Set points
+
-
Error signals
Regulatory control (i.e., PI controllers) (seconds)
Control actions
Sensor measurements
Dynamic process

Figure 2-2 - Ideal conventional control hierarchy

Conversely, Figure 2-3 illustrates a single optimising controller, which stabilises the process and provides optimal control responses (i.e., control actions) in lieu of the process' production objectives. Larsson and Skogestad (2000) considered the integration of the optimisation and control into a single optimising controller as an unrealistic goal, due to the cost of detailed dynamic modelling that such a scheme would require. The good local control performance of standard feedback controllers, without much need for modelling, is cited as another reason for not pursuing a scheme as in Figure 2-3. Further, Larsson and Skogestad (2000) claimed that a decentralised control system is less sensitive to model uncertainty (no explicit model is typically used). By designing a rational and systematic control configuration with engineering judgement, the necessary process information is included in the control system. A centralised controller would need to determine this process information from a dynamic process model. Hereby, Larsson and Skogestad (2000) ignore the advances in empirical modelling techniques that allow rapid, automated data mining of historical plant databases for determining system dynamics. In contrast, Stephanopolous and Ng (2000) expect advancement by the explicit consideration of the non-linear nature of chemical processes in model-based MIMO control systems. Particularly, process controllability should be addressed from a non-linear perspective.

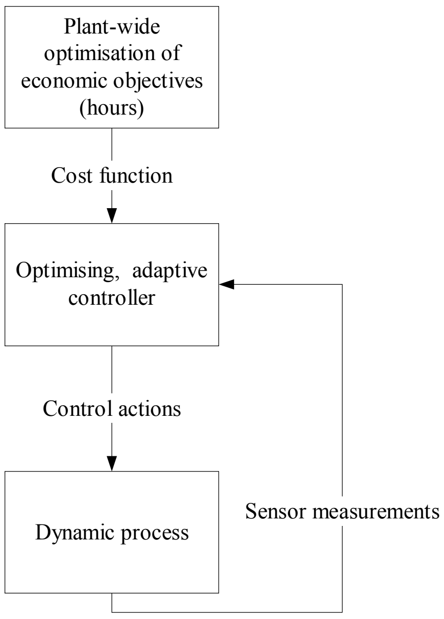

Plant-wide optimisation of economic objectives (hours)
Cost function
Optimising, adaptive controller
Control actions
Dynamic process
Sensor measurements

Figure 2-3 - Integrated optimisation and control for control layer.

The process design determines the process' controllability and flexibility. Controllability is inherent to the process (i.e., the open loop response) and is independent of the particular control structure or parameters. Controllability reflects the process' ability to reject disturbances and negate the severity of multi-variable interactions. Flexibility is the ease with which the process may be shifted from one operating region to another. Deviations from the optimal operating conditions, owing to poor controllability, could have severe economic implications (Mohideen et al., 1997). In an increasingly competitive economic climate, chemical plants need to reduce inventories (i.e., operating capital used ineffectively) and must be capable of delivering product on specification, at the required rate, on demand. A responsive process facility should track customer demands on both quality and quantity of product, making frequent feasible transitions between operating points necessary (Rowe et al., 1997). This suggests that interaction between process design and control in terms of operability should be addressed at the earliest stages of the process design (Luyben and Floudas, 1994a).

Contrarily, process flexibility and stability (i.e., controllability analysis of the open-loop response at the operating conditions via bifurcation analysis) are rarely considered in the process design. Most process design approaches concentrate only on obtaining the optimal economic process design, selected from a practical number of alternate designs that satisfy the operational constraints. Controllability is generally omitted from consideration in the process design, even though poor control could impact severely on the process operating cost. Conversely, traditional process control approaches consider a fixed process design configuration when selecting the pairing and tuning parameters between controlled and manipulated variables. However, small changes to the process flow sheet could simplify reaching the control objectives profoundly (Stephanopoulos, 1983). Conventional methodologies only determined the controllability after the process design has been fixed (Luyben and Floudas, 1994a). Thereby, the process design and controller development progress in a sequential manner. This sequential approach may lead to economically sub-optimal and inefficient designs. The ability of a control system to reject disturbances and remain robust, despite model uncertainty, is highly dependent on both process design and the operating region. Most integrated design and control research has focused on analysis tools that provide a comparable measure of controllability for alternate process designs (Bansal et al., 2002). However, controllability analyses provide minimal insight into how the process design should be augmented to meet the economic objectives and operating constraints. Revised control system development and retrofit solutions become a common occurrence to address operability bottlenecks, should the steady state economics and control not be considered simultaneously. The long-term economic viability of a production facility is highly dependent on the flexibility and stability of the process.

2.1.2 Over-design in the process industries

The chemical industry in the United States may be cited for numerous instances of over-design. Over-design usually compensates for model and design uncertainties, allowing for increases in capacity and for margins of safe operation. However, over-design also results in order to avoid operation near or within complex operating regimes. Such operating regimes may be characterised by hysteresis and periodic or chaotic behaviour. Designs that avoid these operating regions are frequently deemed prudent (i.e., if these regions were known or suspected to exist) as safeguard against unreliable operation. However, designs that constrain key design parameters, p, to avoid these regions of operation, may prevent operation near steady-state economic optima. Common process characteristics (Table 2-1) that cause control difficulties for non-linear and linear controllers alike, are more pronounced in more complex operating regions. Despite the possible greater economic benefit of operating at process conditions with complex dynamics, this over-design approach prevails (Seider et al., 1990). Unit operations that exhibit complex dynamic and steady state behaviours include exothermic reactors, aerobic fermentation, heterogeneous azeotropic distillation columns and supercritical extraction.

Table 2-1 - Common Process Control Design Complications (Bequette, 1991)

Common Process Characteristics
• Multivariable interactions between manipulated and controlled variables
• Unmeasured state variables
• Unmeasured and frequent disturbances
• High-order and distributed processes
• Uncertain and time-varying parameters
• Constraints on manipulated and state variables
• Dead-time on inputs and measurements
• Sensor noise and inaccurate measurement

Brengel & Seider (1992) regarded this over-design phenomenon as an important opportunity and challenge facing design engineers. Design, modelling and particularly control techniques that allow closer operation to more complex regimes should reduce the instances of over-design in the process industries sharply. The process design and control system synthesis should be coordinated, thereby maximising a profitability objective function penalised for poor controllability. Maximising the process' flexibility is a key requirement.

Luyben and Floudas (1994a) stated that the goal of coordinated process design and controller development is to determine the best-compromise process configuration among the competing economic and open-loop controllability objectives. This best compromise solution may include minimising the cost of the process equipment and optimising any measure of controllability (e.g., relative gain array) and flexibility. Coordinated process design and controller development thus becomes a multi-objective optimisation (Luyben and Floudas, 1994a).

Luyben and Floudas (1994b) used Bristol's relative gain array (RGA) in assessing controllability. RGA was developed for use in control pairing of linear systems. This controllability objective was incorporated with the steady state economic optimisation in a multi-objective design of a reactor-separator-recycle system. The usefulness of linear analytical tools for multi-loop SISO pairing may be limited for non-linear systems, since the process gain matrix is calculated at the steady state economic operating point. For non-linear systems, small deviations from the desired steady state operating point could change the magnitude and sign of process gains significantly. For the reactor-separator-recycle system the optimal design cost was $511 600 per annum and had an RGA of 3.5 (i.e., RGA analysis of 1.0 considered optimal). Though highly process dependent, Luyben and Floudas (1994b) demonstrated that the RGA could be reduced from 3.5 to 1.8 by augmenting the process design without increasing costs significantly. However, further improving the controllability to an RGA of 1.1, increased the cost to $1 046 300 per annum. For this specific case study, an improvement in controllability to an RGA of 1.8 lightens the complexity of the control task, though controller design techniques that would allow comfortable operation at a RGA of 3.5 are desirable.

Naturally, the most desired coordinated design approach is one in which the problem statement requires less trade-off between controllability and economic return. Only economic considerations should ideally play a role in the design task. Although controllability is defined by the open-loop response of the process, controller development techniques (i.e., design tools) determine whether the process can be maintained reliably at the economic optimum. Clearly, there are two ways of enhancing the operability at the steady state economic conditions, viz. enhance the process design and improve controller development techniques.

Consider a continuous fermentation process as illustrated in Figure 2-4. The economic objective is to minimise the capital and operating costs. The most economic design must optimise bioreactor volume (i.e., height and diameter), filter duty, valve sizes and piping from both an equipment cost and an operating cost perspective. The raw material cost of the nominal, S_F, and concentrated, S_C, substrate feed and the filtration costs must be included for evaluation. The operating constraints may include cell mass concentration (i.e., oxygen transfer limitations), substrate limiting cell growth, vessel hold-up limits and maximum flow rates of nominal and concentrated substrate feeds. Discrete design decisions may include deciding to use a recycle stream or not. Continuous design decisions relate to bioreactor vessel height and diameter. Random disturbances are expected to exist in the feed rate, feed composition and cell growth kinetics and need to be considered in the process design. Also, a number of parameters in the process model may drift with time, such as the growth kinetics and cost of substrate feeds (i.e., varying economic climate). The coordinated process design and control development is complicated by the open-loop response of the bioreactor at various residence times, dictated by the inherent dynamics of the micro-organisms. As the residence time is decreased the open-loop response changes from open-loop stable, to open-loop unstable eventually exhibiting both hysteresis and chaotic dynamics. A bifurcation analysis revealed that the economic optimal operating point has the most complex dynamics. Multiplicity, i.e. the existence of more than one steady state at the same residence time, destroys the global stability of the set point; thereby much of the power of linear theories. In fact, controllers with integral action may create additional steady state attractors that introduce elements of instability, not originating from the open-loop dynamics of the system (Chang & Chen, 1984). Conceivably, genetic engineering could be used to develop a different micro-organism that did not have such complex dynamic behaviour, but this is clearly outside the scope of conventional chemical engineering process design. For the bioreactor, improvements in controllability necessitate a loss in economic return regardless of the process design parameters such as reactor volume and recycle. While linear control may be effective in open-loop stable operating regions, the economic return is 30 [%] less than at the steady state economic optimum.

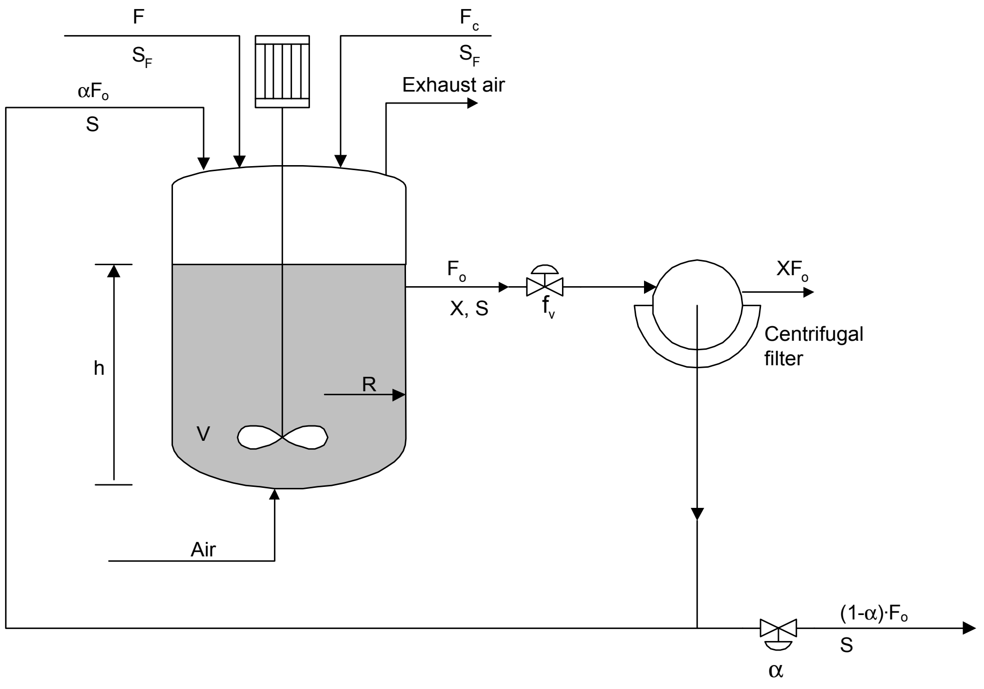


Figure 2-4 - Bioreactor flow sheet (Brengel & Seider, 1992).

Clearly, different process designs result in different control structure designs, related to both the pairing of controlled and manipulated variables and the tuning of controller parameters. Any methodology that solves this integrated process design and controller development task must, (1) be capable of optimising non-linear dynamic systems, (2) ensure robust operation despite unmeasured disturbances and time invariant process uncertainties, (3) select the optimal process design from an economic standpoint and (4) select the optimal control strategy. Both structural (i.e., discrete) and continuous decisions are involved in this optimisation.

A comprehensive algorithmic approach to such an optimisation problem has been demonstrated by Bansal et al. (2002), using Mohideen et al.'s (1996) mixed-integer dynamic optimisation (MIDO) approach. The dynamic distillation model has realistic complexity, comprised of hundreds of differential-algebraic equations. However, the 5x5 control structure design is limited to seven possible alternatives using linear PID controllers in the MIDO analysis. Though PID controllers are the industry standard, linear controllers are likely to have sub-optimal performance when controlling such a non-linear process. The controllability of the process is a function of the plant design, but the type of controller determines feasible operation over a range of disturbances (Bahri et al., 1997). Also, limiting the MIDO optimisation to only seven discrete possibilities for control structure does not assure a global optimal solution. Numerous pairing possibilities exist for a 5x5 dynamic system. Although engineering judgement may reduce the number of feasible control structures, the interface to and from a chemical plant may be via hundreds of sensor measurements and numerous final control elements (i.e. valves, heat duty etc.). The control structure is not a discrete (or integer) optimisation variable and is an important component in a plant-wide control methodology.

2.2 ELEMENTS OF PLANT-WIDE CONTROL

As discussed in section 2.1, the need for a plant-wide approach to process control is chiefly due to the manner in which plants are designed with greater heat integration, recycle and less inventory. However, even without recycle and heat integration, disturbances in upstream units impact on downstream units as all units need to have the same steady state throughput (Larsson and Skogestad, 2000). Luyben et al. (1997) stressed the extreme complexity and open-ended nature of the plant-wide control problem. The problem constitutes a combinatorial number of possible decisions and alternate strategies with no unique "correct" solution. Stephanopoulos (1983) noted that the complex flow of information from the measurements to the manipulated variables may escape systemization.

Systemisation is imperative, since a process facility's control strategy links the plant operation to the business-decision processes, making plant-wide control a key component of business optimisation. The plant-wide control strategy includes increasingly higher layers of business objectives dictated to by market supply-demand cycles. Process down-time due to failures, poor flexibility in moving from one operating mode to another, prolonged operation at sub-optimum conditions, long periods of off-spec production and mismatch between business production requirements and actual production, lead to a reduction in contributed (i.e., added) value. Uncertainties such as unknown disturbances and process/model mismatch require a solution that relies on feedback control structures that are provided with error information for corrective action (Stephanopoulos and Ng, 2000). A perspective beyond single unit operations has been lacking, wherein the plant-wide control problem constitutes a large dynamic optimisation task.

The plant-wide control problem has characteristics, as listed in Table 2-2, not present in the control of individual unit operations. The problem is multi-objective, combinatorial with large numbers of variables and needs to be solved without precise models of the process. The set of controlled variables are not as intuitive as for individual unit operations. Also, local control decisions may have far-reaching consequences on downstream unit operations. Finally, the large number of variables makes finding the global solution difficult.

Table 2-2 - Character of the plant-wide control problem (Stephanopoulos and Ng, 2000).

1. *Multi-objective problem that may entail trade-offs.*
2. *Combinatorial nature in selecting appropriate sub-sets of measured and manipulated variables.*
3. *Imperative information for synthesis may be unavailable or costly to obtain.*
4. *Multi-variable problem of high dimensionality.*
5. *Number of degrees of freedom may be too few to satisfy all operating objectives.*

The difficulties in pairing controlled and manipulated variables and the complexity of the combinatorial problem make multi-loop SISO approaches intractable (Stephanopoulos & Ng, 2000). Robinson et al. (2001) proposed using an optimal control approach to assess whether a decentralised architecture, a MPC architecture or a combination is required for a given plant-wide control problem. Unfortunately, Robinson et al.'s (2001) approach relies on a linear dynamic model that may poorly represent highly non-linear plants. Multivariate control theory seems most suitable to plant-wide control problems. Despite advances in multivariate control system theory, plant-wide control research has remained fixated on classical, multi-loop PID controllers. No regard is given to the poor suitability of such control structures to numerous non-linear, highly interactive processes. Contrary to Skogestad & Larsson (2000), Stephanopoulos and Ng (2000) noted that the use of multi-loop SISO control systems in plant-wide control "lack any significant merit". Robust control (i.e., dealing with model uncertainty), large-scale multivariate controllers such as model predictive control and real-time process optimisation stand to have a profound impact on the design of effective plant-wide control strategies.

Advances in the analysis of controllability have been forthcoming for screening predetermined control structures, but a systematic approach for generating promising control structures remains elusive. In the majority of controller development tasks, the configuration is determined without the use of existing theoretical tools.

Several criteria exist for evaluating control structure design methods. These are (1) generality, (2) applicability to non-linear systems, (3) controller autonomy, (4) quantitative, (5) method simplicity, effectiveness and efficiency and (6) the degree of theoretical development. Few methods could meet even a subset of these requirements (Larsson & Skogestad, 2000).

Conventionally, a unit-based approach is still followed despite greater material and heat integration. A unit-based approach decomposes the plant into individual unit operations. The best control structure for each unit is developed individually. Thereafter, these control structures are combined to complete the plant's control system. Process interactions between individual control structures are eliminated by mutual trial-and-error tuning. Based on the criteria in the preceding paragraph, the unit-based approach lacks general applicability to non-linear systems, is mostly limited to linear multi-loop SISO control structures, lacks efficiency and rests heavily on subjective engineering judgement.

The unit-based approach has largely been replaced by approaches to plant-wide control structure synthesis, typically categorised as mathematically orientated approaches (i.e., optimisation based) or process orientated approaches (i.e. heuristic based) (Larsson & Skogestad, 2000; Robinson et al., 2001). Control structure synthesis is complex to define mathematically (i.e., algorithmically). The number of possible control structures grows exponentially as the degrees of freedom increase. Also, the cost and synthesis of detailed fundamental dynamic and steady state models for evaluation purposes may be prohibitive. Heuristic approaches, based on experience and process knowledge, constitute the norm (Larsson and Skogestad, 2000). Seider et al. (1990) concluded that applying heuristic decision-making should be undertaken with caution for processes with complex physical and chemical behaviours. Morari et al. (1980) affirmed that steady state and dynamic models are required to evaluate interactions and the effects of non-linearities. Therefore, most plant-wide control methodologies include elements of both approaches. Table 2-3 summarises typical strategies to plant-wide control.

Table 2-3 - Typically strategies applied to plant-wide control (Larsson & Skogestad, 2000)

1. *Decentralised control structures based on unit operations.*
2. *Temporal classification by control loop speed, i.e. fast inner loops are closed first, followed by slow outer control loops.*
3. *Hierarchical methods that formalise the synthesis in a top-down manner based on control objectives such as economics, throughput and product purity (active constraints).*
4. *Hierarchical methods based on process structure, i.e. input-output structure, recycle structure, general separation system structure and material/energy interaction.*
5. *Empirical rules.*

The regulatory control tasks are categorised into material balance control and product quality control. Major control concerns relate to: (1) which variables should be controlled (i.e. measured or inferred), (2) which measurements should be paired with which manipulated variables and (3) the nature of the control law (Stephanopoulos, 1983).

Larsson and Skogestad (2000) advocated a mathematical approach to the control structure design task as in Table 2-4 and proposed a top-down design of control objectives (i.e., task 1 to 2) and a bottom-up design of the control system (i.e., tasks 3 to 5). The top-down analysis involves finding a set of controlled variables that maximise the economic return of the process, based on the anticipated disturbances to the process. In this mathematical approach, a degree of freedom analysis is accompanied by economic evaluations using a steady state process model. The bottom-up design may entail a controllability analysis, stabilising control and disturbance rejection through regulatory control. A controllability analysis typically involves linear tools such as the relative gain array, although for non-linear systems bifurcation analyses may provide more process insight. Stabilising control may be established through pole vector analysis. Local disturbance rejection (i.e., partial control techniques) may be accomplished by controlling secondary measurements (i.e., inner control loops) that minimise the effect of disturbances on the primary controlled variables. Decentralised control may be considered should the process be non-interacting and the constraints invariable. For such decentralised control, interactions are minimised by pairing of the primary controlled variables and the manipulated variables using the linear RGA. Multivariate control, such as MPC, improves the performance of interacting processes and for tracking changing active constraints. The supervisory control layer is designed to keep the primary controlled variables at optimal set points using real time optimisation of the operational objective through a steady state model. The industrial approach typically follows the systematic outline in Table 2-4 without using mathematical tools that may be (e.g., Bristol's gain array) available for each task.

Table 2-4 - Control structure design tasks (Morari et al. ,1980)

Control structure elements
1. Select a set of controlled variables to attain a number of specified objectives.
2. Select a set of final control elements (e.g., control valves) that manipulate the selected controlled variables.
3. Select a set of process variables that measure directly the controlled variables or from which the controlled variables may be inferred.
4. Select a control structure connecting measured and manipulated variables for informational flow, i.e. pairing of controlled and manipulated variables.
5. Tune the parameters in the selected control structure, ensuring minimal process interaction between controlled variables.

A heuristic approach sub-divides or decomposes the complexities of plant-wide control synthesis, creating a step-by-step approach that makes the synthesis task manageable. Luyben et al. (1997) proposed nine steps for plant-wide control synthesis listed in Table 2-5.

Table 2-5 - Plant-wide synthesis procedure (Luyben et al., 1997)

1. *Establish control objectives.*
2. *Determine control degrees of freedom by summing the number of independent final control elements (i.e. control valves, electrical heating coils).*
3. *Establish energy management by removing heats of reaction and preventing propagation of thermal disturbances.*
4. *Select control element to control production rate (i.e. throughput). The final control element should have minimal impact on the separation processes, but have a pronounced effect on the reaction rate in the reactor. Reaction rate may be controlled through the reaction temperature, reagent concentrations, reactor hold-up for liquid phase reactions and the reactor pressure in vapour phase reactors. The choice of throughput determines the remaining inventory control system.*
5. *Control product quality, safety, operational, and environmental constraints. For these constraints dynamic relationships between the controlled and manipulated variables should have small time constants, minimal time delay and large steady state gains.*
6. *Control inventories (pressures and levels) and fix a recycle stream flow rate. Inventories should be controlled by final control elements that establish the greatest changes. Liquid recycle flows should be controlled at constant rates to reduce large load disturbances to the separation units and prevent recycling of disturbances. Gas recycle rates should normally be set at maximum circulation rate to attain the highest yields.*
7. *Check component balances by considering a component's steady state composition at a given point in the process. No accumulation is tolerable with purge streams venting inert or undesirable components.*
8. *Control individual unit operations. Effective control strategies have been established for most common unit operations. These historic strategies should be exploited.*
9. *Optimise economics or improve dynamic controllability. After utilising some degrees of freedom to establish the above regulatory control, the remaining degrees of freedom should be used for improving steady state economic return or dynamic control responses.*

Step 3 and 4 in Table 2-5 related to the throughput manipulator, as the reactor is typically the production bottleneck. Price et al. (1994) emphasised the throughput manipulator as the critical element in plant-wide control design. Larsson and Skogestad (2000) recommended careful analysis of the throughput manipulator, as the optimal choice may change depending on the prevailing disturbances. MPC could accommodate such a change in throughput manipulator without using logic configurations as would be necessary for multi-loop SISO. It should also be noted that when using an optimising layer (e.g., NLMPC) that determines the set points of the multi-loop SISO regulatory control, more degrees of freedom are left for economic optimisation (Ricker et al., 1995). From this perspective, though the mathematical and heuristic methodologies deal primarily with the regulatory layer, an optimisation layer is critical to any plant-wide control strategy.

2.3 ECONOMIC AND PROCESS CONTROL OBJECTIVES

Morari et al. (1980) stated that the principal objective of a control system is to translate economic objectives into process control objectives. Likewise, the objective of a chemical plant is the "maximisation of the generated business-wide economic value via plant operation" (Stephanopoulos and Ng, 2000). This generated value is therefore reduced by process upsets owing to sustained operation at sub-optimal conditions or off-spec production, down-time owing to failures, long transition periods from one operating mode to another. The mismatch between the optimal business production strategy and the true plant output must thus be minimised (Stephanopoulos and Ng, 2000). For example, a polyethylene reactor must be capable of producing different polymer grades. A transition from one grade to another may require several hours, which corresponds to a loss in prime production. Though a scheduling concern, transition may be necessary up to twice a week, resulting in financial losses between \$10 000 to \$50 000. Minimising transition losses is compounded by the process non-linearities (Piché et al., 2000).

Optimal operation is defined as the continuous, dynamic optimisation of plant operation using a perfect plant model (i.e., no process/model mismatch), thereby minimising a cost function, J, by adjusting the degrees of freedom. The scalar objective function, J, typically reflects the operating cost. Owing to process model uncertainty and inaccurate sensor inputs, optimal operation is generally not attainable. The discrepancy between the actual value for the cost function J and the global optimum for J, is defined as the process loss (Larsson & Skogestad, 2000). Larsson and Skogestad (2000) defined the concept of an acceptable loss as a process loss resulting from using a fixed vector of set points, without re-optimising in response to process disturbances (or the prevailing market conditions) within a specified time frame. Ideally, process loss should not result due to a trade-off between optimal economic operation and controllability, but only due to model and state information uncertainty. However, separating the optimisation and control layer is frequently necessary, particularly since the time constant of the regulatory control system is usually smaller than the minimum re-optimisation time frame. Provided the control response to the newly re-optimised steady state carries no cost penalty due to a poor response, the economic operation of the plant is determined by the newly calculated steady state operating point (Larsson & Skogestad, 2000).

Zheng et al. (1999) also emphasised the use of economic considerations when making plant-wide control structure decisions. Zheng et al. (1999) noted that the RGA for single unit operations differed significantly from the same unit operations in a plant-wide control scenario. Configurations that worked well for unit operations alone, did not necessary work well in the entire system. In addition, the multiple objectives of plant-wide control problem may not be weighted equally. When there is more than one objective function to be optimised, solutions exists where one objective cannot be further improved without sacrificing performance in other objectives. Such solutions are defined Pareto optimal and the set of all Pareto optimal solutions form the Pareto front. Generating a Pareto front based on varying weighted objectives may prove valuable, though computationally intensive. For example, most plant-wide control challenges require that a cost function, *f*, be minimised within a product purity equality constraint, *h*, as in equation 2-4:

$$\min_{\underline{x},\underline{p},\underline{u}} \phi = \omega_1 \cdot f\left(\bar{x},\bar{u}\right) + \omega_2 \cdot h\left(\bar{x},\bar{u}\right) \qquad \textbf{(2-4)}$$

Within the specified range for product purity, the unit cost or added value may vary dramatically for highly non-linear processes. Successively relaxing the weight, ω_2, on the product purity, generates a Pareto front. Without a full Pareto analysis, operating objectives must be ranked based on their total economic impact and thereby weighted accordingly to allow for calculation of a single scalar cost function (Stephanopoulos and Ng, 2000).

The production objectives of any process may vary significantly between two classes of economic objectives, viz. maximum production (seller's market) and lowest possible unit cost (buyer's market). Translating economic objectives into process control objectives entails finding a function of the process variables (i.e., state variable representations) in terms of the manipulated variables, which moves the state trajectory along an optimal path. Each possible operating region in the state space has an innate economic value. This may entail keeping a set of process variables constant in the absence of disturbances. However, during disturbances the control system should attempt to track the optimal economic trajectory as it relates to the state space. The overall operational (i.e., the economic objective) may be the minimisation of a scalar cost function, subject to operational constraints such as product purity, safety (e.g., maximum vessel pressures before mechanical failure) and controllability. A control system's robustness is determined by the controllability in the operating region of highest economic return (Morari et al., 1980).

Morari et al.'s (1980) "Optimising Feedback Control Structure" provides the essential framework in response to Foss' (1973) process control critique. The absence of applicable techniques for solving the plant-wide control problem has been attributed to the absence of a mathematical formulation and a clear statement of the objectives (Morari et al., 1980). The main goal of a control system design is to create a dynamic structure of process (i.e., measured) and manipulated variables, which meet production objectives continuously. Several process variables must be guided from an undesired state to a desired state, with an appropriate response during plant disturbances. Operational objectives vary based on management strategies determined by present and future economic outlooks. The optimal operating conditions change with the external disturbances and drifting process kinetics. Industrial practice shows that a changing production policy is feasible, though shifting the operation from one set of process conditions to another may require a change in the control structure. A fixed control structure may not assure a smooth transition from one operating region to another for better economic return (Morari et al., 1980).

In modern day real-time optimisation systems, the control system should be selected that yields the highest profit for a range of disturbances that may occur between each optimisation of the set point values (Skogestad, 2000a). In conventional control system development this implies finding a fixed control structure that is robust for a variety of disturbances, with only the set points and controller parameters changing with successive optimisations and adaptive control adjustments. A fluid control structure that changes along with the changes in the process and market conditions is more desirable.

2.4 SELECTION OF CONTROLLED VARIABLES

Skogestad (2000a) presented a hierarchical six-step method for selecting candidate control variables. The first five steps encompass the problem definition, while the sixth step entails optimising the control structure based on the problem scope outlined in the first five steps. The problem definition involves a degrees of freedom analysis, cost function specification, defining operating constraints, steady state optimisation, identifying disturbances, and identifying candidate controlled variables. Thereafter, the economic loss is evaluated for the alternate sets of controlled variables in the presence of the identified disturbances. Sections 2.4.1 to 2.4.4 outline the problem definition, while section 2.4.5 pertains to the solution methodology.

2.4.1 Degrees of freedom analysis

In order to keep the plant operating at the desired steady state, the available degrees of freedom must be specified. With all the degrees of freedom fixed and provided the process does not have multiple steady states (i.e., exhibit multiplicity), the remaining process variables are uniquely determined. Any sub-set of the available process variables could conceivably be selected as controlled variables. However, disturbances and sensor noise introduce uncertainty. Certain process variables respond more readily to disturbances that shift the operating point away from the economic optimum. The final control elements must be used effectively to drive the process back to the optimal steady state in the presence of disturbances, sensor noise and model uncertainty. The control system's response to these process uncertainties depends on the set of controlled variables. Hence an optimal set of controlled variables exists, depending on the prevailing disturbances and process conditions (Larsson & Skogestad, 2000). The selection of controlled variables should be made based on the overall operational objective; thereby the economic objectives must be translated into process control objectives.

The number of degrees of freedom for control, N_m, is apparent from the number of available manipulated variables (i.e., valves, electrical and mechanical final control elements). The number of manipulated variables for maintaining the optimum operating condition is typically less than N_m. N_{opt} is equal to the number of manipulated variables, N_m, less the number of manipulated variables, N_{m0}, that have no effect on the cost function, J. For example, the minimum agitation speed may provide uniform mixing and further increasing the agitation speed has no effect on the mixing performance or unit cost. Furthermore, a sub-set of the N_{opt} manipulated variables must be used to satisfy all active constraints (e.g., product purity) with the remainder providing for optimal operation of the unconstrained process variables. Therefore, a degrees of freedom analysis determines the minimum number of controlled variables that need to be selected for the control system structure.

2.4.2 Cost function, constraints and optimisation

A mathematical formulation of the plant-wide control problem allows for quantitative analysis while resorting to minimal engineering judgement. The plant-wide control problem may be expressed as a scalar objective function formulated on the economics of the process, as the economics relate to the process' state space. This cost function is typically subject to operational constraints that complete the non-linear programming definition as in equations 2.1 - 2.3.

As in Ricker (1995), the steady state economic optimum needs to be found based on this scalar objective function. From 1970 to 1990, two approaches have been used, almost exclusively, to locate the optimal process steady state; viz. a reduced gradient algorithm as implemented in the MINOS program (later versions included a projected augmented Lagrangian strategy) and successive quadratic programming (SQP) as implemented in the OPT program. For highly non-linear systems, these optimisation algorithms have difficulty in converging to the global optimum (Seider et al., 1990). More effective optimisation strategies are based on the Newton homotopy-continuation algorithm for solving non-linear equations (Brengel & Seider, 1992). Particularly, evolutionary algorithms such a genetic algorithms offer unique opportunities to finding the global optimum for highly non-linear processes.

The open loop dynamics of a non-linear process may vary considerably depending on the operating conditions. Even though the optimisation routine may have located the global optimum, complex open loop dynamics may prevent linear controllers from providing robust control during disturbances. A sub-optimal steady state may need to be selected as the operating point to accommodate the limitations of available controller development techniques. Analysing the impact of disturbances at the operating point using the cost function, provides a quantitative measure for selecting controlled variables, though disturbance identification remains a difficult task.

2.4.3 Disturbance classification

Disturbance classification is an important step in developing a control structure, as disturbances carry economic implications and therefore require appropriate control action (Morari et al., 1980). Disturbances may include (1) mismatch between the actual process and the model used for optimisation, (2) operational disturbances caused by process upsets and (3) implementation errors owing to measurement noise or poor control (Skogestad, 2000a). Although the global optimum of the process model may have been found, process/model mismatch introduces a set point error akin to an operational disturbance. Operational disturbances are common, since feed compositions, physical properties and feed supply rates may vary significantly. Sensor noise (particularly complex sensor technologies such as biosensors) and poorly tuned controllers introduce artificial disturbances that may affect the economic return adversely.

Typically a disturbance is defined as any process occurrence that causes a deviation from set point. However, a deviation from set point does not necessarily imply reduced economic return. For example, the substrate feed concentration to a bioreactor may vary considerably, but substrate feed concentrations above the nominal concentration may allow for greater economic return, provided the control system takes advantage of such a disturbance (Brengel & Seider, 1992). The definition of a disturbance needs to be broadened.

Skogestad (2000a) distinguished between a set point error and an implementation error in a closed loop where the process variable, C, is maintained at C_s. The set point error, e_{cs}, is the difference between the optimal value, C_{opt}, and the set point, C_s. The implementation error, d_c, is the difference between C and C_s. Disturbances and changes in raw material costs induce set point error, while poor control and sensor noise bring about implementation error. Using these two definitions, the overall error, e_c, is defined as the sum of e_{cs} and d_c.

Frequently, chemical plants are exposed to disturbances of unknown origin, which may not be identifiable or easily measured (Stephanopoulos, 1983). Disturbance identification is further complicated by uncertainty related to the frequency, magnitude or type (i.e., gaussian, step, ramp or white noise) of a given disturbance. However, identified or expected disturbances should be classified as non-stationary and stationary. Non-stationary disturbances are those that vary slowly with time, such as catalyst deactivation. Stationary disturbances are classified as those having a fast and local effect on the economic objective function. The disturbance classification introduces a temporal control hierarchy with two time scales. Economically significant non-stationary disturbances are candidates for optimising control, while most stationary disturbances are rejected by regulatory control. The time horizon for regulation is thus shorter than for optimisation. The regulation task maintains the process around a set of controlled variables, while the higher layer determines the optimal set points for those controlled variables in response to non-stationary process disturbances (Morari et al., 1980).

Once the expected disturbances to the process have been identified and classified, the impact of these disturbances on the cost function needs to be assessed for alternate sets of candidate controlled variables (Skogestad, 2000a).

2.4.4 Determine alternate sets of candidate controlled variables

Skogestad (2000a) proposed a largely qualitative approach to selecting alternate sets of candidate controlled variables from the large number of possible combinations. This approach entails satisfying the active constraints (section 2.4.4.1), heuristic dominant variables (section 2.4.4.2), and process insight and guidelines as proposed by Luyben et al. (1997) in Table 2-5. These alternate sets are evaluated in more detail using an evaluation of loss analysis as discussed in section 2.4.5.

2.4.4.1 Operating constraints

As discussed in section 2.4.1, a minimum number of controlled variables need to be selected from the total number of process variables to fulfil operational constraints. Figure 2-5 illustrates the concepts of an active constraint, an unconstrained flat optimum and an unconstrained sharp optimum at steady state conditions. Figure 2-5a shows the cost function at the minimum when the controlled variable is at its high constraint (e.g, owing to safety considerations). Frequent set point changes are typically unnecessary for such constrained controlled variables, since disturbances and changing market conditions rarely shift the optimal value away from the active constraint. For example, the conversion in a chemical reactor may continually increase as the pressure is increased (function of the reaction kinetics). Regardless of disturbances or changes in the cost of raw materials, the optimum reactor pressure remains at the highest safe operating pressure. However, operation near such safety constraints may be complicated by poor control. Although operation just below the safety valve lifting pressure may be more profitable, poor control may require a large safe operating margin, carrying a large economic penalty. However, product purity constraints are also active constraints and may require frequent set point changes based on market demands.

Figure 2-5b shows an unconstrained controlled variable where the cost is insensitive to the steady state value of the controlled variable. Such controlled variables may be vessel levels (i.e., hold-up), which have minimal steady state effect on the operating cost. Such vessel levels may be allowed to float above a minimum level, alternately the level becomes a controlled variable to maintain material inventory control. Figure 2-5c shows a sharp unconstrained controlled variable, such as the reaction temperature in a chemical reactor, where any deviation to a steady state other than the optimal temperature dramatically increases the operating cost.

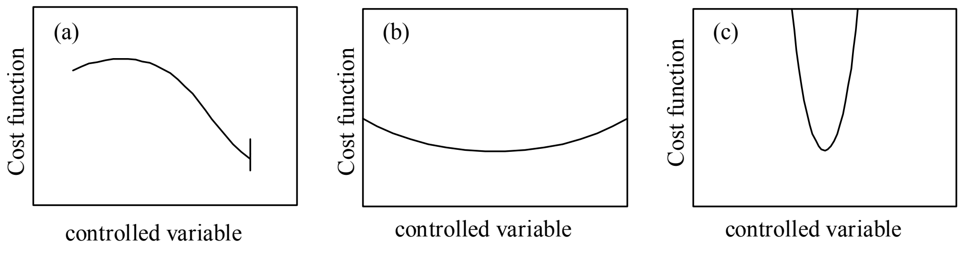


Figure 2-5 - Cost function relationship with controlled variable (Skogestad, 2000a).

The viable controlled variable candidates from the set of process variables need to be chosen for the manipulated variables, N_{opt}. Skogestad (2000a) considered square closed loop systems where $N_c = N_{opt}$. However, particularly for disturbance rejection, selecting more control inputs than strictly necessary provides some feedforward control in the control structure. Constrained variables are mostly automatic candidates as controlled variables. Constraints may be product specifications (e.g. minimum purity), manipulated variable constraints (e.g., non-zero flows) and operational (e.g., equipment limits to throughput) or safety limits (e.g., maximum pressure). A minimum purity constraint (i.e., active constraint) makes the composition a clear controlled variable. Similarly, should pressure constraints exist for safety reasons, such a process variable should be controlled. Thereafter, the remaining unconstrained process variables are subjected to the engineering insight and experience criteria in Table 2-6, which determines their suitability for inclusion as controlled variables (Skogestad, 2000a & 2000b).

Table 2-6 - Controlled variable requirements.

Controlled variable requirements
1. The optimal values of the controlled variables should be insensitive to disturbances, ensuring that the set point error, as defined by Skogestad (2000a), is small.
2. Measurement ease and control accuracy. Unreliable and infrequent measurement complicates the control task.
3. A controlled variable should be sensitive to manipulated variable changes. The gain between the manipulated variable and the controlled variable should be large.
4. For multivariable systems, the selected variables should not be highly correlated. In other words, the controlled variables should be the state variables of the process.

Requirement 1 in Table 2-6 ensures that operation remains near-optimal between successive optimisations of the controlled variable set points. Requirement 1 also prevents significant changes to set points that cater for linear controller implementation. The implementation error is not exasperated by continually changing set points. For PID control, small implementation errors from set point prevent saturation of the manipulated variables. Also, the linearisation of non-linear process models around the set point of the controlled variable should remain more valid for small implementation errors. Nonetheless, process disturbances need to be detected through the controlled variables as early as possible, thereby allowing timely corrective (feedback) control actions (Skogestad, 2000a). Stephanopoulos and Ng (2000) advocated a departure from linear control structures and therefore proposed that controlled variables be selected based on their high sensitivity to disturbances that have a significant impact on the operating profit. Furthermore, where L is the opportunity cost or loss due to set point errors, $(\partial L/\partial d)_c$ should clearly not be minimised at the cost of a higher L. Operation at the global economic optimum must be assured.

Ideally, criteria 1 and 2 in Table 2-6 should not be achieved at the expense of operating at the global economic optimal. For example, the global optimum for the bioreactor state variables in Figure 2-4 is highly sensitive to disturbances and control is complicated by chaotic dynamics. Less sensitivity to disturbances and greater control accuracy is possible at local optimum operating points, but at a significant economic cost. In other words, a small implementation error should not be accompanied by a large set point error, where the set point error is always defined from the global optimum. For highly non-linear systems with linear control systems, adhering to criteria 1 and 2 may necessitate a large overall error, though the overall error has been minimised within the performance limits of the linear control system. Criteria 3 and 4 in Table 2-6 relate to the appropriate pairing of controlled variables and manipulated variables, thereby avoiding large changes in final control elements to disturbances and avoiding process interaction. Both criteria 3 and 4 are more important to multi-loop PID systems than to MIMO control systems.

Adhering to the four criteria in Table 2-6, requires drawing enormously from engineering insight and experience. In addition, the concept of dominant variables may prove useful in selecting alternate candidate sets of controlled variables.

2.4.4.2 Process insight and dominant process variables

As discussed in section 2.3, the search for the optimal set of controlled variables may be a large combinatorial problem. A candidate set of controlled variables is selected from the total number of process variables. The economic performance of a process facility is determined primarily by the steady state condition. Process variables that have no steady state effect (i.e., no economic consequence) are removed from the total set of process variables that may be selected as candidate controlled variables. Consequently, the dimension of the combinatorial problem is reduced by eliminating the liquid levels in separator or buffer tanks that typically have no steady state effect (Skogestad, 2000a). The level in a vessel may have a steady state effect, should the level determine the available heat transfer area. The residence time, determined by the liquid level, in reaction vessels may also have a profound effect on the control strategy at steady state. Some process variables may thus be eliminated from contention as possible candidate controlled variables based on engineering judgement and heuristics.

The vector of primary or economic performance variables, Y_P, defines the product specifications and the process conditions (Larsson & Skogestad, 2000). Equation 2-5 describes a linear dynamic system with three state variables and one input variable:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \cdot u_1 \qquad \textbf{(2-5)}$$

where, x_n is a state variable, A_{mn} are the elements of the state matrix, B_m are the elements of the input matrix and u_m is the input. In other words, the primary process variables are the state variables, i.e. the smallest set of variables that together with the input (i.e., manipulated) variables uniquely determine the steady state and dynamic behaviour of the process. The state variables are often not measured directly, owing to unavailable sensor technology, a lack reliable sensor technology or sensor expense. For example, the temperature in a binary distillation column is an inexpensive, reliable means of inferring the concentration in the product. The state variables are related to the measured variables so that:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \\ C_{41} & C_{42} & C_{43} \\ C_{51} & C_{52} & C_{53} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \textbf{(2-6)}$$

where, y_m are the measured process variables and C_{mn} is the input matrix. Note that the total number of measured variables, Y_d, may be greater or less than the number of state variables. Typically, a control system only utilises a sub-vector, Y_{cd}, of the total available process variables. The control objective is thus to maintain Y_p at the optimal operating conditions by keeping Y_{cd} at fixed values by manipulating the input vector U_d. A process variable vector Y_{cd} thus needs to be matched with U_d, ensuring maximum compensating effect on Y_p.

The dominant variables are those process variables that have the greatest influence on the dynamics of the process, e.g. the reaction temperature in a chemical reactor. Dominant process variables are invariably those process measurements that most accurately represent the dominant state variables. For example, in equation 2-6, y_1 and y_2 may be dominant process variables with the state variables x_i having lesser representation in y_3 - y_5.

Conceptually, dominant state variables may be explained by considering dynamic model reduction and state estimation studies. The Tennessee Eastman challenge process is a rigorous dynamic model of a plant-wide industial process. The process dynamics is described completely by 50 state variables and 12 manipulated variables. However, Ricker & Lee (1995b) showed that 26 state variables and 10 manipulated variables are sufficient for estimation of the dominant dynamic and steady state responses. Assumably, a large number of the 50 state variables in the complete model are required for describing minor transient and steady state behaviour.

Partial control implies the use of only process variables that represent the dominant state variables, thereby obtaining acceptable control of the other state variables that may be unknown or immeasurable. Larsson & Skogestad (2000) doubted the mathematical usefulness of this concept, since no explicit mathematical procedure exists for identifying such dominant variables. Partial control has historically been approached in a heuristic manner. Clearly, biological control systems (i.e., organisms) are highly reliant on identifying sensori-input variables (e.g., shape, colour, sound, smell etc.) that represent the dominant state variables on a given moment or situation, using these sensori-input variables to constructing cause-effect models of their environment and act accordingly.

Nevertheless, Skogestad (2000a) emphasised that insight and experience of dominant process variables remains vital, since the number of candidate process variables may be large and the possible combinations prohibitive for timely analysis. For systems with many process variables, selecting alternate sets of candidate controlled variables is non-trivial. Though the guidelines by Skogestad (2000a) are useful, Table 2-6 and the heuristic approach of assessing dominant process variables offer no systematic means of selecting a number of alternate sets of candidate controlled variables.

Ideally, each possible set of candidate controlled variables should undergo a loss evaluation, which assesses the penalty (economic loss) for fixing their set points during known disturbances. It is conceivable that the even after meticulous loss analyses of a few alternate sets selected on engineering judgement, the optimal set of controlled variables may not have been considered.

Also, process dynamics have not been considered, such as controllability. Instead, Skogestad (2000a) proposed using only a steady state model to assess the economic loss associated with identified or known disturbances. However, most disturbances in chemical processes remain unmeasured and unidentified, owing to complex causal relationships; otherwise feedforward control could be used to improve disturbance rejection.

Stephanopoulos and Ng (2000) emphasised that the selection of controlled variables cannot be determined without considering the impact of disturbances and model uncertainties with the optimal controller design simultaneously. Stephanopoulos and Ng (2000) concluded that a plant-wide control configuration entails essentially a design of a large multi-variable controller, assumably using a large number or all of the available process variables, in the presence of uncertainty. Stephanopoulos and Ng (2000) further concluded that process non-linearities and high dimensionality require sophisticated methods outside the realm of linear control theory. Separating the control structure selection, i.e. the controlled and manipulated variable selection, from the controller design necessitates an approach with numerous iterations.

Following the hierarchical procedure by Skogestad (2000a), once the alternate sets of candidate controlled variables have been determined, these sets of controlled variables are subjected to an evaluation of loss.

2.4.5 Compile the evaluation of loss

Although existing control theories, particularly linear control theory, are mature in devising control laws from linear models, these theories offer minimal assistance to selecting the optimal vector of controlled variables for such control law specification. The primary difference between the mathematically orientated approach and the heuristic approach rests in the selection of the controlled variables. The mathematical approach relies on formalised optimisation of the control objectives as these objectives relate to process economics and the impact that disturbances have on the process' economic return. The heuristic approach relies primarily on engineering judgement and guidelines related to process structure (see Table 2-5) for determining the set of controlled variables.

Morari et al. (1980) and Larsson & Skogestad (2000) stressed an economic approach so that the steady state sensitivity of the economic loss, L, to disturbances is minimised $(\partial L/\partial d)_c$. The set of controlled variables that yields the lowest profit loss for a range of prevailing disturbances should be selected.

The loss, L, is the difference between the actual operating cost using a specified control strategy and the true minimum operating cost, i.e. $L = J - J_{opt}$. For multi-loop PID control, the least complex control strategy involves maintaining all controlled variables at constant set points. Thereby, the complex on-line dynamic optimisation is reduced to a more straightforward feedback problem. Self-optimisation control is achieved should such a constant set point strategy result in acceptable operation or loss, without re-optimisation during disturbances (Skogestad, 2000a).

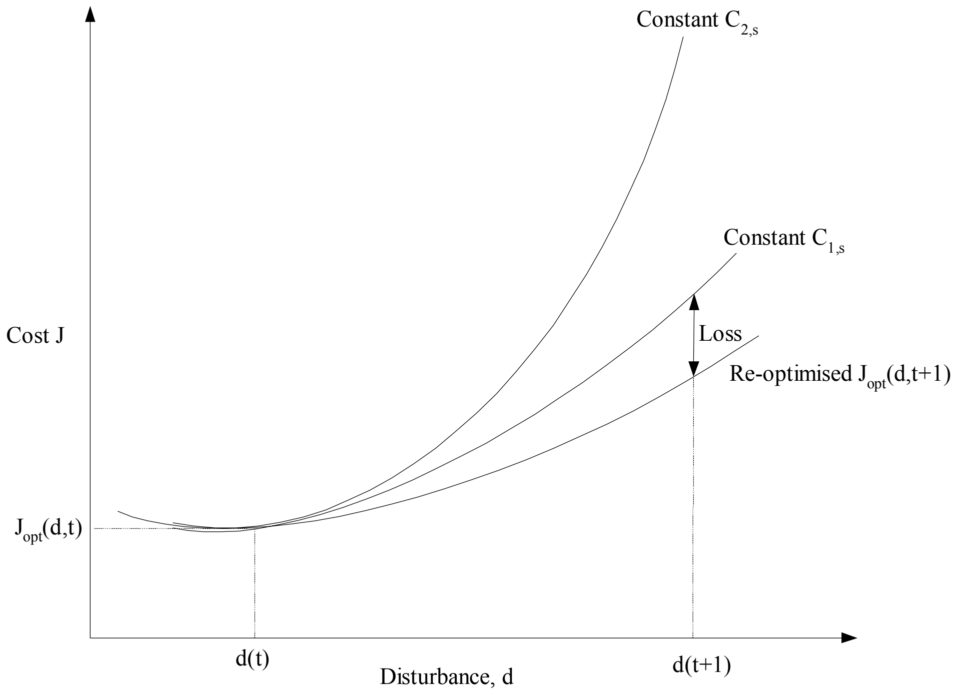


Figure 2-6 - Economic loss induced by not re-optimising the set point vector for the controlled variables in response to a change in a particular disturbance d. Using the set of controlled variables, $C_{2,s}$, results in a large economic penalty as compared to the more economic loss of keeping $C_{1,s}$ constant (Skogestad, 2000a).

For each evaluation, the set points are typically fixed at the nominal optimum and the effect of each identified disturbances is recorded. In Figure 2-6, assume that the nominal optimum has the operating cost, $J_{opt}(d,t)$ with no disturbances at *d(t)*. Introducing a single disturbance at *d(t+1)*, while keeping $C_{1,s}$ constant has an associated self-optimising loss. Note that keeping an alternate set $C_{2,s}$ constant has a far larger self-optimising loss than $C_{1,s}$ for this individual disturbance. The mean loss, L_{mean}, is computed for alternate sets of controlled variables, $C_{i,s}$, by averaging the loss for each identified disturbance. Based on the number of identified disturbances, the evaluation of loss is weighted in the average.

As noted in section 2.4.3, disturbances should not always be classified as detrimental to the process economics. Averaging the effect of disturbances could mask the negative effect of individual disturbances. In addition, disturbances may have interacting dynamics, whereby two simultaneous disturbances result in far greater economic loss than the sum of the economic loss associated with each individual disturbance. In this regard, even a finite set of disturbance combinations may not be sufficient to gauge the full impact of the identified disturbances. Also, this loss analysis assumes that the process engineer has knowledge of all possible prevailing disturbances. In the wider process design framework, weighting the effect of disturbances gives no hint as to how the process may be changed to better reject a particular disturbance. Also, it remains unclear as to how the above evaluation of loss is calculated should the process be open-loop unstable at the economic optimum.

In general, as is apparent from the discussion in section 2.4, the controlled variable selection process relies on critical decisions that are frequently made with no quantitative justification, resorting to subjective decision-making. Operational objectives are translated into a set of desired control variables by a guiding set of principles that has no assurance of finding the global optimum.

2.5 SELECTION OF MEASURED AND MANIPULATED VARIABLES

Typically, only a sub-set of the available process measurements and manipulated variables are used in feedback control synthesis. As in section 2.4, the vectors of potential measurements and manipulated variables are largely determined by the impact of disturbances on the operational objectives. The set of manipulated variables dictates the ability of the control system to maintain the control objectives (Morari et al., 1980).

Measurements aim to monitor the selected controlled variables either directly or inferentially. The key criterion for measurement selection pertains to observability of all controlled variables. An uncertainty analysis should also accompany the measurement selection, viz. sensor noise, measurement delay and incorrect inference of the actual controlled variables (Stephanopoulos and Ng, 2000).

Stephanopoulos and Ng (2000) contented that all available manipulated variables should be included in the control of the process in a large model predictive control implementation. Exceptions are listed in Table 2-7. Also, a number of manipulated variables may have no economic impact regardless of being varied. For example, an agitation system may provide uniform mixing at the lowest agitation speed, meaning that the system has been over-designed by including a variable speed drive.

Table 2-7 - Exceptions to using all the manipulated variables (Skogestad & Ng, 2000)

1. *A large multi-variable controller is infeasible due to high dimensionality or significant modelling uncertainties.*
2. *A decentralised or hierarchical approach with PID controllers is preferred, requiring appropriate pairing of the measured and manipulated variables.*
3. *Sensor failure is likely or possesses a significant risk to the control strategy.*

Naturally, the chosen set of measurements and manipulated variables is highly correlated with the control system's performance. Therefore, the selection of the optimal measurement-manipulated variable structure must be considered in unison with the quantitative evaluation of the control system. . For multi-loop SISO systems, process interactions determine feasible sets of manipulated variables for maintaining the controlled variables. Process controllability, which implies an analysis of interactions between variables, may be assessed with linear techniques such as the RGA, singular values and singular vectors (Larsson & Skogestad, 2000). These techniques may have limited application to structuring a large multi-variable control task (Ricker, 1995).

2.6 CONTROLLER CONFIGURATION AND CONTROL LAW

Once the controlled variables that best translate the economic objectives to control objectives have been identified, the synthesis of information flow from measured to manipulated variables follows. This may be a non-trivial task should significant overlapping interdependencies exist between measured and manipulated variables. It is important to understand that the methodology in section 2.4 only selects a possible optimal set of controlled variables, but does not suggest how these controlled variables should be paired with the available manipulated variables. Incomplete information compounds the synthesis process requiring robust solutions to uncertainties. A sensitivity analysis is paramount to ensure robust performance (Stephanopoulos, 1983). Morari et al. (1980) stated that it is desirable to define a control structure with the lowest degree of complexity necessary to accomplish the control task. Given the unavoidable mismatch between the actual process and the process model, the "lowest control structure complexity" should rather be substituted with the most pliable structure (i.e., affording plasticity) to account for uncertainty.

Plant-wide control structures have ranged from decentralised PID to MPC strategies. Should the performance of a decentralised PID strategy compare to a MPC implementation, industry will necessarily adopt the decentralised PID strategy. Structural simplicity and operator acceptance favour a decentralised PID strategy. Conversely, should the MPC control offer substantial performance improvements, economic considerations outweigh historic preference (Robinson et al., 2001). PID strategies serve as a reduced model to a plant-wide control problem, owing to the perceived simplicity of design and reduced engineering time.

Decentralised control is motivated by a decomposed control systems that may be less prone to model uncertainty. Process judgement imposes a certain control configuration, which explicitly provides process information. A centralised controller (e.g., MPC) would need to obtain such process information from a dynamic process model. However, process judgement is frequently subjective and non-intuitive process behaviour may result in incorrect control configurations, which impact the rest of the control design. Furthermore, efficient decoupling of decentralised SISO controllers requires a dynamic model, otherwise decoupling is also based on trial-and-error tuning. Decentralised control is naturally preferred for non-interacting processes, but few chemical plants are non-interacting. Multivariable control (e.g., MPC) improves the control performance of interacting processes dramatically and provide robust tracking of moving constraints (Larsson & Skogestad, 2000).

Despite the clear benefit of a non-linear, multivariate approach to plant-wide control, research on plant-wide control has remained confined to linear multi-loop SISO control structures (e.g. simple PID control, cascade control and ratio control). Nevertheless, considerable advances in the industrial implementation of large model predictive and other multivariable control systems have been made. Simple multivariable model predictive control structures overcome the pairing and tuning complexities arising from multi-loop SISO control implementations. Control methodologies such as model predictive control, robust control and real-time process optimisation reduce the dependency on multi-loop SISO controllers.

Once the controlled variables and manipulated variables have been selected, the regulatory and supervisory control layers are designed. The regulatory layer has a stabilising and disturbance rejection function. The regulatory design addresses controllability such as the pairing of controlled and manipulated variables. Bristol's relative gain array may have limited use for highly non-linear processes. Most control configurations are comprised of nested control loops, which make use of secondary measurements. Secondary measurements are used in fast inner control loops, where local disturbances are rejected before these disturbances impact on the slower outer control loop. Secondary measurements should be selected so that updates to their set points by the outer loops are minimal. (Larsson & Skogestad, 2000).

Table 2-8 - Methodology proposed by Price et al. (1994)

Hierarchical decomposition on control objectives (Price et al. (1994)
1. Inventory and production rate control
2. Product purity control
3. Operating and equipment constraints
4. Improved economic performance

Selecting the controlled variables in section 2.4 involved a top-down approach, whereas the control system design is a bottom-up approach. Plant-wide control departs from the unit operation based approach, which has proven unsuccessful for plants with significant material recycle and heat integration. Plant-wide control follows a hierarchical decomposition based on either process structure or throughput, control objectives or time scales. Luyben et al. (1997) proposed the approach in Table 2-5.

Price et al. (1994) proposed the methodology as outlined in Table 2-8, which first stabilises the process and determines the throughput manipulator, after which the production specifications are addressed. McAvoy and Ye (1994) based their decomposition on time scales. McAvoy and Ye (1994) proposed a four stage method as described in Table 2-9. Steps 1-3 are based on control loop speeds. Step 1 rejects local disturbances, while step 2 involves screening with controllability tools such as RGA.

The selection of the sets of controlled and manipulated variables is unavoidably related to the performance of the final control system. The selection of the input-output connections, that form the control loops, must be considered simultaneously with the quantitative evaluation of the developed control system. Zheng et al. (1999) evaluated a large number of different control structures, control laws and tuning parameters based on the integral absolute error of the product purity during process disturbances. Owing to this combinatorial problem, Zheng et al. (1999) highlighted the difficulty in selecting a set of controlled variables and emphasised the lack of an efficient method.

Theoretical considerations in pairing controlled and manipulated variables rest on controllability measures such as the relative gain array (RGA), singular values and the condition number along with singular vectors. The benefit of these controllability techniques has been evident for unit operations, but less so for effective structuring of plant-wide control systems (Stephanopoulos & Ng, 2000). Stephanopoulos & Ng (2000) proposed using the modular multivariable controller design that selects the best set of controlled variables based on (1) each manipulated variable's effect on the controlled variable (i.e., analysis of the open-loop gain), (2) the model uncertainty and (3) the non-minimum phase behaviour of the input-output relationships.

Table 2-9 - Hierarchical decomposition based on time scales.

Hierarchical decomposition based on time scales (McAvoy & Ye, 1994)
1. Design inner cascade loops
2. Design basic decentralised loops, other than the control loops associated with purity and production rate.
3. Production and purity control loops.
4. Higher layer controls.

Once a control structure has been designed, real-time optimisation, using steady state models and operational objectives, should update the set points for the controlled variables (Larsson and Skogestad, 2000). Particularly for large disturbances and shifting market conditions, the new set points may be located far from the original operating region. The PID tuning parameters in this new operating region may also need revision for highly non-linear processes. The final process control system must be validated as an important final step through non-linear dynamic simulation (Larsson and Skogestad, 2000).

2.7 CONCLUDING REMARKS

As discussed in the following two chapters, evolutionary reinforcement learning (ERL) circumvents the use of heuristic methods in plant-wide control designs. Though not algorithmic, or strictly mathematical in nature, evolutionary reinforcement learning offers real opportunities. ERL does not require identification of major disturbances or a throughput manipulator, which may be non-trivial tasks. ERL implicitly identifies the necessary controlled variables and implicitly pairs the controlled variables with the available manipulated variables in a non-linear multivariate controller.

Self-optimisation, as defined by Skogestad (2000a), appears plagued by heuristic considerations that may not produce the desired result. ERL methods, such as symbiotic memetic neuro-evolution (SMNE) also seek self-optimising control but aim to achieve this via neural network generalisation, which requires no explicit disturbance identification. The efficient generalisation of neural networks to novel process conditions, typically allows robust performance despite significant disturbances. Where robust generalisation is not achievable, adaptive control via adaptive neural swarming (ANS) provides for on-line adaptation of neural network weights to track a changing economic optimum.

Chapter 3 introduces introductory concepts for efficient reinforcement learning. Chapter 4 presents the neurocontrol algorithms SMNE and ANS that form the cornerstone of a plant-wide neurocontrol strategy.

3 EVOLUTIONARY REINFORCEMENT LEARNING

OBJECTIVES OF CHAPTER 3
• Highlight the complexities that need to be addressed for effective reinforcement learning in a dynamic environment. These elements of reinforcement learning are discussed in the following sections.

3.1 AVAILABLE SOURCES OF INFORMATION FOR CONTROL SYSTEM DESIGN

In some control tasks a human operator provides the feedback control action, especially where it has been difficult to design an automatic controller using standard control techniques. It may have been impractical to obtain an analytical model of the controlled system. For example, industrial flotation control is typically operated in the open-loop, where experienced human operators make control decisions based on visual input such as froth colour and froth bubble size. Given an experienced human operator, supervisory control, where an automatic controller mimics the actions of a human operator, becomes possible. A neural network or an expert system may provide the knowledge representation and required control formalism. Training such a neural network involves providing the network with the same sensory input information that a human operator may receive. During training, the neural network targets outputs that correspond to a human's control input to the process. Similarly, expert systems rely on gathering process knowledge from plant operators and formalising this knowledge in fuzzy rules. Such training techniques are deemed supervised learning, where the error information from a desired target output is used to adjust the neural network weights.

Obtaining the set of input vectors that correspond to desired target output vectors becomes a major control challenge. The ability to obtain supervisory information from operators is limited by the operator's ability to communicate such information. For example, a flotation operator (regardless of articulation) may not be consciously aware of what visual state information he extracts from the froth surface. He may agree with an interviewer that bubble shape, size and froth colour are important considerations, not being consciously aware that he also considers bubble surface velocity in determining the control action. Also, the accuracy of such information may be questionable. Information from one operator to another may be contradictory, requiring reconciliation of expert knowledge. As knowledge acquisition in the design of control expert systems has proven to be a bottleneck, research in machine learning has attempted to compensate by broadening the base of accessible sources of knowledge (Greffenstette et al., 1990).

In rare instances a detailed fundamental non-linear dynamic model is available. Uncertainty regarding the applicability of the linearisation for formal linear control design may require a more robust control approach. As described in chapter 2, the appropriate plant-wide control configuration using multi-loop PID controllers may not be determined easily. Alternately, process information for control purposes may also be obtained by inverting such rigorous process models. This approach also fits into a supervised learning guise. This approach relies significantly on the fidelity of the inverse model used as the controller. The robustness provided by the training algorithm is paramount. A lack of robustness has primarily been attributed to a lack of error feedback to the controller. Static inverse neural networks have been prone to steady-state off-set. On-line learning may alleviate this robustness concern, by adjusting the inverse of the model on-line (Hunt et al., 1992). In addition, although a causal relationship exists been the input and output plant data, there is no assurance that the inverse dynamics has a corresponding unique relationship (Krishnapura and Jutan, 2000).

Kim et al. (1997) also alluded to learning stability difficulties and a lack of robustness that arises in the inverse control scheme. Chemical plants may be operated at high pressure and/or temperature, thus the control scheme must guarantee closed-loop stability. To improve robustness, a feedback controller may be included in the direct inverse control scheme. Furthermore, Stephanopoulos & Han (1996) indicated that there is a growing body of evidence that the use of neural networks to map the process inverse directly from operating data is brittle and prone to failure.

The most widely applied technique in multivariable controller design involves developing input-output empirical models from historical plant data for use in a model predictive control (MPC) framework. MPC extracts information for control purposes from a forward, dynamic model using optimisation algorithms, such as the Powell's algorithm (Powell, 1964), to minimise an objective function on-line. The success of MPC has been attributed partly to the ability of control engineers to construct empirical input-output models from plant test data (Morari & Lee, 1999). Such input-output models are constructed in a supervised training guise, since large sets of training data are available from plant historical databases. However, the MPC framework is non-supervisory, since no examples of exemplary target data are used to devise the control action. Rather MPC is goal-directed using an objective function.

The format of available knowledge dictates the selection of an appropriate learning technique or controller development method (Greffenstette et al., 1990). Many real world practical problems that may be subject to automated learning, do not fit well into the supervised learning model (Yao, 1999). Process control is part of a class of problems described as sequential decision tasks, for which effective performance evaluation is only possible after a series of sequential actions have been executed. A goal-directed approach may be more applicable than a supervised learning approach. In this case, the development of effective decision tasks is subject to testing the developed controller against a simulated model of the task environment. The controller may subsequently be incrementally modified based on the simulation experience in a reinforcement learning framework (Greffenstette et al., 1990). Rather than optimising control performance over a limited horizon as for MPC implementations, reinforcement learning extracts control knowledge from a process model into a generalisation tool representation (e.g. neural networks, fuzzy logic) off-line.

The motivation for using simulation models to learn control strategies lies in that making mistakes on real systems may have costly or dangerous consequences. Learning may require experimenting with control strategies that occasionally produce unacceptable results in a real world environment. The use of simulation models has been instrumental in several reinforcement learning efforts (Greffenstette et al., 1990).

3.2 ELEMENTS OF REINFORCEMENT LEARNING

Reinforcement learning is a computational approach to understanding and automating goal-directed learning and decision-making. It is distinguished from other computational approaches by its emphasis on learning from direct interaction with a dynamic environment, without relying on exemplary supervision via target data or complete models of the environment. A learning agent consequently needs to sense the state of its environment and take actions that affect the current environmental state. Furthermore, the agent requires a clearly defined goal relating to a desired environmental state or what constitutes the successful completion of a task. Reinforcement learning provides a framework for explicitly defining the interaction between a learning agent and its environment in terms of states, actions and rewards in achieving a specified goal (Sutton & Barto, 1998).

Evolutionary reinforcement learning (ERL) searches the solution space of possible control strategies to find a control strategy that encompasses effective performance in the dynamic environment. This approach has been taken utilising genetic algorithms and genetic programming, as well as other novel search techniques, such as simulated annealing and tabu search (Kaelbling et al., 1996).

A decision making agent interacts with a dynamic process in an iterative fashion (Figure 3-1). In process control the process is generally deterministic and may be stationary or non-stationary. In non-stationary processes the agent's evaluation phase must be long enough to allow for the non-stationary trend to emerge. The evaluation phase must be long enough for the neurocontroller to obtain useful experience from the simulated process. Initially the system may be at some state, s_t. The agent is connected to the dynamic process via perception (i.e., sensors) and actions (i.e., manipulated variables). At each interaction step, the agent receives sensory information that is either a full or partial state representation of the process. The agent observes the current state of its environment (or a representation thereof) and selects a control action, a_t. The action impacts the process that enters a new state, s_{t+1}. A reward, r_t, is assigned based on the objective function value of the current state, s_t. The rewards, r_t, over a defined evaluation period are accumulated and the objective is to determine a sequence of tasks that will maximise the expected total reward. The agent can learn to do this over time by systematic causal learning; mapping states to actions guided by a wide variety of algorithms through the reward signal (Kaelbling et al., 1996). The input signals to the agent only provides state information for the environment and do not explicitly direct the agent towards a control strategy. Learning is only based on the total reward during the evaluation period (Greffenstette et al., 1990).

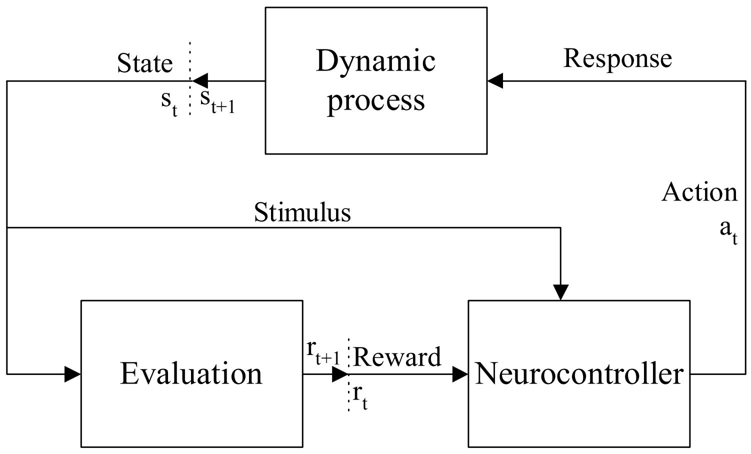


Figure 3-1 - Agent-environment interaction in a reinforcement learning framework. A neurocontroller agent interacts with a dynamic process to learn an optimal control policy from cause-effect relationships.

As reinforcement learning problem definitions become more complex, the number of possible states observations for the environment generally grows exponentially. Therefore, most practical reinforcement learning problems have a large number of possible state observations. For large state spaces, learning agents cannot possibly encounter every state and must apply action decisions, learned from previously observed states, to states not observed during learning.

Real world applications preclude storing the rewards associated with all possible states and actions. Except in very small environments, storage of reward representations presents an impractical memory requirement. It also makes inefficient use of experience. In a large, smooth state space it is generally expected that similar states will have similar reward values and similar optimal actions. A more compact representation than a simple look-up table is thus required. The difficulty in learning in large state spaces is addressed through generalisation techniques, which allow for compact storage of learned information and sharing of knowledge between "similar" states and actions. The reinforcement-learning model requires the storage of a variety of mappings; including state to action policy transitions ($S \rightarrow A$) and deterministic state transitions (S_t x $A \rightarrow S_{t+1}$). Generalisation tools such as artificial neural networks, fuzzy logic and genetic programming may be utilised to compactly represent these mappings (Kaelbling et al., 1996).

The learning process is impacted largely by the observability of the state (i.e., the Markov property), the balance between exploration and exploitation of the environment, how reward is allocated (i.e., credit assignment problem) and how the future is taken into account (i.e., behaviours of optimality).

3.2.1 The Markov Assumption

3.2.1.1 The Markov Property

The greater majority of learning algorithms for reinforcement learning application focus on decision tasks that are described as Markov decision processes (MDP). A Markov control task implies that for each moment in time the agent, (1) directly observes the full state of the environment and (2) the outcome of the action, s_{t+1} and r_t, depend only upon the action taken, a_t, and the current state, s_t.

An agent may have access to all state information either through external sensors or internal representation. Process systems of this type are said to be memoryless and to satisfy the Markov property. The Markov property implies that knowledge of the full state is always sufficient for optimal control. Thus, even though control strategies may incorporate additional information (e.g. a history trace), these strategies cannot outperform the best control strategy based upon full state information. Formally, a decision task is non-Markov, if additional information other than the observed state can be used to better model the process dynamics and improve control (Whitehead & Lin, 1995).

Although the Markov assumption holds for a wide variety of control problems, numerous tasks are inherently non-Markov. These tasks are referred to as hidden state tasks, since information for representing the full state of the environment is omitted from the agent's input. Two significant control problems that are not naturally formulated as Markov decision processes, are (1) where the system has a significant degree of control over the information collected by its sensors (e.g., in active vision) and (2) where the system has a limited set of sensors that are not able to provide adequate information regarding the present state of the environment (Whitehead & Lin, 1995).

3.2.1.2 The ubiquity of non-Markov tasks

Markov tasks are an ideal. Non-Markov tasks are the norm. An agent that is uncertain of the full environmental state necessarily faces an internal decision problem that is non-Markov. Sources of uncertainty abound. Sensors have physical limitations and are often matched imperfectly to a given task. Sensor data is typically noisy, sometimes unreliable, and full of spurious information. Sensors also have limited range. State information may also be hidden in time, that is, the agent requires past and current sensor information in order to take appropriate action (Whitehead & Lin, 1995).

Such hidden state tasks thus arise when temporal features (such as velocity and acceleration) are important for optimal control, but not included in the system's primitive sensor set. Even if perfect sensors were available, many control problems are too ambiguous or ill-posed to specify the full state space in advance. Indeed, part of the agent's task may be to discover a useful set of state variables for solving the problem. Integrating the learning and active perception tasks invariably leads to non-Markov decision tasks. Active perception requires an intelligent agent to actively control its sensors in order to sense and represent information that is relevant to its ongoing activity. If an agent must, as part of the task, learn to control or select appropriate sensors, its internal control problem will necessarily be non-Markov. During learning there will be periods of time when the agent will improperly control or utilise the available sensors, fail to attend to a relevant piece of information, and fail to unambiguously identify the state of the external task (Whitehead & Lin, 1995).

3.2.1.3 Difficulties for reinforcement learning

Intelligent control systems must deal with informational limitations posed by their sensors. The straightforward application of traditional reinforcement learning methods to non-Markov decision problems in many cases yields sub-optimal results and in some cases severely degraded performance. These difficulties stem from the agent's inability to obtain accurate estimates of the environment's state variables from the sensor inputs. Without accurate estimates the agent cannot accurately gauge the applicability of the control strategy. Also, relevant states (hidden states) may be omitted as sensory input to the agent. Due to poor sensor information two actual states in the environment may be misperceived as a single state in the agent's internal representation, viz. perceptual aliasing (Whitehead & Lin, 1995). Evolutionary methods are suited to problems where the state of the process cannot be easily determined (Sutton & Barto, 1998).

3.2.2 Exploration and Exploitation

Even with full state representation from the agent's sensor inputs, a balance must be maintained between exploration and exploitation in the agent's search for an effective control strategy. In order to gain a greater reward, an agent must prefer actions that it has tried in the past and found to be effective in producing rewards. However, in order to discover such actions, it needs to perform actions that it has not attempted in the past. The agent has to exploit what it has learned in order to obtain reward, but it also has to explore to pursue more effective actions in the future. Neither exploration nor exploitation may be pursued exclusively, without failing to learn the task. The agent must try a variety of actions and progressively favour those that produce improved rewards (Sutton & Barto, 1998). A greedy strategy prefers exploitation over exploration, which could lead to discovering local optima rather than the global optimum (Kaelbling et al., 1996). For neural network applications, the available neurons thus map regions of the sub-optimal regions of the state space, starving the true optimal regions of representation. Exclusive exploitation thus reflects a convergence method prone to finding a local optimum.

Evolutionary reinforcement learning methods effectively balance exploration and exploitation of the solution space by maintaining a population of agents that execute different control strategies. As multiple strategies are represented in the population, evolutionary techniques sample various control decisions. Evolutionary search methods progress non-randomly by probabilistically assigning a greater number of evaluations to strategies that display more effective behaviour than an average strategy. This may be regarded as an exploitation step. The evolutionary operators (e.g., crossover) produce offspring that represent an exploration step into the solution space. Other than in the case of random exploration strategies, this exploration step is directed towards solutions that have a greater probability for producing greater rewards. Offspring of effective parents will probabilistically have a similar or improved performance in comparison with their parents. Exploitation is also maintained as parents with high fitness values are often transferred unchanged to the next generation for evaluation. Maintaining genetic diversity ensures that the learning algorithm is also able to adapt to non-stationary environments.

Learning from causal interactions implies that the current best control strategy needs to be updated to a new control strategy that produces greater reward. Consideration needs to be given to how much greater the reward must be, before it is considered prudent to update the current control strategy (Moriarty, 1997). Evolutionary strategies make revisions by applying genetic operators to the population, only once several agents in the population, each over a sequence of actions, have been evaluated. Revisions are thus based on a wide pool of global information. Updating using information gathered from several states and actions, reduces the probability of error owing to reward biasing as a result of occurrences in the environment (Moriarty, 1997).

An attractive feature in maintaining a population of agents lies in that even though a single agent's reward may not reflect the agent's intrinsic value, evolutionary algorithms tend to maintain a degree of robustness. A genetic algorithm's replacement policy typically replaces the weakest individuals in the population. Provided that the agent's biased reward (fitness) is not significantly lower than its actual intrinsic reward, the agent may survive without change into the next generation. Another evaluation opportunity is thus afforded.

3.2.3 Credit Assignment and Behaviours of Optimality

An agent's actions not only determine the reward at the current time step, but also the next state of the process. The agent must take into account the possible reward value of the next state in completing the task, along with the state's current reward when it decides which action to select. The model of long-run optimality determines exactly how the agent should take the value of the future into account. The agent needs to learn from delayed reinforcement. It may take a long sequence of actions, receiving insignificant reinforcement to finally arrive at a state with high reinforcement. The agent must learn which actions are correct based on reward that is likely to be obtained arbitrarily far into the future (Kaelbling et al., 1996).

To provide effective reinforcement to an agent, a specification is required regarding how the agent should take the future into account, when taking control actions in the present. Many models of optimal behaviour exist. Recall that each state, s_t, has an associate reward value, r_t. The finite horizon model maximises an expected reward for a limited number of time steps into the future. The infinite-horizon discounted model takes a large time horizon into account, but requires the agent to increase performance from one time step to the next by discounting (i.e., reducing) the value of future rewards. The infinite-horizon discounted model thus requires an agent to reach states with high rewards in less time, as such states have reduced reward farther into the future. The average-reward model requires the agent to take actions that optimise its long-run average reward. No distinction is made between a control policy that gains significant reward in the initial execution phase and a control policy that garners significant reward in the latter execution phase. However, an average credit assignment makes comparisons between two agents with the same total reward difficult (Kaelbling et al., 1996).

A given behaviour of optimality affects the control response of an agent, but does not provide information on how individual control actions contributed to the total reward. Control problems that rely exclusively on the comparatively uninformative failure signal for feedback, present a challenging problem to reinforcement learning techniques. Consider a sequence of discrete (binary 1 or 0) control actions followed by a failure signal (F). At any instance, the agent may thus select only between a high (1) or a low (0) output signal in response to its environment. In the following action sequence 100011110000F a classic credit assignment problem exists (Whitley et al., 1993).

In such an action sequence, distributing credit (reward) for success or failure among the many control actions that produced success or failure becomes a difficult task. Each ultimate success (or failure) is thus associated with a vast number of internal decisions. For learning to occur from such sparse reinforcement information, the task must be divided into components. The measure of success lies in the completion of the entire goal. If a goal is achieved, its sub-goals (rather the transformation function that completed the sub-goal, i.e. s_t x $a_t \rightarrow s_{t+1}$) are reinforced; if not such transformations are inhibited. Assigning $^1/_{12}$ of the credit to each individual task for either success or failure, would only be feasible should each task have a sufficient degree of independence. Assigning reinforcement to any real world individual sub-goal, would thus typically involve determining the relative contribution of interrelated sub-goals (Minski, 1961).

Evolutionary algorithms, such as genetic algorithms, assign contribution effectively by only propagating genes that aided in achieving the control objective.

3.3 FOUNDATIONS OF GENETIC ALGORITHMS

Genetic algorithms (GA) typically initialise by randomly generating a population of encodings or genotypes that represent possible solutions to a particular problem. These genotypes are evaluated to obtain a quantitative measure of performance based on an objective function. For a 16 bit binary encoding of the genotype, the total number of possible solutions (i.e., in the hyperspace) is 2^{16}. Consider the following binary genotype: 1101 0011 0010 1101. The genotype may represent, for example, a simple neural network with four links, where each connection weight is represented by four bits. The goal of genetic recombination is to find a set of parameters that yield an optimal or near optimal solution to the problem. Recombination requires two parents. Using two "break-point" recombination a second binary genotype yxyy xyxx yyyx yxxy (where x and y represent 0 and 1 respectively) may be recombined with 1101 0011 0010 1101 to produce the following offspring -

1101 0yxx yyyx y101
yxyy x011 0010 1xxy

Reproduction opportunities are allocated so that the best genotypes receive more reproduction opportunities than those having poor performance. Even a small bias in reproduction opportunities typically produces discriminatory pressure to allow "artificial selection". More trials are consequently allocated to genotypes containing pertinent fragments of solution information (regions in the hyperspace) that tend to contribute to above-average solutions (Whitley et al.,1990).

To understand how recombination on binary genotypes can be related to hyperspace, consider a "genotype" that is encoded with just 3 bits. With three bits the resulting search space is three dimensional and can be represented by a simple cube (Figure 3-2). Let points 000, 001, 010 and 011 be the front face of the cube. The front plane of the cube can be characterised as all the genotypes beginning with 0. If * is used as a wild card match symbol, then this plane can also be represented by the similarity template 0**. These similarity templates are referred to as schemata; each schemata corresponds to a hyperplane in the search space. All bit genotypes that match a schema lie in a particular hyperplane. In general, every binary encoding corresponds to a corner in a L-dimensional hypercube and is a member of 2^L-1 different hyperplanes, where, *L*, is the length of the binary encoding. For example, the genotype 011 not only samples its own corner in the hyperplane (011) but also the hyperplanes represented by the schemata 0**, *1*, 01*, 0*1, *11 and *** (Whitley et al. ,1990).

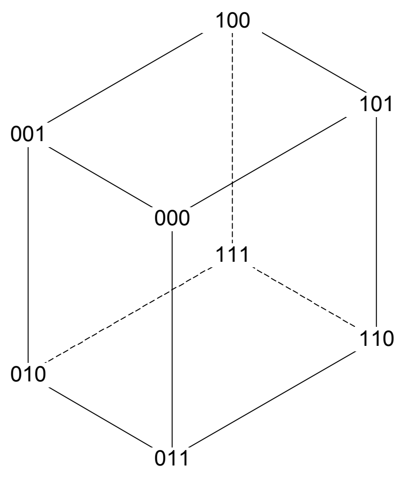


Figure 3-2 - Hyperplane Cube.

The characterisation of the search space as a hyperplane is not simply a way of describing the space. It relates directly to the theoretical foundations of genetic search. Each schema (chromosome) represents a different genetic "fragment", a different combination of "alleles" (genes). When recombination occurs, genotypes are exchanging hyperplane information. For example, if 10101100 and 11011110 are recombined, the offspring reside in the hyperplane 1***11*0. This represents an "order-4" hyperplane, as four bits are specified in the schema. This hyperplane contains 6 ¼ [%] of the all genotypes in the search space - all genotypes with 1 as first bit, 1 as fourth bit, 1 as fifth bit and 0 as eighth bit. The section of the search space that is in contention between the two genotypes, is -010--0- and -101--1-. That is, the bit positions not represented by a dash (-) may be either 0 or 1, depending on the result of the applied genetic operator (such as crossover) (Whitley et al. ,1990).

The offspring resample (re-evaluate) those hyperplanes (gene combinations) inherited unchanged from the parents, but resample (re-evaluate) these gene combinations in a new context – evaluating from a new corner in the L-dimensional hypercube. By testing one new genotype, additional information is gained about the 2^L-1 hyperplanes that intersect at a corner in the hypercube where that genotype resides (Whitley et al. ,1990).

After a genotype has been tested, it is probabilistically given the chance to reproduce at a rate that reflects its "fitness" relative to the remainder of the population. Recombining the "genetic material" from two parents by crossing the binary encodings, allows other hyperplanes in the search space to be sampled (resampled). However, these new probes will be biased towards hyperplanes that have displayed above-average performance. If a schema is common to genotypes that have above average performance, this indicates that the schema may represent a hyperplane that on average contributes to optimising the target problem. Credit assignment is thus addressed by rewarding (i.e., propagating) effective schema in the population. These hyperplanes thus represent sections of the search space that also require greater exploration. Recombining allows for hyperplanes, which have displayed an above average performance, to increase their representation in the population. Genetic search thus proceeds by changing the sampling rate of hyperplanes in the population (Whitley et al. ,1990).

Genetic algorithms are capable of performing a global search of a solution space as there is reliance on hyperplane sampling to guide the search, instead of searching along the gradient of a function as with other optimisation routines. The property of genetic search that allows efficient sampling of numerous hyperplanes in parallel, is referred to as implicit parallelism. This implicit parallelism property allows for a robust search method capable of effective directed search without using gradient information (Whitley et al., 1993).

3.4 DIFFICULTIES IN EVOLVING NEURAL NETWORKS

Genetic algorithms inherently do not scale up effectively. A bias exists against schemata that are highly separated in the binary encoding. As the genotype length increases, a greater proportion of the sampled hyperplanes will span more than half the genotype length, which will thus have a greater probability of being disrupted by crossover. This observation is assumed to contribute to scale-up (i.e., More complex solution representations require longer binary encodings) problems with binary encodings (Whitley et al. ,1990).

The structure of the genotype encoding the phenotype may also present challenges. One difficulty in evolving neural networks with genetic algorithms is that multiple symmetric representations exist for any single neural network. Recombining two functionally effective, but structurally dissimilar networks, may result in offspring that are not viable solutions or present a considerable loss in functionality. Such inconsistent performance feedback to the genetic algorithm, present as high variance hyperplane sampling (section 3.3), may significantly retard the genetic search. For example, assume (Figure 3-3) that four hidden nodes perform tasks A, B, C and D as a solution to some problem. With the same connectivity pattern, the two networks in Figure 3-3 are functionally identical. The position of a functional hidden node in the feed-forward neural network has no effect on the neural networks performance. Recombining these two functionally similar neural networks, could result in children with functionality ABDB and CACD (one point crossover on a hidden node boundary). These two combinations do not represent a solution to the problem, as one functional hidden node is absent from each offspring network. The probability of different structural/functional mappings increases as the number of hidden nodes increase, should the hidden nodes have identical connection patterns. Also, functionally similar hidden nodes may have sets of weights with a different scaling. Recombining networks with dissimilar weights may result in offspring with poor performance. This network recombination problem is referred to as the structural/functional mapping problem or competing conventions (Whitley et al. ,1990).

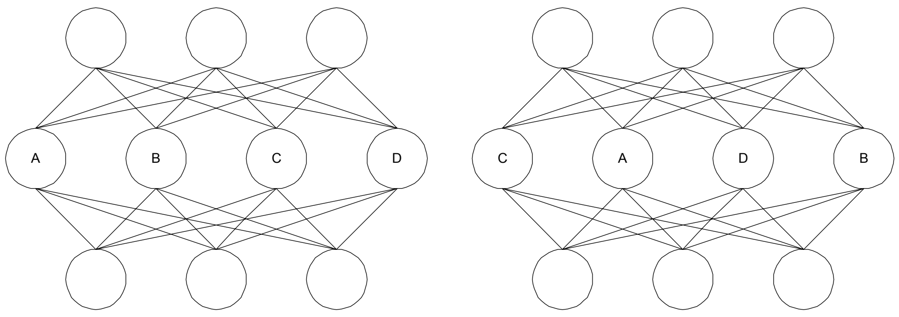


Figure 3-3 - Structural / Functional Mapping problem.

Hyperplane samples which show considerable variance in their fitness values make the search space more difficult to explore. Considerable sampling variance provides inconsistent and conflicting feedback about whether a hyperplane presents a good region in the search space or otherwise. The structural/functional mapping problem typically slows the rate of convergence and increases the number of objective function evaluations dramatically (Whitley et al. ,1990).

By far the greatest challenge to solving complex optimisation problems with genetic algorithms, is avoiding convergence to a local optimum. Maintaining genetic diversity is the key to maintaining a search for the global optimum.

3.5 MAINTAINING DIVERSITY

Evolutionary algorithms operate by continually breeding the population's best genotypes to find the best combination of the available genetic building blocks. This process typically leads to convergence around a single "type" of genotype. Should a genetic algorithm fail to find the global optimum, it is most frequently the result of premature convergence around a genotype that encodes a local optimum. By increasing the representation of high performance hyperplanes in the population, diversity is lost. Exploitation has thus reduced the potential for further exploration. Greater diversity in the population translates to more hyperplane information to drive a continued search (Whitley & Starkweather, 1990).

Local convergence has limited impact in function optimisation, but proves to be of great consequence during the evolution of complex decision strategies. The mutation operator typically aids the evolutionary algorithm from escaping a convergence to a local optimum. Mutation contributes to a more random search of the solution space, which is highly inefficient for large complex problems. A reliance on mutation slows the genetic search considerably. Maintaining a diverse population allows for the continued efficient use of recombination (crossover), allowing for more effective traversals of the solution space. In real world problems, timely solutions are paramount (Moriarty & Miikkulainen, 1998).

Related to the difficulty of maintaining diversity is generalisation. In many settings, convergence by the GA to a single specialised genotype is not appropriate. In a classifier system, the GA needs to evolve a set of rules that are specialised to various tasks (or niches) rather than producing a homogeneous (converged) population of similar rules. A computational model of the immune system illustrates this principle. A population of antibodies needs to be evolved to recognise a set of antigens. Should the antibody population be sufficiently large, the evolution of different types of antibodies that specialise in recognising different classes of antigens, is desirable. The evolution of one generalist antibody, that weakly matches all antigens, is necessarily an inferior solution. Should the population be insufficiently large to evolve an antibody specialisation for each type of antigen, a reasonable response would be to evolve generalists that recognise subsets or specific classes of antigens. In traditional GA optimisation, the generalist usually does not emerge as the population typically specialises in optimising a single fitness peak. The evolution of generalists is imperative for real world applications, where a single genotype (antibody) must address a number of environmental circumstances. The degree of generality should emerge implicitly as a result of the characteristics of the environment and should not be explicitly specified in the search algorithm (Smith et al., 1993).

Maintaining diversity for controlled convergence and ensured generalisation remains an elusive goal. The genetic operators, through their mutual interaction, should provide an optimal balance between exploration and exploitation. A high rate of mutation or less aggressive genetic selection strategies are most commonly employed to attempt diversity maintenance. Increasing the mutation rate only succeeds in artificially introducing noise into the population to promote diversity. Less aggressive selection strategies more often only delay convergence at the expense of a slower search. Despite these implications, such algorithms generally produce better results than aggressive convergent evolutionary algorithms (Moriarty & Miikkulainen, 1998).

A simple method of sustaining genetic diversity is to use large population sizes. Empirical tests have shown that this has the desired effect, except that it is usually necessary to double population size for an incremental increase in performance. This approach also tends to double the search time. The most successful approaches to maintaining diversity involve a mechanism that controls and prevents an unbalanced proliferation of genotypes (Davidor, 1991). The importance of promoting population diversity has lead to the development of several more advanced methods. These include crowding, distributed genetic algorithms, local mating and fitness sharing. These techniques rely on external genetic functions to maintain diversity in the genetic material. Diversity is thus assured through the use of computationally intensive operations (Moriarty & Miikkulainen, 1998).

3.5.1 Restricting the selection process (preselection & crowding models)

Preselection and crowding maintain diversity by limiting co-existence of similar genotypes in the population. Preselection schemes recognise that calculating a dissimilarity index for each genotype may be computationally prohibitive. Preselection assumes that parent genotypes share many similarities with offspring genotypes. Should one of the offspring have a higher fitness than the worst parent, it replaces the parent (MahFoud, 1992).

Crowding is inspired by an ecological phenomenon. Similar genotypes in a natural population, often of the same specie, compete for limited resources. Dissimilar genotypes tend to occupy different niches, which limits competition between dissimilar members. At equilibrium in a fixed-size population, new members of a particular niche replace older members of that niche. The overall number of members of a particular niche should ideally remain constant (MahFoud, 1992).

The traditional crowding scheme thus elaborates on the preselection mechanism. Crowding establishes niches by replacing genotypes that have a similar schema with new genotypes. A "steady-state" GA creates new genotypes one at a time and inserts these solutions into the population by replacing existing genotypes. A sub-set of the population is selected at random (equal to the crowding factor) and the genotype in that sub-set that shares the most schema (hamming distance) with the new genotype, is replaced. The more similar a genotype becomes to other genotypes in the population, the greater the selection pressure becomes against its continued survival in the population (MahFoud, 1992).

While such a scheme appears plausible for maintaining diversity, though computationally intensive, this is not the case in practice. When genotypes are replaced, stochastic errors create significant genetic drift. Genetic drift causes a genetic algorithm to converge to one region of the search space, even though many different regions have equal fitness (MahFoud, 1992).

A similar approach to preselection and crowding involves a uniqueness operator. The uniqueness operator maintains diversity by incorporating a censorship premise. Offspring are inserted into the population only if such offspring are different from all genotypes at a specified number of loci (hamming distance) (Davidor, 1991).

3.5.2 Dividing the population into subpopulations

Several parallel GA approaches have been introduced that explicitly divide the total population into a number of subpopulations. A traditional genetic algorithm executes for each subpopulation separately. Each subpopulation, due to inherent sampling biases, exploits its genetic material in a slightly different manner and converges to a slightly different, but similarly competitive, solution. An attractive feature of this approach is that each subpopulation may have a different crossover and mutation rate, maintaining a balance between exploration and exploitation in a novel way. The concept stems from population genetics, in which random genetic drift allows each subpopulation to explore a different region of the solution space. This maintains a diverse search preventing premature convergence to local optima (Tanese, 1989).

In the distributed genetic algorithm, each subpopulation is allowed to evolve in isolation with sporadic migration of genotypes between subpopulations. Migration intends to convey significant discoveries between subpopulations. The migration interval specifies the number of generations between migrations, while the migration rate specifies the number of migratory genotypes. Additional offspring are generated before migration, followed by random selection from the "overfull" subpopulation. As fitter genotypes are more likely to reproduce, the additional offspring contribute a higher proportion of above-average genotypes. Fitter genotypes are thus more likely to be chosen for migration. The migrants randomly replace genotypes in the destination subpopulation (Tanese, 1989). Such distributed algorithms are more adapt at discovering global optima, but it has been demonstrated that even limited migration eventually leads to convergence and subsequent loss of diversity (Smith et al., 1993).

An alternative distributed GA involves parallel genetic algorithms without migration. Such implementations are termed partitioned genetic algorithms. Partitioned algorithms consistently find solutions closer to the global optimum than the traditional genetic algorithms. Tanese (1989), however, found that partitioned genetic algorithms are unable to maintain a high average fitness of the entire population, which slows the global population search.

3.5.3 Restricting the mating procedure (local mating)

The traditional genetic algorithm incorporates panmictic selection and mating, that is, any genotype may potentially mate with any other in the population. Otherwise similar to crowding (which is panmictic), these methods prevent the population from becoming too homogeneous by largely disallowing the hybridisation effect of crossover.

Local mating creates niches without explicitly subdividing the population. This approach avoids an explicit subdivision of the population, which implies a prior knowledge of the number of niches (and relative size) in the environment. The goal of local mating is to maintain a number of genotypes on more than one peak in the solution landscape simultaneously (Collins & Jefferson, 1991). The population is, for example, arranged geometrically in a two-dimensional plane with crossover only occurring between genotypes that are geographical neighbours. Random genetic variation between subgroups of genotypes allows for exploration of different regions of the solution space. These islands or geographical subpopulations gradually encode local optima as the subpopulations mature. As the evolution progresses certain islands of highly fit genotypes consume lower fitness islands which are in their vicinity. A stepwise island growth and controlled convergence is maintained until the entire grid is occupied by the genotype encoding the global optimum (Davidor, 1991). These methods slow convergence considerably, but stable subpopulations are not maintained (Smith et al., 1993).

3.5.4 Explicit fitness sharing

Explicit fitness sharing establishes subpopulations by penalising a genotype when other similar genotypes exist in the population. This establishes uncrowded productive niches. Explicit fitness sharing modifies a genotype’s fitness calculation, leaving the standard genetic algorithm unchanged.

In ecological theory such strategies are called negative frequency-dependent selection. Explicit fitness sharing assumes that environmental niches have finite resources available to them. As the number of genotypes in a given niche increases, the availability of resources in this niche decreases. This decreases the viability of genotypes in the niche, and subsequently decreases the niche's members. To maintain a viable population in a niche, the population size must come into equilibrium with the availability of resources (Smith et al., 1993).

Explicit fitness sharing's mechanism causes genotypes to cluster around peaks in the search space. Hereby, the proximity of similar genotypes reduces a particular genotype's fitness. The shared fitness is calculated as follows:

$$f_i' = \frac{f_i}{\sum_{j=1}^{N} Sh(d_{ij})} \qquad \textbf{(3-1)}$$

where d_{ij} the distance between i and j under a given metric and $Sh(d_{ij})$ is the sharing fitness. The critical parameter in the fitness sharing technique is σ_s, which dictates a cut-off distance, beyond which no sharing occurs:

$$Sh(d_{ij}) = \begin{cases} 1 & if \quad d_{ij} = 0 \\ 1 - \left(\dfrac{d_{ij}}{\sigma_s}\right)^{\alpha} & if \quad d_{ij} \le \sigma_s \\ 0 & otherwise \end{cases} \qquad \textbf{(3-2)}$$

The method is robust, but has a number of limitations as presented in Table 3-1. Fitness sharing is, however, able to establish stable niches in contrast to other methods that simply slow the approach to convergence (Smith et al., 1993).

Table 3-1 – Explicit fitness sharing limitations.

Limitations of explicit fitness sharing
• Involves a significant number of fitness comparisons, which is computationally intensive (N^2 comparisons in a population of size N).
• Requires an appropriate setting for σ_s based on prior knowledge of the number of optima in the solution space.
• The efficacy of the parameter σ_s is dependent on the presence of uniformly distributed optima in the solution space. Less uniformly distributed optima may be ignored in the search.

3.5.5 Implicit fitness sharing

Implicit fitness sharing entails the search for cooperative species which encode the optimal solution collectively, within a single population of competing and cooperating genotypes. Cooperative-competitive evolution isolates the most fundamental type of cooperation, which entails problem decomposition via niching. Cooperation between niches and competition between genotypes in the same niche occurs concurrently. This model of intertwined, multi-level cooperative and competitive interactions appears natural, based on real-world examples of ecologies, economies, and societies. However, its potential power for expressing complex concepts and behaviours is offset by the tremendous complexity of the population dynamics (Horn & Goldberg, 1996).

In this model various species solve different elements of the problem, earning separate, distinct rewards. Each genotype no longer represents a complete solution to the problem, but rather a partial solution that must cooperate with other partial solutions. Each genotype's role in the complete solution is thus limited, forming a co-evolutionary search for different kinds of genotypes that make up the complete solution. Genotypes that perform the same task, compete for the same rewards, viz. weak cooperation. Genotypes that do not overlap in their tasks cooperate in an indirect manner, viz. strong cooperation. The goal of weak cooperation is to compete (i.e., exploit) for as much of a specific resource as possible. The natural mechanism for dealing with such competition is a search for uncontested resources. The population of genotypes thus divide the total available resources implicitly, sharing rewards (credit or fitness) among all genotypes. Successful weak cooperation thus leads directly to strong cooperation. Such induced speciation or niching is an emergent phenomenon that is a prerequisite to all other types of cooperation.

Implicit fitness sharing is a powerful promoter towards cooperation. Even under constant and rigorous (i.e., aggressive) application of evolutionary operators, such as crossover, diversity is maintained in the population. The only way to maintain high-quality diversity in the face of high selection pressure is to balance convergence with a restorative force, such as "niching pressure" (Horn & Goldberg, 1996).

In a simple genetic algorithm each genotype is evaluated according to a single scalar fitness function, independent of other genotypes in the population. The simple GA's task may be viewed as an optimisation of the average population fitness. The optimum population thus consists entirely of copies of the best genotype. When genotypes influence each other's fitness evaluation, the task of the GA may no longer be modelled as an optimisation of the total population fitness (Horn et al., 1994).

Implicit niching reaches a dynamic equilibrium of diverse genotypes quickly (i.e., a fast convergence rate) and this equilibrium is maintained (i.e., high restorative pressure causes long niche extinction times). The ability to balance selection pressure with restorative force allows for the virtual indefinite maintenance of high quality, diverse niches (Horn & Goldberg, 1996).

As the search entails the optimisation of the various elements of the problem, several parallel searches are performed in the decomposition of the solution space. This allows for the more aggressive application of the evolutionary algorithm, increasing markedly the rate of evolution (Moriarty & Miikkulainen, 1998). The evolutionary algorithm developed in this study, Symbiotic Memetic Neuro Evolution (SMNE), largely falls into this category of evolutionary algorithm, allowing for the evolution of neural networks via implicit fitness sharing.

3.6 CONCLUDING REMARKS

Heuristic rules and subjective engineering judgement play a significant role in current plant-wide control methodologies as described in chapter 2. However, other sources of process information are available, such as (1) expect knowledge formally extracted from experienced operators, (2) detailed fundamental models and (3) empirical input-output models obtained from plant historical data. These sources may be exploited in either a supervised learning or a goal-directed learning approach. As a goal directed learning approach, reinforcement learning offers unique opportunities for developing control strategies, provided complex learning fundamentals are addressed efficiently. A learning algorithm needs to explicitly address the Markov property, balance exploration and exploitation and effectively assign credit to appropriate control actions. Evolutionary algorithms attend to these learning fundamentals, but also present method-specific challenges, viz. scale-up difficulties and the structural/functional mapping problem. A global search may only be ensured by maintaining genetic diversity, which is all-important for robust learning in complex control tasks. Implicit fitness sharing offers superior benefits over other diversity mechanisms. The learning algorithm in this work, symbiotic memetic neuro-evolution (SMNE), relies significantly on implicit fitness sharing. SMNE also facilitates a synergy between a global evolutionary and a local cultural search from neural network controllers as described in chapter 4.

4 EVOLVING NEURAL NETWORKS

OBJECTIVES OF CHAPTER 4
• Describe typical applications of neural networks in control systems.
• Highlight different approaches to evolving neural networks within a generational context.
• Describe the SANE algorithm, a second generation algorithm, predecessor to the SMNE algorithm.
• Describe the SMNE algorithm, developed in this study, within the context of other third generation algorithms.
• Describe ANS as a means of adapting neurocontrollers on-line.
• Demonstrate the SMNE and ANS algorithms within a bioreactor case study.

4.1 INTRODUCTION

Linear controllers are often used in the process industries, frequently in non-linear applications, owing to ease of implementation. For non-linear processes, Economou & Morari (1986) contended that linear controllers are unable to match the autonomy of rationally designed non-linear controllers. Non-linear control methodologies have emerged to address the short-comings in linear control. Most chemical plants exhibit inherent non-linearity and frequently have a large operating region. A linear controller may be unable to compensate for non-linearities, resulting in poor performance or even instability. Many processes exhibit discontinuities, such as dead zones, saturation and hysteresis, making linear control solutions ineffective. Non-linear controllers generally also compensate better for model uncertainties (i.e., plant-model mismatch) than linear controllers. More stringent product and operating specifications have also necessitated non-linear control approaches. Linear controllers thus incur an economic opportunity cost, as a consequence of sub-optimal performance in non-linear processes (Kuttisupakorn et al., 2001).

Advances in computing power have also placed the greater computational requirements of non-linear control methods within the reach of industrial computation. However, conventional non-linear control design necessitates extensive mathematical analysis, a significant degree of engineering judgement and expert process knowledge. Prior to commencing a non-linear controller design, the process engineer needs to have a clear understanding of how the control strategy will be implemented. Conventional non-linear control design methods are difficult to automate, which hampers widespread implementation (Brengel & Seider, 1992).

These difficulties with conventional non-linear control can be surmounted by means of intelligent control techniques. Control systems using artificial intelligence methods have been designated 'intelligent control'. A control system is classified as intelligent, provided it has the ability to act robustly in uncertain environments. Intelligence is a property of the system, emerging from the generalisation tools and combinatorial search methods applied to input data, which produce an appropriate control action. Appropriate control actions are defined as those that increase the probability of success, where success pertains to achieving sub-goals that support the control system's overall control task. Machine learning or artificial intelligence methods include expert systems, fuzzy logic, neural networks, evolutionary computational or any other means by which data is analysed, organised and converted into knowledge (RayChaudhuri et al., 1996). RayChaudhuri et al. (1996) recognised that a combination of neural and evolutionary methods could emerge as efficient and robust tools for developing control systems that are self-designing and adaptive.

The use of neural networks in any process application is motivated by a desire to achieve the non-linear information processing of the brain. Neural networks have been applied in process modelling, process control, inferential estimation (i.e., soft-sensors) and supervisory applications (Wills et al., 1991). Process modelling may include predictive non-linear forecasting for controlled variables (Ydstie, 1990; Bhat & McAvoy, 1990; Foster et al., 1992; Willis et al., 1992)) and modelling of unknown model parameters (e.g., reaction kinetics) in ordinary differential equations forming hybrid models (Henson, 1998). Neural networks in process control may also provide gain scheduling to linear PID controllers, mapping the state space to a particular set of control parameters (Narendra & Parthasarathy, 1990). Also, a neural network may function as the control law, mapping the state space to control actions (Conradie et al., 2000). Frequently, off-line analyses have long sampling periods. Neural networks have been trained on off-line analyses to serve as soft-sensors, providing an on-line, inferential estimate to an off-line analysis (Willis et al., 1992). The pattern-classification of neural networks has been implemented in a supervisory capacity, suggesting re-calibration of instruments via fault detection and re-tuning of controllers based on closed-loop performance (Ungar et al., 1990).

Neural networks possess characteristics advantageous to control as in Table 4-1. In process control applications neural networks may be incorporated into the control loop in either direct or indirect control methods. In the direct method, a neural network is trained with input-output data to represent the system's inverse dynamics. The resulting controller is used in a feedforward fashion. In the indirect method, the neural network is trained with input-output data to represent the forward dynamics. Given the current state and the current control action, the network learns to predict the next state of the system. This process model may consequently be used by a control algorithm to calculate the control action (Psichogios & Ungar, 1991).

Table 4-1 - Neural network characteristics relevant to control systems (Kim et al., 1997)

Relevant control features of neural networks
• Ability to represent arbitrary non-linear relations.
• Capability of learning uncertain systems through both off-line and on-line weight adaptation.
• Flexibility to handle input information that is transformed to internal representation allowing data fusion, with both quantitative and qualitative signals.
• Parallel distributed processing architecture allowing fast processing for large scale dynamic systems.
• Neural network architecture provides a degree of robustness through fault tolerance and graceful degradation.

Despite possessing characteristics favourable to process control, the implementation of neural networks in control loops poses difficulties. When employing a control scheme using neural networks as the process model, there are numerous challenges in obtaining effective data to train the neural network. Generally, data is obtained by making various random changes in process inputs over the whole operation range. Perturbation of process inputs is, however, practically prohibited especially in chemical processes, due to economic loss and safety considerations. Additionally, as chemical processes consist of many sequential sub-units, the effects of such perturbations may be propagated to downstream units. Should large scale plants need to be disturbed significantly for neural network training, the economic loss may be significant (Kim et al., 1997). Training data may, however, also be obtained from historical plant data. Most industrial plants are operated by simple linear controllers. PID controllers operate in a region of the state space that may be too limited to allow for effective model development. Satisfactory neural models may not be obtainable from historical process data (Kim et al., 1997).

The control literature presents a number of neural network control schemes or methodologies for use in industrial control applications (Agrawal et al., 1997). Hunt et al. (1992) focused on those structures that have a direct reliance on system forward and inverse models. These structures are, from a mainstream control theory viewpoint, well-established and their properties have been extensively analysed. Hunt et al. (1992) classified the control structures using neural networks as supervised control, direct inverse control, model reference control, internal model control, predictive control and gain scheduling. Of these, neural networks have found considerable application in direct inverse control and model predictive control structures.

4.1.1 Direct Inverse control

Direct inverse control (Figure 4-1) utilises an inverse system model. The inverse model is cascaded with the process, so that the composed system results in an identity mapping between desired response (i.e. the network inputs) and the controlled system output. The network thus acts directly as the controller in such a configuration.

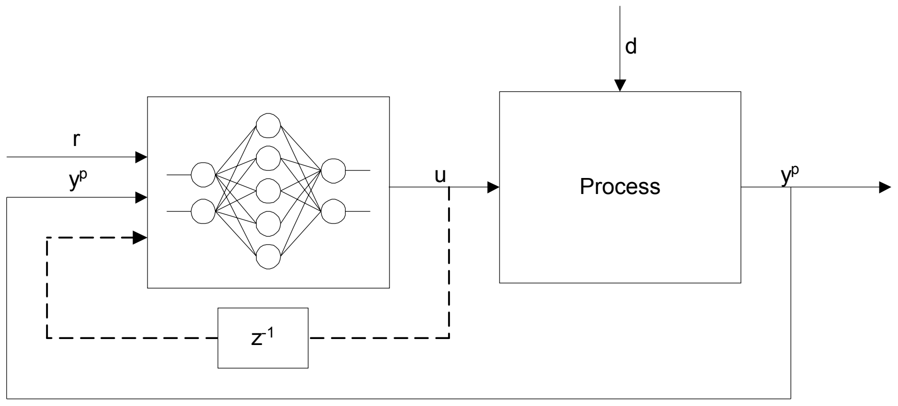


Figure 4-1 - Direct Inverse Control.

Though direct inverse control is well-tested in simulation studies, few laboratory or pilot-scale experimental work has been reported. Real-time application typically involves first identifying the forward dynamics from open-loop perturbation experiments (e.g., step changes). This forward dynamic model may have the following general form as in equation 4-1:

$$u_k = f(y_{k+1}, y_k \dots y_{k-n-1}, u_{k-1}, u_{k-2} \dots u_{k-n-1}) \qquad \textbf{(4-1)}$$

where y_k is the process variable and u_k is the manipulated variable at time step k. The methodology for training direct inverse neural networks is illustrated in Figure 4-2.

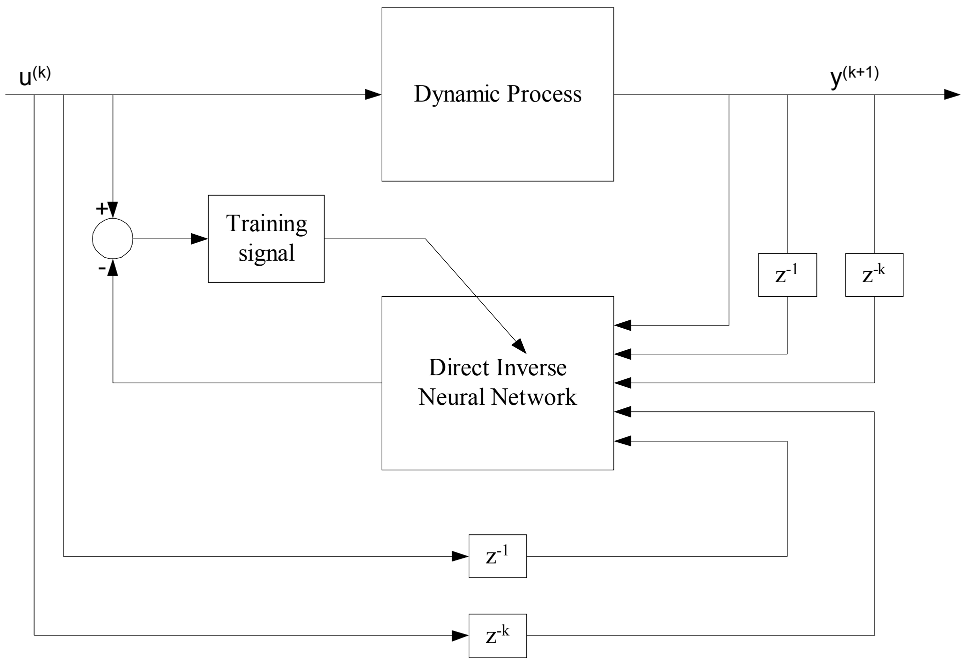


Figure 4-2 - Framework for training direct inverse control neural network controllers.

Khalid & Omatu (1992) trained a neural network to learn the inverse dynamics for temperature control of a 8 [dm^3] water bath. The water bath was equipped with a stirrer and the single final control element was a 600 [W] electric heater. The control objective required stepping the temperature set point with minimal error. The neurocontroller performed better than conventional PID control and had effective disturbance rejection capabilities. Dirion et al. (1995) also used back-propagation to learn the inverse temperature dynamics of a jacketed water bath. Two heat exchangers in the cooling circuit allowed for both heating and cooling of the water bath. Although a single open-loop experiment was sufficient to determine the forward model, Dirion et al. (1995) stressed the importance of covering a wide range of possible input space data (i.e., domain).

Savkovic-Stevanovic (1996) developed a conventional direct inverse neurocontroller for a pilot plant distillation process. The top product composition was the controlled variable with the reflux rate the single manipulated variable. Open loop training data were collected in the region of the desired set point. The inverse model included feed rate, pressure drop and bottoms flow rate as input nodes, thereby providing additional process information as feedforward control within the feedback neurocontrol structure. Savkovic-Stevanovic (1996) did not explain the selection criteria for the input layer's structure in terms of time delay inputs of the controlled variable nor the addition of other process variables as feedforward information. The control response proved to be oscillatory, possibly due to the identity mapping produced by direct inverse control.

Palancar et al. (1998) studied real-time direct inverse control for the neutralisation of aqueous solutions of acetic and propionic acids with NaOH in a laboratory scale CSTR. The controlled variable was the reactor's downstream pH with the NaOH flow rate as manipulated variable. A simplified fundamental model (i.e., the dynamics of pH sensor and the valve transients were not incorporated) was used to generate open-loop data for training a forward neural network model of the neutralisation process. An inverse model was trained based on the forward neural network. The back-propagation algorithm trained both the forward and inverse models. The forward neural network model was updated on-line and used in the control loop to adapt the inverse controller on-line. The control response was highly oscillatory and due to a lack of sufficient training data for typical operating conditions, the generalisation to disturbances was unsatisfactory.

Hussain and Kershenbaum (2000) considered the real-time control of a 100 [L] exothermic CSTR fitted with an external heat exchanger. Forward and inverse neural network models were used in an internal model control (IMC) strategy. Owing to safety and cost considerations, the heat generation and product composition were computed using a simulated process model in tandem with the actual process. Steam was sparged into the reactor to generate the calculated heat generation, making only the composition a purely calculated variable. The control system had the calculated concentration and the measured reactor temperature as the process inputs, with the temperature set point of the cooling jacket as the single manipulated variable. The IMC controller thus served as a master controller to a temperature PID slave controller that manipulated cooling water flow into the external heat exchanger. The training data for the forward model was generated in the open-loop by perturbing the jacket temperature set point. Open-loop data collection was necessitated by the absence of an appropriate method to compute process dynamics from closed loop data. A real exothermic CSTR could never be operated safely in the open-loop. Back-propagation was used to train both the forward and inverse neural network models. The controller proved sensitive to plant/model mismatch and the accuracy of the inverse model.

These real-time applications all involve using back-propagation as the learning rule and their complexity is limited to SISO control applications. The back-propagation algorithm has a sensitive dependence on chosen initial conditions and hence finding the global optimum is not assured (Ydstie, 1990). Critically, these real-time applications emphasise the performance impact of choosing the correct input space (i.e., vectors) based on past process variable measurements, but do not present a systematic or theoretical approach to selecting the input space. The input vector in each case was selected using trial-and-error evaluation. Though model validation techniques have been proposed as guidelines to assess the appropriate number of past inputs and outputs, the final selection was determined on a case-specific basis

(Hussain and Kershenbaum, 2000). The number of delayed inputs to the neural network was frequently estimated from the order of the system and dead time of the process. Model development was always based on open-loop response data, limiting the application of these techniques to open-loop stable processes.

Though these real-time applications typically involved conventional set point tracking, Nguyen and Widrow (1990) noted that neural networks may be trained to minimise cost functions instead of a performance error. This approach contrasts with direct inverse control in that the identity mapping is not sought as the objective function. The desired performance of the closed loop is specified by a reference model or cost function (Hunt et al. ,1992). Hunt et al. (1992) denote this variant of direct inverse control as model reference control (Figure 4-3). In general, this approach is more robust than direct inverse control that seeks perfect control (Seborg, 1989).

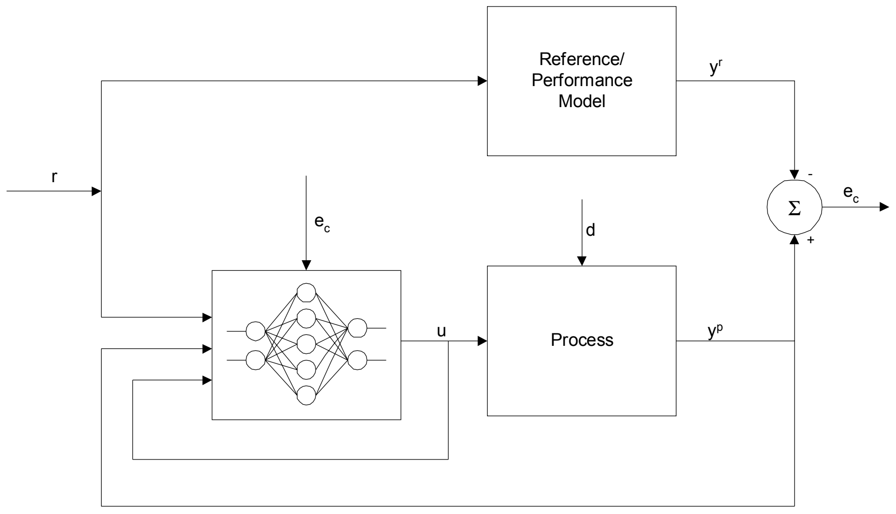


Figure 4-3 - Direct model reference control.

4.1.2 Model Predictive Control

Model predictive control has been successfully extended to non-linear processes. The advantages of non-linear predictive control include explicit handling of process time-delays, constraints, the ability to handle non-minimum phase systems and incorporating knowledge of future set point changes (Sistu et al., 1993).

Model predictive control (MPC) is defined as a control scheme in which the controller repeatedly determines (optimises) a manipulated variable profile. An open-loop performance is optimised on a time interval extending from the current time to the current time plus a prediction horizon. Feedback is incorporated by using process measurement to update the optimisation problem for the next time step. The receding horizon technique is introduced as a natural, computationally feasible feedback law. The method has proven to have desirable stability properties for non-linear systems. Also, the generality of the performance objective, as opposed to standard integral square error between measurement and set point, provides the opportunity to design MPC controllers for higher level functions such as energy or waste minimisation (Eaton & Rawlings, 1992).

As a consequence of its structure (Figure 4-4), a MPC is a feedforward controller for known process changes and a feedback controller for unknown process changes. Thus, MPC can reject measured disturbances more rapidly than conventional controllers, by anticipating their impact on the process. Set point changes are achieved efficiently through their ability to predict an optimal sequence of manipulated variable outputs. The feedback element of a MPC compensates for the effects of unmeasured disturbances on the process outputs and deviations between model outputs and those measured (process/model mismatch) (Brengel & Seider, 1989).

The prediction horizon allows the MPC controller to take control action at the current time in response to forecast error even though the error at the current time is zero. Also the predictive controller may be given information about future constraints and future inputs such as planned set point changes or forecasts of loads or disturbances. Eaton & Rawlings (1992) showed that it is precisely this property of the MPC controller that is beneficial for controlling (scheduling) non-minimum phase plants.

Implementing MPC with a neural network approach, involves utilising a neural network model to provide predictions of the future plant response over the specified horizon. The predictions supplied by the neural network model are passed to a numerical optimisation routine, which attempts to minimise a specified performance criterion in calculating a suitable control signal. The control signal may be chosen so as to minimise a quadratic performance criterion -

$$J = \sum_{j=N_1}^{N_2} \left[y^r(t+j) - y^m(t+j) \right]^2 + \sum_{j=1}^{N_2} \lambda_j \left[u'(t+j-1) - u'(t+j-2) \right]^2 \qquad \textbf{(4-2)}$$

subject to the constraints of the dynamic model. The constants N_1 and N_2 define the horizons over which the tracking error and control increments are considered. The values of λ are the control weights. The remaining parameters are illustrated in Figure 4-4 (Hunt et al., 1992).

Another alternative, is to train a further neural network to mimic the action of the optimisation routine. The controller network is consequently trained to produce the same control output for a given plant input (Hunt et al., 1992).

Model predictive control's multi-step strategy has proven performance in controlling processes in unstable operating regimes. However, the MPC approach remains sensitive to modelling errors in these unstable regions. A disadvantage of the MPC approach is the computationally intensive execution of the optimisation algorithm, especially where linearisation of non-linear system is not applicable. Also, the solution of the optimisation problem - therefore the controller behaviour - depends on a number of tuning parameters. These tuning parameters include the weighting coefficients in the objective function, the convergence criterion, the scaling of the variables and the magnitude of the velocity bounds (Psichogios & Ungar, 1991).

The costs and effort required to implement an advanced control algorithm, such as MPC, include (1) the development of models to describe the process dynamics, (2) the dedication of processing power, and the (3) tuning of more parameters relative to analogue controllers. An MPC implementation is normally only justified for processes that cannot be adequately controlled by using less complex algorithms. These processes typically include the production of chemicals in high purities, chemical reactors with multiple steady state and periodic attractors, extraction processes with narrow two- and three- phase regions (e.g. supercritical extraction), distillation towers with temperature and concentration fronts sensitively coupled to the reflux ratio (azeotropic distillation towers), and processes required to operate in the region of many design and operating constraints (Brengel & Seider, 1989).

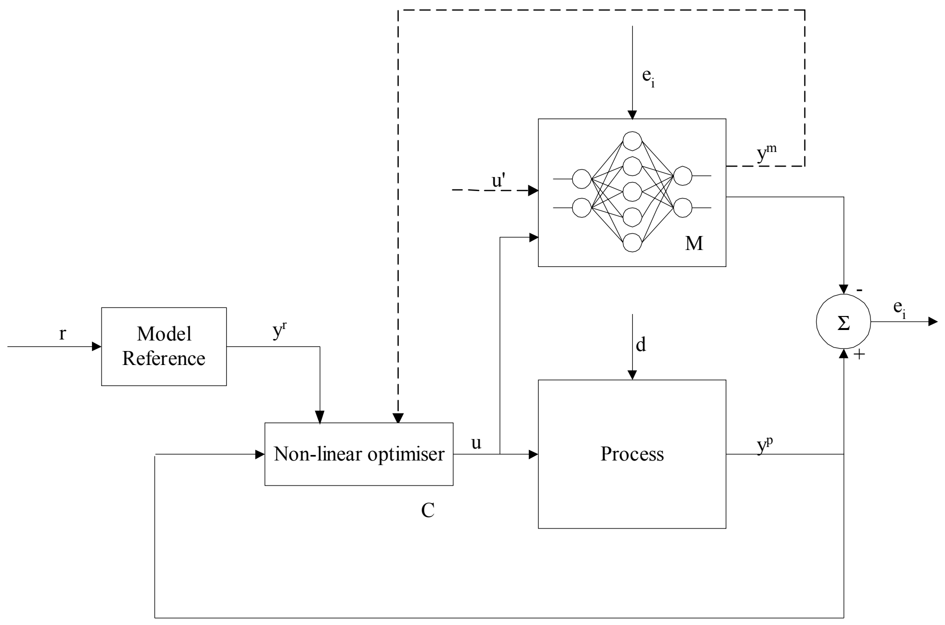


Figure 4-4 - Model Predictive Control structure utilising neural networks

Model predictive control is the industry standard for advanced process control. The complexity of practical problems solved in MPC frameworks far outstrips those attempted with direct inverse control frameworks. Temeng et al. (1995) implemented a non-linear MPC strategy for an industrial packed bed reactor for converting SO_2 to SO_3. The reaction is exothermic and requires interstage cooling after each catalyst pass. The process is characterised by significant process interaction, long time delays, slow dynamics and frequent disturbances. The process is normally operated in the open loop by human operators. Furthermore, the product composition constraints (i.e., environmental considerations) and minimum reaction temperatures pose hard operating constraints. The process has five process variables and five manipulated variables. Dynamic response tests in the open loop generated the training and validation data for MIMO neural network modelling of the process. The control task involved regulatory control of the three pass temperatures, manipulating the cooling water valves that supply the heat exchangers directly. The NMPC strategy was implemented and good disturbance rejection and set point tracking was evident. Temeng et al. (1995) concluded that owing to the significant process interaction, better regulator control was achieved than was possible with decentralised multi-loop controllers (Temeng et al., 1995). Clearly, the interactive, non-linear and MIMO nature of this control problem is far more challenging than those solved with direct inverse control in section 4.1.1.

A different approach to non-linear control entails developing control strategies using evolutionary algorithms, more akin to model reference control (Figure 4-3) than direct inverse control or MPC.

4.2 SINGLE CHROMOSOME AND CO-EVOLUTIONARY CONTROL POLICIES

The use of artificial evolution has become a popular method for developing control strategies in complex problem domains. This requires an approach with marked differences from function optimisation with evolutionary algorithms. Particularly, in real world applications the number of available fitness evaluations is limited, as compared to the often limitless number of fitness evaluations allowed in function optimisation problems (Moriarty et al., 1999).

In the common approach to neuro-evolution, each genotype represents a complete network that is evaluated independently from the other networks in the population (Whitley et al., 1993; Jarmulak et al., 1997). In the GENITOR II algorithm developed by Whitley et al. (1993), which has been implemented in a wide variety of applications, each genotype encodes a full neural network (i.e., complete solution).

GENITOR II incorporates adaptive mutation and a distributed genetic search (section 3.5.2) to sustain genetic diversity. As the population becomes increasingly homogeneous due to convergence, the rate of mutation is increased to maintain diverse genetic material in the population. The population diversity is indirectly monitored throughout evolution by measuring the hamming distance between two parents during reproduction, which may be computationally intensive. The more alike the two genotypes are, the greater the probability of mutation being applied to the offspring. Also, a distributed scheme is incorporated to aggressively exploit subpopulations without exhausting the diversity in the entire population (Whitley et al., 1993). By treating each genotype as a separate, full solution GENITOR II focuses the search towards a single dominant genotype, despite attempts to maintain diversity. Co-evolutionary algorithms contain elements of co-adapting and cooperating niches.

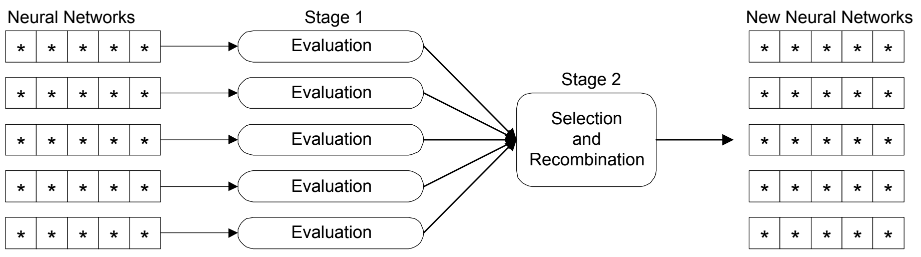
In the computational model of the immune system (Smith et al., 1993), each antibody must compete for survival with other antibodies to recognise a given set of antigens. The antibodies are not dependent on other antibodies for recognising an antigen and only interact implicitly through competition. In the immune system model the fitness of each genotype reflects how effectively it matches an antigen, not how effectively it cooperates with other genotypes. The maintenance of niches in the population is thus based on weak cooperation or co-adaptation. Similar to conventional evolutionary algorithms, genotypes of co-adapting algorithms encode the full solution. However, numerous unrelated optimal solutions exist within an unconverged genetic population.

Cooperative co-evolutionary algorithms present a promising alternative for evolving complex dynamic control strategies. Symbiotic evolution may be defined as a type of evolution where genotypes explicitly cooperate in order to survive in the population. In this respect symbiotic evolution is distinct from co-adapting algorithms, in which genotypes compete rather that cooperate with each other for survival. In cooperative co-evolutionary algorithms each genotype only represents a partial solution to the problem as discussed in section 3.5.5. Complete solutions are compiled by grouping selections of these partial solutions together. This approach thus attempts to optimally combine solution building blocks (i.e., partial solutions) with other partial solutions to form a full solution. Selection pressure exists to form symbiotic relationships. Maintaining these partial solutions or niches in a single population avoids convergence of the population to a single genotype via strong cooperation. Diversity is thus maintained, though a single optimal solution is sought within an unconverged genetic population (Moriarty & Miikkulainen, 1998).

Symbiotic, Adaptive Neuro-Evolution (SANE) was designed as an efficient method for developing neurocontrollers in dynamic environments. Whereas most neuro-evolutionary techniques (e.g., GENITOR II) operate on a population of neural networks (i.e., full solutions), SANE evolves a population of neurons (i.e., partial solutions). Each neuron's task is to determine the appropriate connections and weights that together with other neurons form a robust neurocontroller. As no single neuron may represent a single solution, evolutionary pressure is introduced to evolve neuron specialisations maintaining diversity (Moriarty & Miikkulainen, 1998).

The full and partial encoding schemes were empirically evaluated by Moriarty & Miikkulainen (1996a). SANE (i.e., implicit fitness sharing) was superior to GENITOR II (i.e. adaptive mutation & distributed algorithm) in performance. The different strategies for neurocontroller evolution are illustrated in Figure 4-5. Neuron speciation and more effective credit assignment to genetic material has proven decisive for improved performance.

(a)



(b)

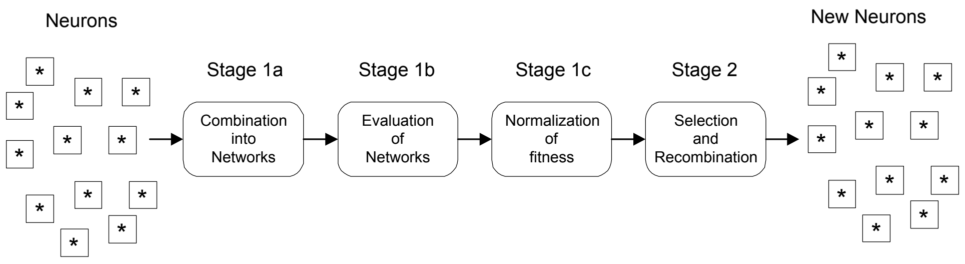


Figure 4-5 - An illustration of the neuro-evolution performed in SANE (b) compared to the standard approach to neuro-evolution (a).

Neuron speciation ensures diversity, which ensures a continued global search. A single neuron cannot dominate a population, since highest fitness may only be obtained in a population where other niches co-exist. Should a niche become too prevalent, its genotypes will not be combined effectively with genotypes in other niches. Consequently, redundant partial solutions are ineffectively combined with other niches and thus obtain lower fitness evaluations. This evolutionary pressure thus selects against numerous genotypes in dominant niches. Solutions are thus found in diverse unconverged populations (Moriarty & Miikkulainen, 1998).

Evolution at the neuron level more accurately evaluates genetic building blocks. In the more prevalent network-level evolution (e.g., GENITOR II), each neuron is implemented with other neurons encoded on a single chromosome. This type of genetic representation may lead to an effective neuron existing along with ineffective neurons. Owing to the resulting low fitness evaluation, knowledge contained in the effective neuron may be lost. In neuron level evolution, continual recombination of a neuron with many other neurons results in a more accurate evaluation of the neural genetic building blocks (Moriarty & Miikkulainen, 1998). Credit assignment is thus more efficient.

Therefore, cooperative co-evolutionary approaches take advantage of a priori knowledge that neural networks are composed of individual neurons. Neuron-level evolution explicitly promotes genetic material in the population that may be useful in constructing complete neural networks. A network-level approach is implicit. Also, evolving at the neuron level, the evolutionary algorithm is no longer expected to identify neurons as the significant building blocks, as neurons are the focus of the evolution (Moriarty & Miikkulainen, 1998).

Despite the advantages of neuron-level evolution, knowledge of effective neuron combinations needs to be maintained.

4.2.1 Knowledge of effective neuron combinations

Neuron evolution alone has proven insufficiently powerful in generating neural networks for complex tasks. Moriarty & Miikkulainen (1996b) enhanced the general principle of implicit fitness sharing (Horn et al., 1994), by investigating how partial solutions may be effectively combined to form complete solutions (Moriarty & Miikkulainen, 1998).

Knowledge of useful neuron combinations must be preserved and exploited. Intelligent direction for neuron combination is desirable. Without intelligent recombination, neurons may not be combined consistently with other cooperative neurons. An effective neuron may be lost if not be effectively combined during a generation. The quality of randomly combined networks would vary to a large extent during evolution. In the initial generations random combination is advantageous, as many different kinds of networks are evaluated to find cooperative neurons. Without intelligent neuron combination the direction of the search during later generations is often stalled, due to inconsistent combination of neurons when the focus should be on better performing networks (Moriarty & Miikkulainen, 1998).

SANE introduces a mechanism whereby a layer of neural network blueprints is evolved along with the neuron population. This blueprint population carries knowledge of effective neuron combinations that may be retained in subsequent generation. The blueprint population results in more accurate neuron evaluations, as neurons are more consistently combined with other neurons that have cooperated well together. Fitter neurons also garner more pointers from the blueprint population, thereby participating in a larger number of networks. This biasing towards historically more cooperative neurons provides more accurate evaluation of the elite neurons. Newer neurons in this model, however, may consequently receive too few evaluations for proper evaluation. Empirically, the allocation of more trials to elite neurons performs better than uniform neuron participation. Figure 4-6 illustrates the relationship between the blueprint and the neuron population (Moriarty & Miikkulainen, 1998).

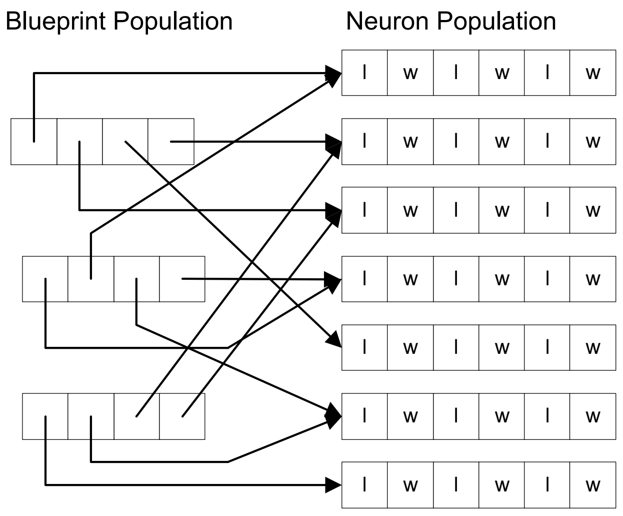


Figure 4-6 - Network blueprint population in relation to the neuron population.

Evolving network blueprints exploits the best networks discovered during evolution. By evolving the blueprint population, the best neuron combinations are also recombined to form new, potentially better, collections of neurons. The blueprint level evolution thus provides a very exploitative search that utilises the best neuron combinations, thereby focusing the search in later generations (Moriarty & Miikkulainen, 1998).

Several third generation evolutionary reinforcement learning algorithms have been proposed since the publication of the SANE algorithm. These include Eugenic evolution (Polani & Miikkulainen, 2000), Enforced sub-populations (ESP) (Gomez & Miikkulainen, 1997), NeuroEvolution of Augmenting Topologies (NEAT) (Stanley & Miikkulainen, 2002) and a cultural algorithm that combines EA with back-propagation (McQuesten & Miikkulainen, 1997). All these newer algorithms have elements of SANE as their foundation. The SMNE algorithm (detailed in section 4.6), developed in this work, draws strongly from the implicit fitness sharing concepts in SANE and as such should be regarded as a third generation algorithm indebted to the

SANE algorithm. In this thesis, the SANE algorithm has been used to develop neurocontrollers and to serve as comparative algorithm to the SMNE algorithm. Section 4.3 details the implementation of the SANE algorithm.

4.3 SANE IMPLEMENTATION

SANE evolves both a neuron and a network blueprint population. Each neuron in the hidden layer specifies a set of weights and connections to the input layer and output layer. Each genotype in the network population specifies a grouping of neurons to include in the network. The neuron evolution thus searches for effective partial solutions (i.e., neurons), whilst the blueprint evolution searches for effective combinations of these partial solutions (i.e., networks) (Moriarty & Miikkulainen, 1998).

Each neuron represents a hidden neuron in a 3-layer partially connected feed-forward network. Gene pairs in each neuron genotype encode an even number of connection-weight combinations. The first gene in a pair encodes a neuron connection and the second gene encodes the weight for that particular connection. Each connection gene is an integer value that ranges between zero and the total number of input and output nodes less one. Decoding a connection gene assigns a connection to either an input or an output node. Should the integer value in the connection field be less than the total number of input nodes, the connection is made to the corresponding input node number. Otherwise, the connection is made to the corresponding output node number. Connections are thus probabilistically assigned to either input or output nodes, based on the number ratio of input to output nodes. Each weight gene is a floating point value with a gaussian distribution around the mean 0, with a standard deviation of typically 2. Initially the connection and weight genes are randomly allocated to each neuron in the population (Moriarty & Miikkulainen, 1998).

Each genotype in the network blueprint population is comprised of a set of neuron pointers (i.e., address pointers) to neuron structures. The number of neurons in each network is fixed, depending on the complexity of the problem. Initially the neuron address pointers are assigned randomly to neuron structures (Moriarty & Miikkulainen, 1998).

SANE's evolutionary generational algorithm operates in two main phases – evaluation and recombination. The macro flow chart of the SANE algorithm is illustrated in Figure 4-7 (Moriarty & Miikkulainen, 1998).

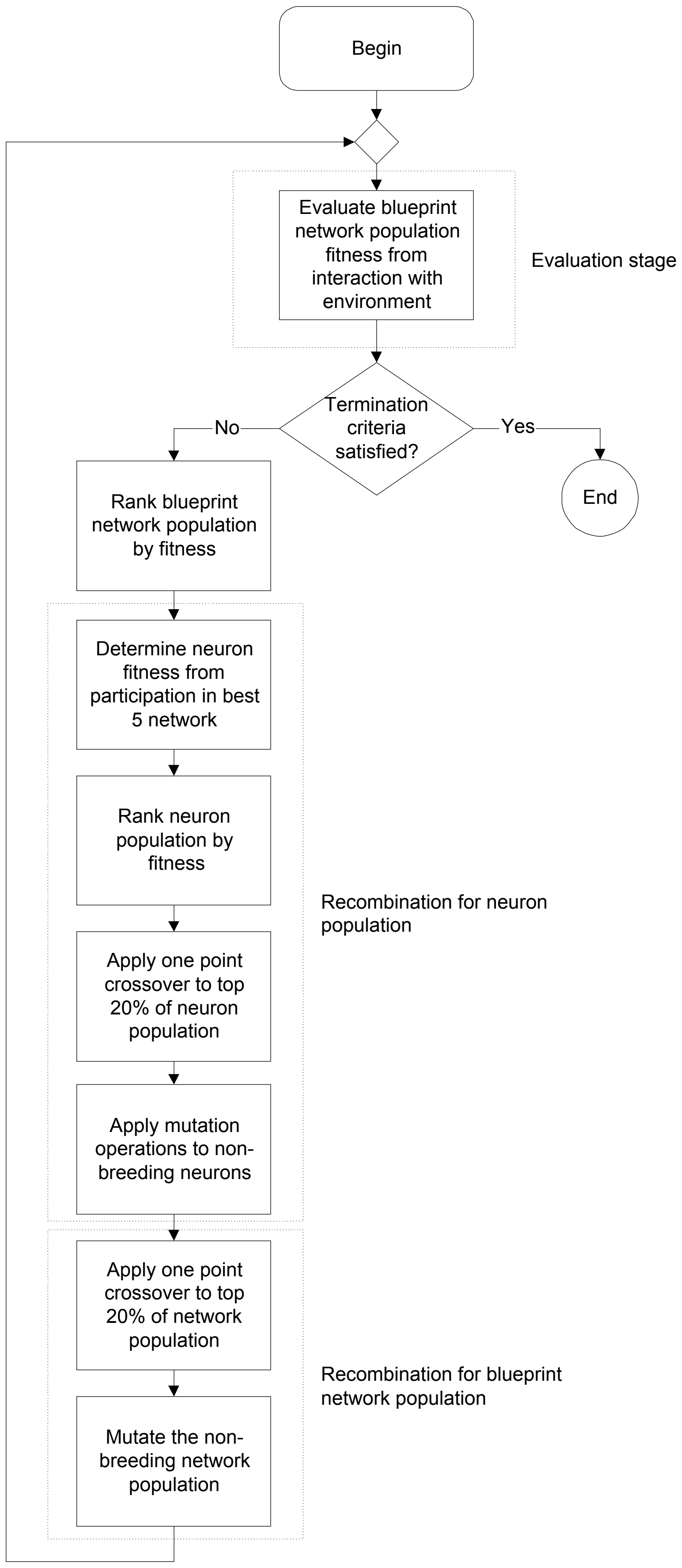

Begin
Evaluate blueprint network population fitness from interaction with environment
Evaluation stage
Termination criteria satisfied?
No
Yes
End
Rank blueprint network population by fitness
Determine neuron fitness from participation in best 5 network
Rank neuron population by fitness
Recombination for neuron population
Apply one point crossover to top 20% of neuron population
Apply mutation operations to non-breeding neurons
Apply one point crossover to top 20% of network population
Recombination for blueprint network population
Mutate the non-breeding network population

Figure 4-7 - Macro flow chart of the SANE algorithm.

4.3.1 Evaluation stage

The evaluation phase determines the fitness of each neuron and network in the populations. Network blueprints are evaluated based on reinforcement from direct interaction with either a dynamic simulation or real world environment (section 3.2). Figure 4-8 illustrates the process of evaluation for a single network genotype. The plant state is initially typically set to an initial condition that lies in the region around the desired set points. The dynamic state equations are solved, using fourth order Runga-Kutta, for a single sample period. The plant enters a new state, s_t, and the state variables and other inputs determine the neurocontroller's control action, a_t. The error from the desired state for the current sample period is calculated and stored for fitness calculation, r_t, purposes. The control action will determine the solution of the state equations for the next sample period. The current plant state is evaluated to determine if a premature failure criterion has been triggered. Should premature failure occur, a specified maximum error is assigned to the remaining time steps (i.e., sample periods) of the trial. Should premature failure not occur, the described sequence of events is repeated, until the evaluation trial terminates. A fitness value is assigned to the blueprint network based on the criterion specified by the objective function.

Each neuron genotype contained within a blueprint network is assigned a fitness based on the summed fitness of the best five networks that the neuron participated in. Utilising only the best five networks prevents an average or "aging" neuron with several pointers from dominating more effective neuron discoveries that have few pointers (Moriarty & Miikkulainen, 1998).

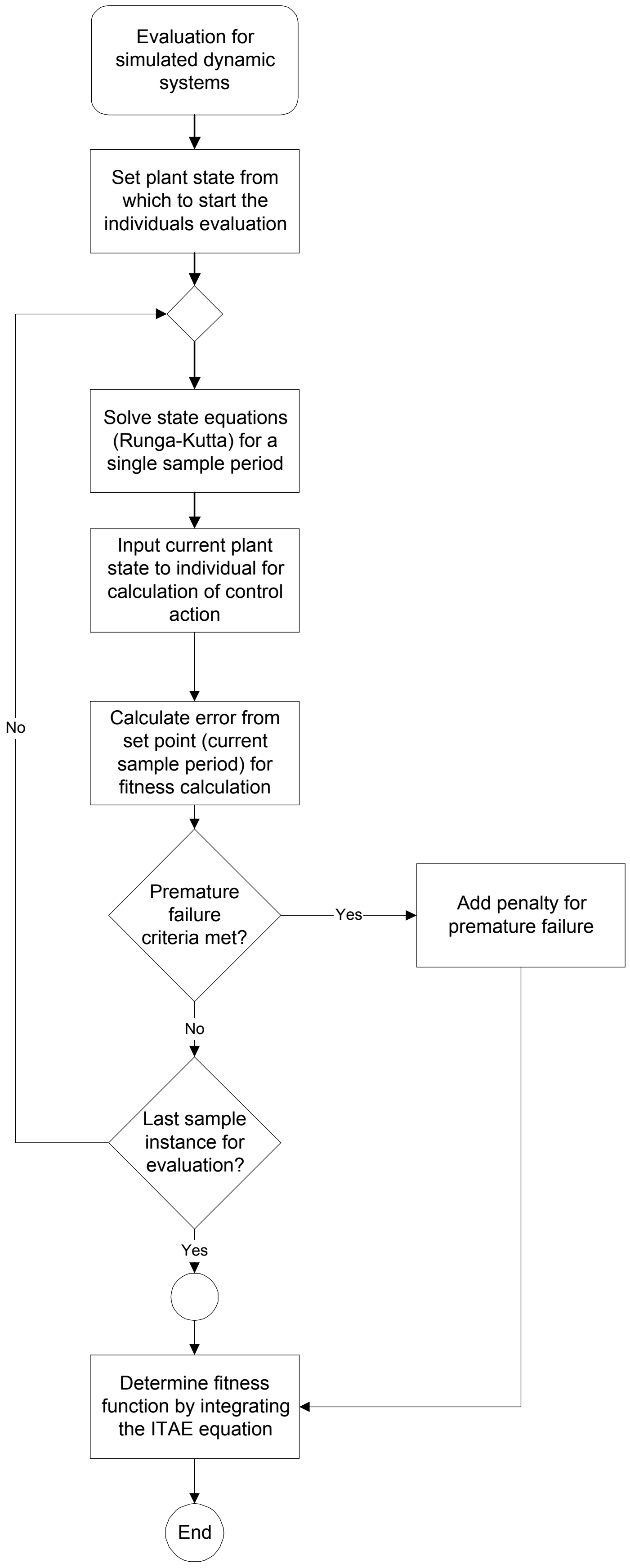

Evaluation for simulated dynamic systems
Set plant state from which to start the individuals evaluation
Solve state equations (Runga-Kutta) for a single sample period
Input current plant state to individual for calculation of control action
Calculate error from set point (current sample period) for fitness calculation
Premature failure criteria met?
Yes
Add penalty for premature failure
No
Last sample instance for evaluation?
No
Yes
Determine fitness function by integrating the ITAE equation
End

Figure 4-8 - Micro flow chart for evaluation phase for a single network genotype.

4.3.2 Recombination for the neuron population

After evaluation, the neuron population and the network blueprint population are ranked based on the assigned fitness. For each neuron in the top 20 [%] of the neuron population (i.e., elite population), a mate is selected randomly from the neurons that have a higher fitness than that particular neuron. Thus, the neuron ranked 3rd may only reproduce with the neurons ranked 2nd and first. Two offspring neurons are created from a one-point crossover operator. One of the offspring neurons is randomly selected to enter the population. The other offspring neuron is replaced randomly by one of the parent neurons, after which the offspring neuron is inserted into the population. Copying one of the parent neurons as the second offspring, reduces the effect of adverse neuron mutation on the blueprint population. The two offspring replace the most ineffective neurons in the population according to rank. This replaces a number of the most ineffective neurons (i.e, double the number of elite neurons) in the population after each generation. No mutation operator is used on the elite or breeding neurons (Moriarty & Miikkulainen, 1998).

Only the non-elite (non-breeding) portion of the population partakes in neuron mutation. For connection genes a 2 [%] probability exists that a connection may be randomly reassigned to either an input or output node. For weight genes a mutation 4 [%] probability exists for a random gaussian weight adjustment and a 0.1 [%] probability for a weight sign inversion. The original weight is modified within a standard deviation of 1.0 (Moriarty & Miikkulainen, 1998).

This aggressive, elitist breeding strategy is normally not incorporated in neurocontrollers evolution, as this would generally lead to premature convergence of the population. As SANE provides for pressure against convergence, SANE performs well with this aggressive strategy (Moriarty & Miikkulainen, 1998).

4.3.3 Recombination for the network population

Crossover in the blueprint population results in the exchange of address pointers to the neuron population. Should a parent point to a specific neuron, one of its children will consequently also point to that particular neuron (Moriarty & Miikkulainen, 1998).

To avoid convergence in the blueprint population, a twofold mutation strategy is incorporated. A 0.2 [%] probability exists that a pointer is reassigned to a random neuron in the neuron population. This promotes the use of neurons other than the neurons in the elite neuron population. A neuron that does not participate in any networks may so doing obtain a pointer from the blueprint population. Also, each offspring neuron is potentially better than or an exact copy of a parent neuron. The blueprint evolution takes advantage of this knowledge, by reassigning breeding neuron pointers to offspring neuron pointers with a 50 [%] probability. These two mutation operators preserve neuron pointers in the top blueprints, by not mutating any breeding (elite) networks (Moriarty & Miikkulainen, 1998).

As pointers are occasionally reassigned to offspring neurons, new neuron structures are evaluated. As neuron pointers are also reassigned to exact copies of parent neurons, some resilience against adverse mutation at the neuron level is incorporated. If pointers were not reassigned to neuron copies, several blueprint networks may point to the same neuron. Any mutation in that neuron would consequently affect each network that points to it. This copy strategy limits possible adverse effects to only a few blueprint networks. This is similar to schema promotion in standard evolutionary algorithms. As evolution progresses, highly fit schemata (neurons) become more prevalent in the population. Mutation to one copy of the schemata should not affect other copies in the population (Moriarty & Miikkulainen, 1998).

4.4 THE SUITABILITY OF SANE FOR EVOLVING NEURAL NETWORKS

Section 3.4 discusses several complications in evolving neural networks using genetic algorithms. These include scale-up and the structural/ functional mapping problem.

The evolution of large complete neural networks (i.e., GENITOR II) requires binary encodings that have long genotype (string) encodings. The longer the encoded solution, the greater the probability that the crossover operator becomes disruptive to the genetic search (section 3.4). SANE deals with this scale-up difficulty by encoding the neuron population utilising real value encoding (Whitley & Starkweather, 1990) as described in section 3.4. This reduces the genotype length and consequently results in less disruptive crossover. The evolution of individual neurons, instead of complete networks, also reduces the probability of disruptive crossover. Individual neurons have far shorter encodings than full network encodings, thus reducing the probability of relevant schema being separated by great genotype distances.

Genetic search relies probabilistically on offspring being potentially better than their parents. However, fully encoded neural networks are subject to the structural/functional mapping problem or competing conventions (section 3.4). In a neuron population, each neuron typically represents a single functionality in solving the complete task, disallowing the structural/functional mapping problem. Loss of neuron functionality due to the crossover operator is not a concern at the neuron level. Particularly, crossover between two functionally different, yet cooperating neurons, may result in two tasks being effectively combined into a single neuron. Neuron crossover thus promotes the emergence of generalists (section 3.5) in an environment with limited resources and network size (Smith et al., 1993). The top neurons in each generation are also copied unchanged to the next generation. Any detrimental crossover or mutation effects thus only affect the offspring neurons. A robust search is thus maintained at the neuron level.

SANE's network (blueprint) population is, however, susceptible to the structural/functional mapping problem (section 4.3.3). Blueprint networks point to functional neurons and crossover may disrupt useful combinations, should functionally similar neurons reside at different positions in the blueprint network encoding. SANE deals with this deficiency indirectly, by copying the top networks unchanged to the next generation. Should poor offspring result from two effective parents, the effect only slows the genetic search. Nevertheless, slower convergence is detrimental, as a slower search implies utilising more evaluations that may prove costly in practice.

The success of ERL in game tree search (Moriarty & Miikkulainen, 1998), controlling chaos (Weeks and Burgress, 1997), robot arm control and maze experiments has motivated its use in controlling non-linear unit operations (Conradie, 2000). In process control, ERL deals effectively with non-minimum phase behaviour, process time delays, operating constraints and actuator delays (Conradie, 2000), motivating its use in plant-wide control applications. For large problems like the Tennessee Eastman control challenge, SANE is slow in converging to an optimal neurocontrol structure. The SMNE algorithm speeds learning as described in section 4.5.

4.5 SYNERGY AND GENETIC DIVERSITY

This section introduces elements of a novel learning methodology, Symbiotic Memetic Neuro-Evolution (SMNE), for developing neural network controllers in a reinforcement learning framework.

4.5.1 Reinforcement learning

As discussed in section 3.2, reinforcement learning is a computational framework that allows automation of the learning process. Reinforcement learning is set apart from conventional non-linear control techniques by minimal reliance on explicit process knowledge. However, reinforcement learning relies heavily on the learning methodology that uses controller performance evaluations to direct the learning process. Evolutionary algorithms (EA) are robust global optimisation methods for solving complicated combinatorial tasks, such as determining optimal controller parameters. Evolutionary algorithms have been used effectively as learning methodologies in reinforcement learning frameworks. In neurocontrol, evolutionary reinforcement learning searches in a population of possible neural network controllers for a strategy that encompasses effective control actions in the chemical process. Neurocontrollers are comprised of collections of neurons, with each neuron specifying the weights from the input layer (sensor readings) to output layer (control actions). In an EA framework, effective neurocontrollers produce offspring, which propagates effective neurons (genetic material) in the population. This genetic propagation of effective neuron structures is key to solving the combinatorial nature of neurocontroller parameter estimation (Moriarty & Miikkulainen, 1998).

4.5.2 Memetic algorithms

EA's propagate effective neuron structures by varying the sample distribution in the solution space, depending upon the evaluation of the objective (fitness) function. This selection biases the search towards regions of the solution space where near optimal solutions have been discovered. Local refinements to these near optimal solutions could significantly accelerate arriving at an optimal solution. However, EA's are not suited to focusing local refinements in large combinatorial tasks. Genetic evolution may be augmented to facilitate local (neighbourhood) search via cultural evolution (Merz, 2000).

Analogous to genetic propagation, cultural transmission (i.e., bird song) is the evolutionary flow of information. However, there are significant differences between cultural and genetic evolution. In cultural evolution, improvements are seldom a result of copying errors or the exchange of co-adapted units of information. Clear-cut combination of exact ideas does not generally lead to innovation. An idea is rather blended with other similar ideas based upon perception and understanding. This blending process is the driving force towards innovation. Genetic evolution does not incorporate an innovative component, as experimentation (reproduction) with new information is governed by biased selection. A gene is not changed based on the quality of other similar genes. The individuals in cultural evolution are conscious entities that use one another's ideas in the search process, subject to cooperation and competition. Genetic evolution has no concern for individual genes, but focuses on improving the population by propagating effective gene combinations (Merz, 2000).

Memetic algorithms (MA) are evolutionary algorithms that use cultural evolution for local search (LS). The local search is applied to solutions in each generation of the EA, creating a process of lifetime learning. The EA searches globally for regions containing significant optima, while the LS searches these regions for the local optimum. The EA is thus responsible for exploration, whilst the LS governs exploitation. A balance between exploration and exploitation ensures that the minimum number of evaluations is employed in finding the global optimum. This balance is dependent on the synergy between lifetime learning and evolution (Merz, 2000).

LS aids the evolutionary process by smoothing the fitness landscape. LS exploits the local fitness landscape, which absolves the EA from devoting resources to searching in areas of local complexity on the fitness surface. This smoothing essentially involves a discretisation of the fitness landscape. Consider the optimisation of the fitness landscape in Figure 4-9.

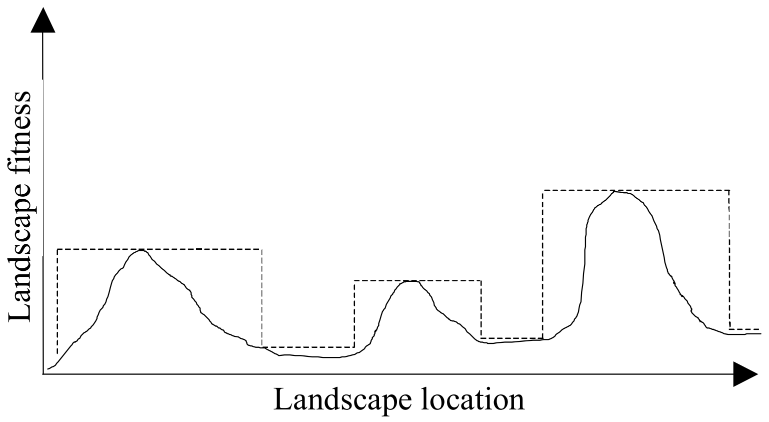


Figure 4-9 - Smoothing of the fitness landscape by local search, thereby reducing the complexity of the EA's solution space.

Assume that any EA solution, located on one of the slopes on the three peaks, is able to locate the local maximum through LS. The EA's task is simplified considerably, in that it only needs to locate three regions of the search space. The dashed lines in Figure 4-9 indicate these three discrete regions. With the added local search capability, the complexity of the EA's solution space is reduced significantly. The plasticity afforded by lifetime learning makes it easier for the EA to climb to peaks in the fitness landscape (Merz, 2000).

Therefore, the EA of a memetic algorithm should not generate multiple solutions in the neighbourhood of a single optimum, but should maintain a diverse (wider) search in the solution space. Thereby, the EA aids the LS by bordering regions (sub-spaces) of the fitness landscape that contain significant optima. Such regions become prime candidates for exploitation by local search algorithms. A synergetic effect, which accelerates evolution, thus exists in an evolving population of individuals, where the individuals are also exposed to learning during their lifetime (Merz, 2000).

A key element to maintaining such synergy is a diversification mechanism in the EA. Genetic diversity is required to continue a global search. Global reliability, which promises convergence to the global optimum, is required to ensure that every region of the solution space is effectively explored (Merz, 2000).

4.5.3 Effective genetic diversification

Genetic diversity prevents convergence to a local optimum and allows continued genetic search, assuring global reliability. The continued introduction of informational variety is critical for effective exploitation by cultural evolution. Numerous methods for maintaining genetic diversity have been proposed; such as crowding, distributed sub-populations with migration, local mating and explicit fitness sharing (section 3.5). These methods are effective in slowing convergence, but have been unable to sustain a diverse dynamic equilibrium in the EA's population.

Implicit fitness sharing entails the search for partial solutions, which cooperate to encode the complete solution. In neuro-evolution, individual neurons are partial solutions to the complete solution (neural network). Neurons that compete to perform the same task, compete for the same rewards, namely weak cooperation. Neurons that do not overlap in their tasks, are cooperating in an indirect manner, namely strong cooperation. Strong cooperation is symbiotic in nature (section 3.5.5).

Strong cooperation maintains high quality diversity in the face of significant selection pressure, by balancing convergence with the restorative force of niching pressure. A niching phenomenon also implies several parallel searches for partial solutions, which should prove more effective than a single search for the complete solution (section 3.5.5).

4.6 SYMBIOTIC MEMETIC NEURO-EVOLUTION (SMNE)

Implicit fitness sharing and the synergetic effect of memetic algorithms, may be combined to enhance global reliability and accelerate evolution in complex combinatorial tasks. This section introduces a novel memetic algorithm, Symbiotic Memetic Neuro-Evolution (SMNE), for developing neurocontrollers in a reinforcement learning framework. A symbiotic genetic algorithm (Figure 4-10) is employed to ensure global reliability, while performing an aggressive explorative search. Particle Swarm Optimisation (PSO), a cultural evolution method, is used for local exploitative search and refinements after each EA generation. Compared to other evolutionary approaches, this synergetic effect accelerates and improves the automated acquisition of process control knowledge from non-linear dynamic models.

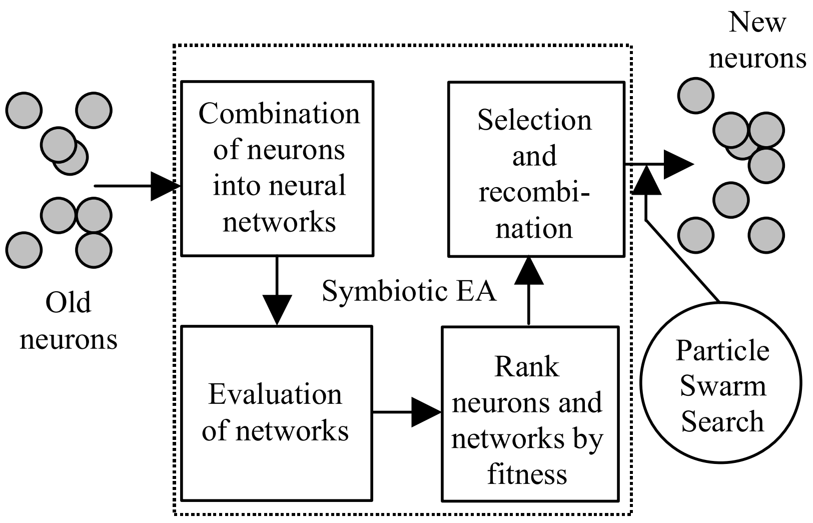

New neurons
Combination of neurons into neural networks
Selection and recombi-nation
Symbiotic EA
Old neurons
Evaluation of networks
Rank neurons and networks by fitness
Particle Swarm Search

Figure 4-10 - Flow diagram for a single generation in the Symbiotic Memetic Neuro-Evolution algorithm.

4.6.1 Symbiotic evolutionary algorithm

Similar to the SANE algorithm (section 4.3), the symbiotic EA (dashed box in Figure 4-10) maintains both a neuron and a network population. Each member of the neuron population encodes a hidden neuron, with weights from the input layer to the output layer. While SANE maintains a single neuron population, SMNE's neuron population is comprised of a number of sub-populations. The network population is constructed from the neuron population. Each network is a collection of pointers to the neuron sub-populations. Each position in a network's hidden layer is filled from a different neuron sub-population.

Competing conventions is avoided in the SMNE's network population, as each network position points to a particular sub-population (compare section 4.4). Competing conventions is also avoided in the neuron population, as each weight connects to a fixed input or output throughout evolution. Real-value encoding of neuron weights also ensures that the crossover location is at the gene (weight) boundaries. This causes less gene disruption during crossover. Unchanged genes are thus carried into the next generation, focusing the search.

Each network is evaluated in the reinforcement learning task (i.e., process control) and assigned a fitness value based on the control performance criteria. High network fitness reflects superior performance in the control task. The network population is ranked after evaluation. The neuron fitness assignment implements implicit fitness sharing. Each neuron is assigned a fitness value, based on the summed fitness of the five best networks in which it participated (Moriarty & Miikkulainen, 1998). High neuron fitness reflects a neuron's ability to cooperate with other neurons in different sub-populations. Rewarding neuron cooperation with high fitness, induces niching pressure in the sub-populations. Strong cooperation between sub-populations facilitates the search for partial solutions that comprise the complete solution. Each sub-population thus serves as a container in which a niche may emerge. The niching pressure retains genetic diversity in the neuron population, allowing the genetic search to continue. The neuron population is ranked after evaluation. Recombination and reproduction is based on the network and neuron ranking (Moriarty & Miikkulainen, 1998).

One-point crossover is applied to the elite neuron population (top 20 [%]). The elite neurons breed across the sub-populations, thereby exploring the solution space between current niches. Each elite neuron randomly selects (on rank) a mate that has a higher fitness than itself. Two effective parents should produce offspring with similar or superior performance. As the best elite neurons are more likely situated in different sub-populations (strong cooperation), their offspring attempt combining two functionalities into a single neuron. This may free sub-populations to pursue other niches. Crossover in the less fit elite neurons has a greater probability of selecting a parent from the same sub-population, which focuses the genetic search (weak cooperation). Each offspring neuron is copied to a neuron sub-population, depending on the gene contribution from each parent. The offspring neurons replace the worst neurons in each sub-population. Mutation is applied, with low probability (2 [%]), to the remainder of the neuron population.

An elite network population (top 20 [%]) retains knowledge of effective neuron combinations (Moriarty & Miikkulainen, 1998). The elite network population's reproduction operator replaces a neuron pointer with one of its offspring that was copied to the same sub-population. This reproduction operator applies, with 25 [%] probability, to all the neuron pointers in the elite network population. The offspring networks replace the worst networks. The remaining networks are constructed randomly from the sub-populations, with a propensity for selecting offspring neurons of the elite neuron population. This scheme ensures that neurons not selected in the previous generation, obtain pointers and therefore a fitness evaluation.

4.6.2 Particle Swarm Optimisation

A local refinement search, Particle Swarm Optimisation (PSO), augments SMNE's symbiotic EA. PSO is implemented after each EA generation, as illustrated in Figure 4-10, thereby inducing lifetime learning. PSO is a population based optimisation method loosely based on the social behaviour of flocks (swarms) of birds. PSO updates each individual based on the collective experience of the swarm particles in the solution space (Shi & Eberhart, 1999):

$$v_{id} := \omega \cdot v_{id} + c_1 \cdot rand() \cdot (p_{id} - x_{id}) + c_2 \cdot rand() \cdot (p_{gd} - x_{id}) \tag{4-3}$$

$$x_{id} := x_{id} + v_{id} \tag{4-4}$$

where v_{id} is a particle's velocity vector, ω is the inertia weight, c_1 and c_2 are constant parameters and x_{id} is a particle's position vector. Each particle retains partial knowledge of its own best position, p_{id}, and the position of the best swarm particle, p_{gd} (equation 4-3). Based on these two knowledge components, each particle updates its velocity vector to determine its next position (equation 4-4) (Shi & Eberhart, 1999). PSO shares numerous characteristics with cultural algorithms. The swarm's movement in the solution space is akin to cultural transmission and the innovative blending of ideas. Also, each particle's momentum protects against entrapment in a local optimum.

Each particle thus blends its own experience and that of the best swarm particle in a unique manner. PSO assumes that the best swarm particle is located in a region of the solution space that contributes to solving the control task. Each particle moves uniquely in the general direction of the best swarm particle. This may lead to the discovery of superior, adjacent regions of the solution space. A new best swarm particle consequently moves the swarm in a new direction. PSO thereby involves cooperation as a result of shared experience and competition for superior fitness (Shi & Eberhart, 1999).

SMNE's particle swarm implementation incorporates a small inertia weight ($\omega = 0.4$ in equation 4-3) to facilitate local search. The parameters c_1 and c_2 (equation 4-3) are also equal to 0.5 (conventionally 2.0), which ensures an exploitative search. Each neuron sub-population contains a separate PSO implementation. The PSO's neuron weight changes are Lamarckian, that is, the weight changes update the genes. The best neuron in a sub-population is the best particle in its swarm. PSO refines each partial solution, by sharing the best neuron's control knowledge with other neurons in its sub-population.

Local search (LS) should only be applied to evolutionary solutions where it will be the most beneficial. A Lamarckian LS may also begin to dominate the genetic population, causing a loss in diversity. LS should thus only be applied to a sub-set of the total population (Merz, 2000). Therefore, the neurons of the elite network population constitute the candidates for LS. These neurons are presumably located in close proximity to the regional sub-spaces that contribute to the search for partial solutions. Complete LS may also involve a large number of evaluations. The LS is only applied for the limited number of five steps. This ensures that evaluations are effectively utilised, after each evolutionary step. This partial local optimisation is in keeping with cultural evolution, ensuring that each particle modifies the swarm knowledge uniquely. Applying LS for a limited number of steps, also avoids a loss of diversity in the elite network population's neurons.

As discussed in section 4.5, the symbiotic EA and the PSO of SMNE result in a synergetic search algorithm that allows effective discovery and refinement of partial solutions. The SMNE algorithm was subsequently applied to a challenging real world problem.

4.7 AGRAWAL BIOREACTOR FLOW SHEET OPTIMISATION - COORDINATED PROCESS DESIGN AND CONTROLLER DEVELOPMENT

Fermenters may be extremely difficult to control, as their dynamic behaviour is invariably non-linear and model parameters may vary unpredictably. Accurate process models are rarely available owing to the complexity of the underlying biochemical process systems. Also, the process state is difficult to evaluate accurately, owing to a lack of reliable biosensors (Henson & Seborg, 1992).

Agrawal et al. (1982) modelled the behaviour of microbial cultures within the framework of lumped kinetic models, which has proven useful in industrial applications. Cell growth rates are frequently inhibited by large substrate concentrations. Agrawal et al. (1982) investigated the effect of this inhibition on steady-state and periodic solutions, by using the following one-hump growth model:

$$\mu\{S\} = k \cdot S \cdot e^{(-S/K)} \qquad \textbf{(4-5)}$$

where μ is the specific growth rate (maximum at $S = K$) and S is the substrate concentration. The specific substrate consumption rate, σ, is related to the specific growth rate through the yield coefficient, Y.

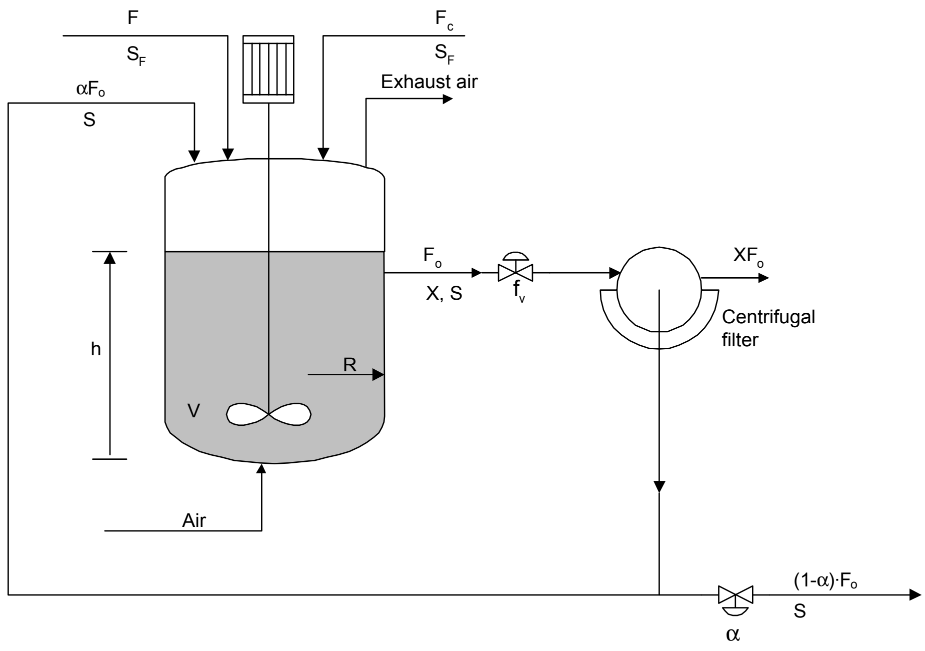


Figure 4-11 - Bioreactor flow sheet to be optimised by the SMNE algorithm from both a design and control perspective.

For the CSTR in Figure 4-11, the cell and substrate mass balances are described by the following coupled non-linear differential equations 4-6 to 4-8. The model variables are defined in Table 4-2.

$$\frac{dh}{dt} = \frac{F + F_c - (1-\alpha)\cdot f_v \cdot h^{0.5}}{\pi \cdot R^2} \tag{4-6}$$

$$\frac{dX}{dt} = -\frac{\left(F + F_c + \alpha \cdot f_v \cdot h^{0.5}\right)}{\pi \cdot R^2 \cdot h} \cdot X + k \cdot S \cdot e^{(-S/K)} \cdot X \tag{4-7}$$

$$\frac{dS}{dt} = \frac{F \cdot (S_F - S) + F_c \cdot (S_c - S)}{\pi \cdot R^2 \cdot h} \cdot - \frac{k \cdot S \cdot e^{(-S/K)}}{a + b \cdot S} \cdot X \tag{4-8}$$

Table 4-2 - Model parameters and variables for bioreactor flow sheet.

Description	*Formulation*	*Unit*
Nominal Substrate Feed Flow rate	F	$[\mathrm{m}^3 \cdot \mathrm{min}^{-1}]$
Concentrated Substrate Feed Flow rate	F_c	$[\mathrm{m}^3 \cdot \mathrm{min}^{-1}]$
Reactor outlet valve position	f_v	$[\mathrm{m}^{2.5} \cdot \mathrm{min}^{-1}]$
Reactor liquid height	h	[m]
Reactor radius	R	[m]
Nominal Substrate Feed Concentration	S	$[\mathrm{kmol} \cdot \mathrm{m}^{-3}]$
Concentrated Substrate Feed Concentration	S_c	$[\mathrm{kmol} \cdot \mathrm{m}^{-3}]$
Cell mass concentration	X	$[\mathrm{kg} \cdot \mathrm{m}^{-3}]$
Recycle Ratio	α	-

The flow sheet in Figure 4-11 needs to be optimised by maximising ϕ in equation 4-9, with the operating profit (P) described by equation 4-10 and the fixed capital cost (FCI) described in equation 4-11. The rate of return (ROR) is fixed at 15 [%]. The reactor height is also limited by the constraint in equation 4-12. The various cost coefficients and process design parameters are presented in Table 4-3.

$$\max_{h,X,S,\alpha,Fc,F,f_v} \phi = P - ROR \cdot FCI \tag{4-9}$$

$$P = \left(c_1 \cdot X \cdot f_v \cdot h^{0.5} - c_2 \cdot F - c_3 \cdot F_c - \frac{c_4 \cdot X \cdot f_v \cdot h^{0.5}}{3600} \right) \cdot P_t \tag{4-10}$$

$$FCI = 1.18 \cdot \left[F_{bm,v} \cdot B_v \cdot V^{0.724} + F_{bm,c} \cdot B_c \cdot \left(X \cdot f_v \cdot h^{0.5} \right)^{0.444} \right] \tag{4-11}$$

$$1.9 \cdot R - h \geq 0 \tag{4-12}$$

Table 4-3 - Parameter and cost coefficients for flow sheet optimisation.

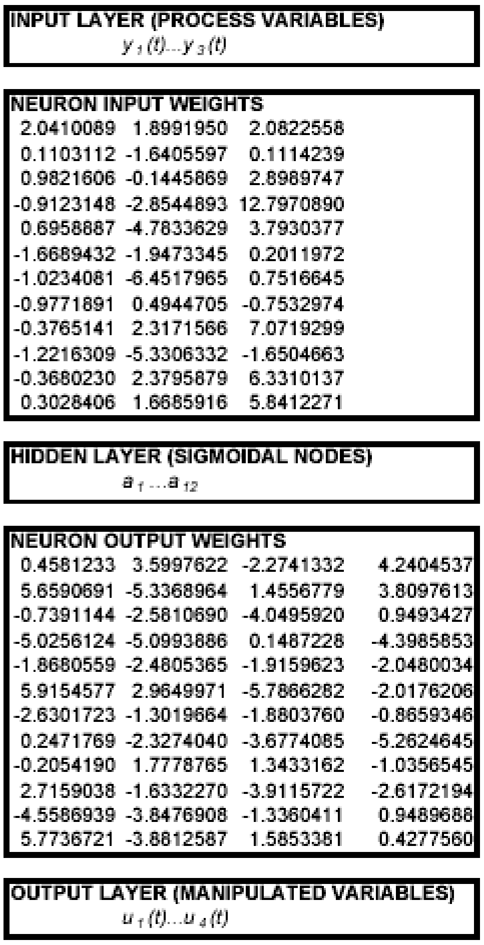
Parameter	*Unit*	*Value*	*Cost Coef.*	*Unit*	*Value*
R	m	3.00	c_1	$\$\cdot kg^{-1}$	5.00
S_F	$kmol\cdot m^{-3}$	0.3	c_2	$\$\cdot m^{-3}$	10.80
S_c	$kmol\cdot m^{-3}$	1.00	c_3	$\$\cdot m^{-3}$	200.00
a	$kg\cdot kmol^{-1}$	5.4	c_4	$(\$\cdot h^{-1})/(kg\cdot s^{-1})$	0.04
b	$kg\cdot m^{3}\cdot kmol^{-2}$	180	$F_{bm,v}$	-	4.5
k	$m^{3}\cdot kmol\cdot s^{-1}$	1.00	$F_{bm,c}$	-	3.4
K	$kmol\cdot m^{-3}$	0.12	B_v	$\$/(m^3)^{0.724}$	1836.5
			B_c	$\$/(kg\cdot min^{-1})^{0.444}$	54 325

SMNE was used to develop a neurocontroller with 3 input nodes, 12 hidden nodes and 4 output nodes (Table 4-4 & Table 4-5). The three input nodes were the system state variables, namely the cell mass concentration (X), substrate concentration (S) and reactor liquid level (h). The four output nodes were the nominal feed flow rate (F), the concentrated feed flow rate (F_c), the recycle fraction (α) and the reactor outlet valve position (f_v) (which reflects the valve coefficient (C_v) multiplied by the flow characteristic of the valve). The four possible manipulated variables were bounded as follows: $0 \leq F \leq 10$; $0 \leq F_c \leq 10$; $0 \leq f_v \leq 3.5$ and $0 \leq \alpha \leq 1$. Twelve hidden nodes over-specify the mapping of state variables to control actions, but SMNE effectively prunes unnecessary connections and neurons from the neurocontroller solution.

Table 4-4 – Ranges of process and manipulated variables, normalised between 0 and 1 as inputs and outputs of neurocontroller.

Neurcontroller inputs & outputs	*Process & manipulated variables*	*Unit*	*Normalised minimum*	*Normalised maximum*
y_1	h	m	0	5.7
y_2	X	$kg \cdot m^{-3}$	0	17.82
y_3	S	$kmol \cdot m^{-3}$	0	1
u_1	α	-	0	0.99
u_2	F_c	$m^3 \cdot min^{-1}$	0	10
u_3	F	$m^3 \cdot min^{-1}$	0.1	10
u_4	f_v	$m^{2.5} \cdot min^{-1}$	0	3.5

Table 4-5 – Optimal SMNE neurocontroller based on equations 4-9 to 4-12 and the parameters in Table 4-3 (sample frequency of 10.5 [min]).



INPUT LAYER (PROCESS VARIABLES)

$y_1(t) \dots y_3(t)$

NEURON INPUT WEIGHTS

2.0410089	1.8991950	2.0822558
0.1103112	-1.6405597	0.1114239
0.9821606	-0.1445869	2.8989747
-0.9123148	-2.8544893	12.7970890
0.6958887	-4.7833629	3.7930377
-1.6689432	-1.9473345	0.2011972
-1.0234081	-6.4517965	0.7516645
-0.9771891	0.4944705	-0.7532974
-0.3765141	2.3171566	7.0719299
-1.2216309	-5.3306332	-1.6504663
-0.3680230	2.3795879	6.3310137
0.3028406	1.6685916	5.8412271

HIDDEN LAYER (SIGMOIDAL NODES)

$a_1 \dots a_{12}$

NEURON OUTPUT WEIGHTS

0.4581233	3.5997622	-2.2741332	4.2404537
5.6590691	-5.3368964	1.4556779	3.8097613
-0.7391144	-2.5810690	-4.0495920	0.9493427
-5.0256124	-5.0993886	0.1487228	-4.3985853
-1.8680559	-2.4805365	-1.9159623	-2.0480034
5.9154577	2.9649971	-5.7866282	-2.0176206
-2.6301723	-1.3019664	-1.8803760	-0.8659346
0.2471769	-2.3274040	-3.6774085	-5.2624645
-0.2054190	1.7778765	1.3433162	-1.0356545
2.7159038	-1.6332270	-3.9115722	-2.6172194
-4.5586939	-3.8476908	-1.3360411	0.9489688
5.7736721	-3.8812587	1.5853381	0.4277560

OUTPUT LAYER (MANIPULATED VARIABLES)

$u_1(t) \dots u_4(t)$

SMNE searched the solution space for the neurocontroller with maximum economic return (equation 4-9). No set point values were provided, as SMNE was required to find the optimal operating point and develop an appropriate neurocontroller simultaneously.

SMNE optimised the design flow sheet for control at the steady state values as presented in Table 4-6. This optimal steady state is similar to that found with the PRODOC (PRocess Design, Operations and Control) research tool that uses homotopy-continuation methods for optimisation (Brengel & Seider, 1992). PRODOC uses the steady state process model to optimise the design objective function. As the design optimisation proceeds, the controllability using model predictive control is evaluated at various steady states on the search trajectory using the dynamic model. PRODOC coordinates the two non-linear programming (NLP) problems for the design optimisation and model predictive control development (MPC), which allows for a trade-off between profitability and controllability. In SMNE, the dynamic model is used directly to find the optimum operating point and develop a neurocontroller. As the genetic search progresses many neurocontrollers are developed in various operating regimes and recombined, until the neurocontroller with the greatest economic reward is found. In SMNE, a single optimisation is thus performed, while PRODOC coordinates two optimisation problems. Furthermore, Seider & Brengel (1992) found that convergence could not be guaranteed for all starting points in the PRODOC search, while SMNE's methodology is robust in that finding the global optimum is assured. As with PRODOC, SMNE may reduce the occurrences of overdesign via process design and control optimisation.

Table 4-6 - Optimum values for design for bioreactor flow sheet.

Variable	*Unit*	*Brengel & Seider (1992) optimal steady state*	*SMNE optimal steady state*
h	m	5.70000	5.69070
X	$kg\cdot m^{-3}$	4.89616	4.89693
S	$kmol\cdot m^{-3}$	0.13009	0.13003
α	-	0.00000	0.00000
F_c	$m^3\cdot min^{-1}$	0.00000	0.00000
F	$m^3\cdot min^{-1}$	7.09087	7.08030
f_v	$m^{2.5}\cdot min^{-1}$	2.97004	2.96775

A bifurcation analysis of the fermenter process model reveals that the optimal steady state (Table 4-6) corresponds to an unstable steady state in the open loop (Brengel & Seider, 1992). In a conservative design approach, it may seem prudent to operate at the highest cell concentration, but to avoid control difficulties not in near proximity to the region of instability. Selecting to operate in the region of stable steady state attractors, however, produces lower cell concentrations with greater residence times. At the global economic optimum (i.e., an open loop unstable steady state), the venture profit exceeds that at the most economic stable steady state (i.e., local optimum) by 31.6 [%] (for a bioreactor with similar volume). Operating at the optimal unstable steady state, the net present value is considerably greater for the same capital investment. The SMNE neurocontroller design approach does not limit the development of a controller to the open loop stable region of the process. Controller robustness in complex, unstable regions may thus be achieved with a large degree of autonomy.

The solution found by SMNE has a number of positive design implications. The reactor volume was selected as the largest within the design constraint of equation 4-12. The bioreactor volume had thus been appropriately determined by the SMNE algorithm, given the capital cost implication to the proposed project as in equation 4-11. The centrifugal filter has also been sized, based on the capital cost of the proposed filter equipment at the desired filter duty. A concentrated substrate feed stream and a recycle stream are unnecessary from a control perspective for the nominal process conditions. Omitting these streams reduces the capital cost of the proposed flow sheet. Also, the inlet and outlet valves may be appropriately sized based on the flow rates that assure the appropriate residence time in the bioreactor.

A typical neurocontroller response from a stable steady state initial condition ($X = 2.736$ [$kg \cdot m^{-3}$], $S = 0.025$ [$kg \cdot m^{-3}$], $h = 4.5$ [m]) is presented in Figure 4-12 to Figure 4-13. This typical response may be observed from any possible initial condition, as the neurocontroller effectively generalises to initial conditions not encountered during learning. From Figure 4-12 it is evident that the constraint in equation 4-12 represents a hard active constraint, in that the optimum reactor volume is at the upper limit of the *h(R)* constraint. During learning, should the constraint, $h \leq 5.7$ [m], have been exceeded, premature failure would have occurred for that evaluation. The best neurocontroller thus learnt to maintain the level of the bioreactor at 5.691 [m] during an evaluation. The response in the reactor level thus had no overshoot (i.e., overdamped response) and the approach to the optimum steady state was gradual, so as to avoid the hard constraint limit. The response for the cell mass reveals an inverse response (i.e., non-minimum phase behaviour) caused by the presence of a right half plane zero in the process model. Process systems that exhibit non-minimum phase behaviour may cause linear controllers (i.e., PID controllers) to suffer from poor robustness, making optimal performance difficult to attain. The manipulated variable responses in Figure 4-13 are also gradual and the final control elements are thus not subjected to excessive control actions. The SMNE algorithm developed a neurocontroller with an annual venture profit of $48.477 million for the model parameters and cost coefficients as presented in Table 4-3. The instantaneous operating profit settles to the global economic optimum of 96.86 [$\$ \cdot min^{-1}$] as in Figure 4-14.

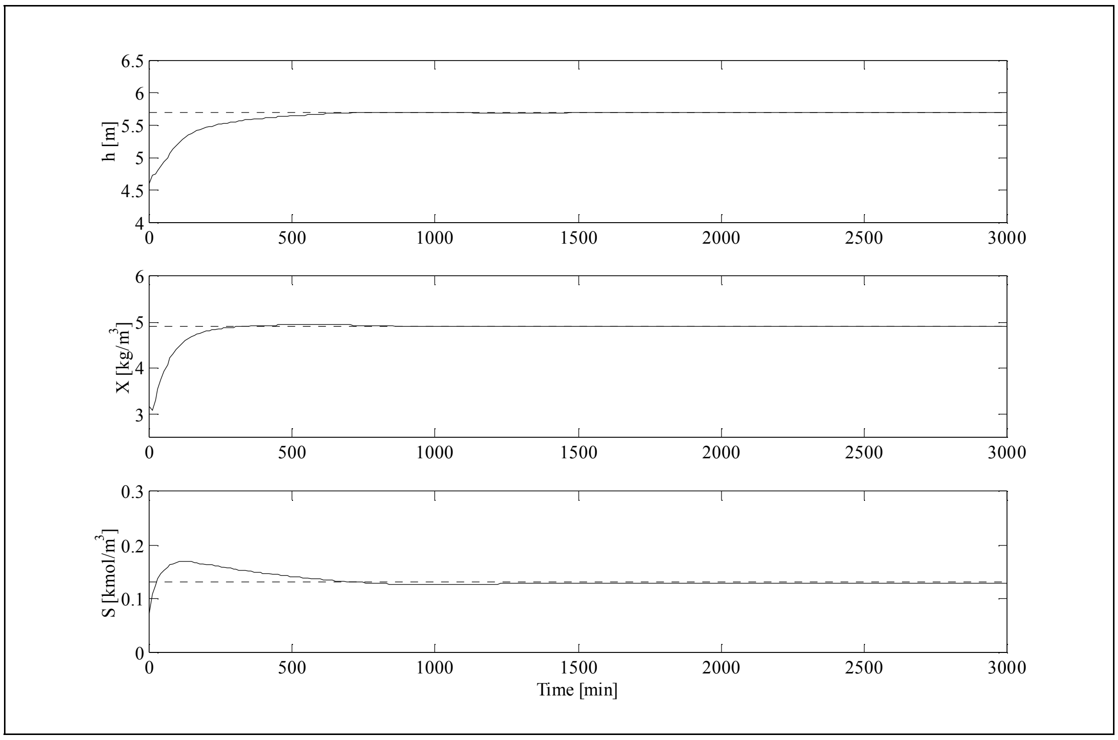


Figure 4-12 - Response of the state variables from initial condition X = 2.736 [kg·m^{-3}], S = 0.025 [kg·m^{-3}], h = 4.5 [m].

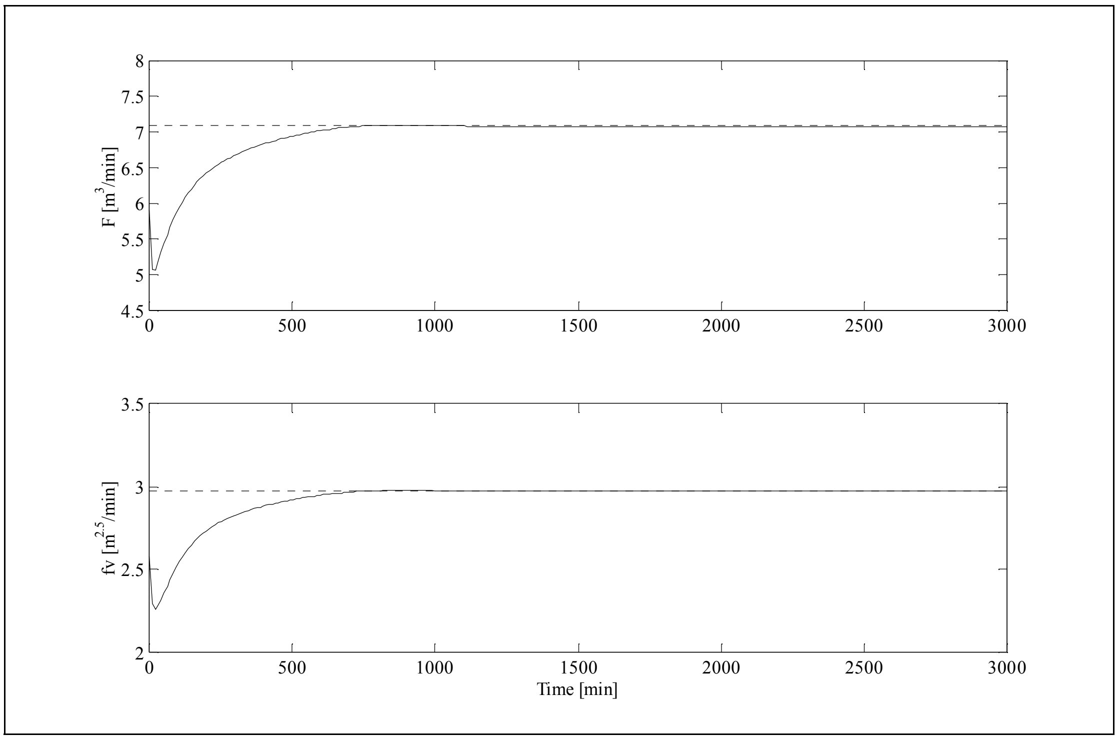


Figure 4-13 - Control action of the manipulated variables from initial condition X = 2.736 [kg·m^{-3}], S = 0.025 [kg·m^{-3}], h = 4.5 [m].

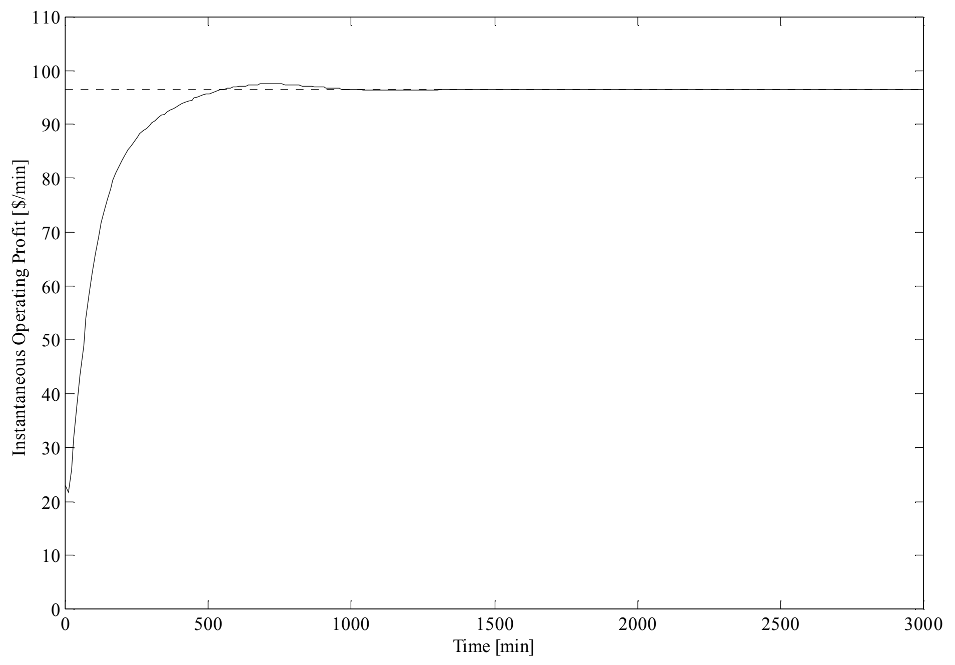


Figure 4-14 - Instantaneous economic optimum without disturbances, starting from initial condition X = 2.736 [kg·m^{-3}], S = 0.025 [kg·m^{-3}], h = 4.5 [m].

Bioreactors are typically subject to uncertainty in the substrate feed concentration, S_F, owing to the complexity of nutrient media. To simulate this uncertainty, gaussian noise was introduced around the nominal value of $S_F = 0.3$ [kmol·m^{-3}] with a standard deviation of 0.025 [kmol·m^{-3}] at every 10 sample period intervals (Figure 4-15). This is a difficult disturbance to reject in a CSTR reactor, as the unmeasured disturbance has an immediate impact (step change) on the substrate concentration in the uniformly mixed reactor. The state variable and control action responses are illustrated in Figure 4-17 and Figure 4-18. As the presence of this disturbance was not simulated during learning, any control action that attempts to reject the disturbance is an indication of the neurocontroller's ability to generalise in the presence of uncertainty. From Figure 4-16, the neurocontroller remains robust in the presence of significant uncertainty and remains in close proximity to the economic optimum for $S_F = 0.3$ [kmol·m^{-3}]. For $S_F <$ 0.3 [kmol·m^{-3}] the instantaneous operating profit can not be maintained at 96.86 [\$·min^{-1}], as the maximum attainable cell mass concentration is well below 4.89 [kg·m^{-3}]. This is reflected in the reduced instantaneous profit (Figure 4-16) as S_F is reduced below 0.3 [kmol·m^{-3}] (Figure 4-15). However, the optimal instantaneous profit is above 96.86 [\$·min^{-1}] for $S_F > 0.3$ [kmol·m^{-3}]. From Figure 4-17 it is thus evident that the neurocontroller is unable to take advantage of an increase in S_F to increase the operating profit.

The neurocontroller is thus able to maintain robust performance in the presence of a feed disturbance, but is unable to track the optimum instantaneous operating profit with changes in the model parameters (such as S_F). The neurocontroller is thus specialised for optimal performance with S_F = 0.3 [kmol·m^{-3}]. Unmeasured disturbances, however, reduce the ability of the neurocontroller to maintain optimal performance in the light of equation 4-9. The net effect of a disturbance in S_F (Figure 4-15) on the profitability of the process is a reduction in the annual venture profit to $38.36 million from the nominal $48.477 million. On-line adaptation using Adaptive Neural Swarming (ANS) offers opportunities for improving profitability (section 4.9).

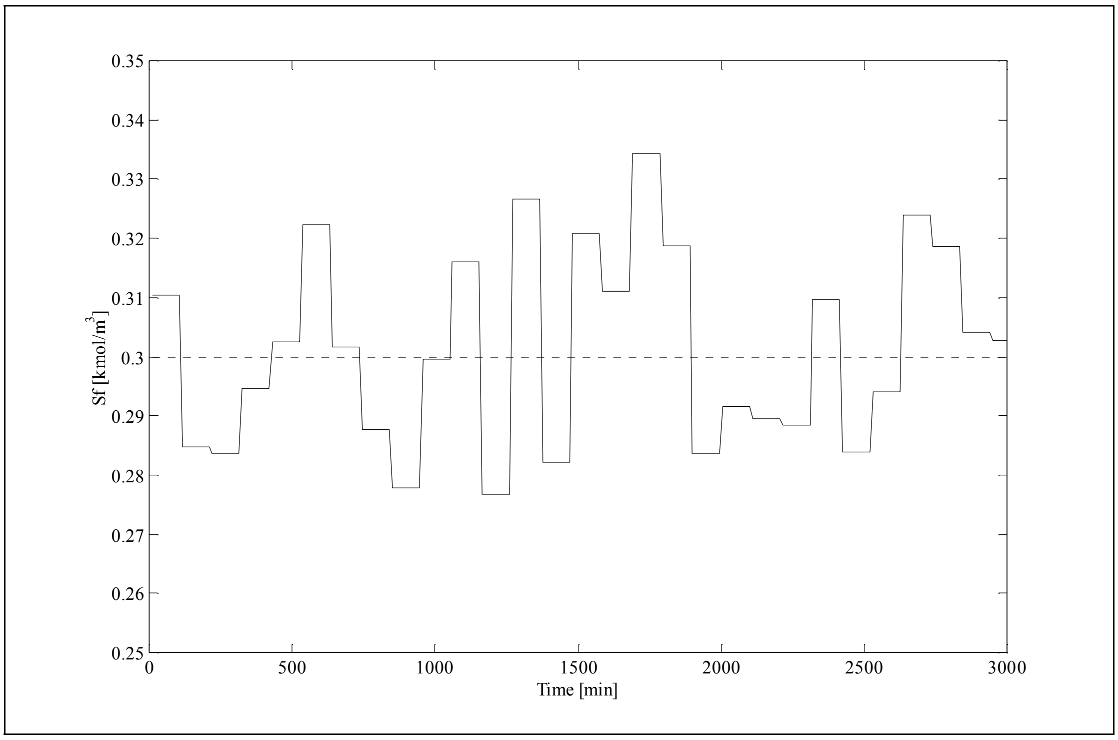


Figure 4-15 - Gaussian noise disturbance in the substrate feed concentration. (----) Nominal S_F.

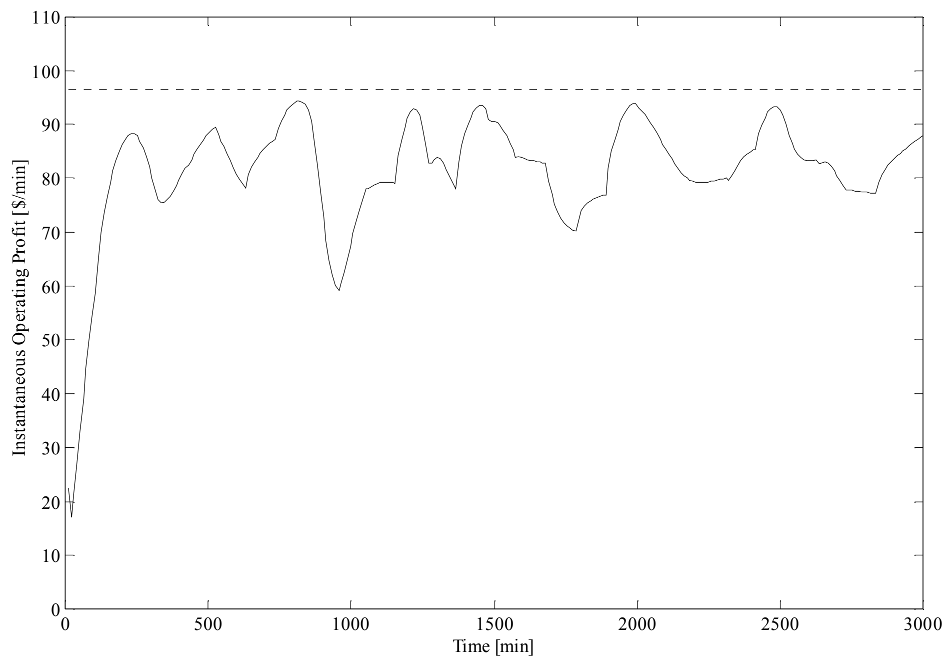


Figure 4-16 - Instantaneous operating profit for the introduced gaussian noise on S_F. (-----) Optimal instantaneous operating profit for $S_F = 0.3$ [kmol·m^{-3}].

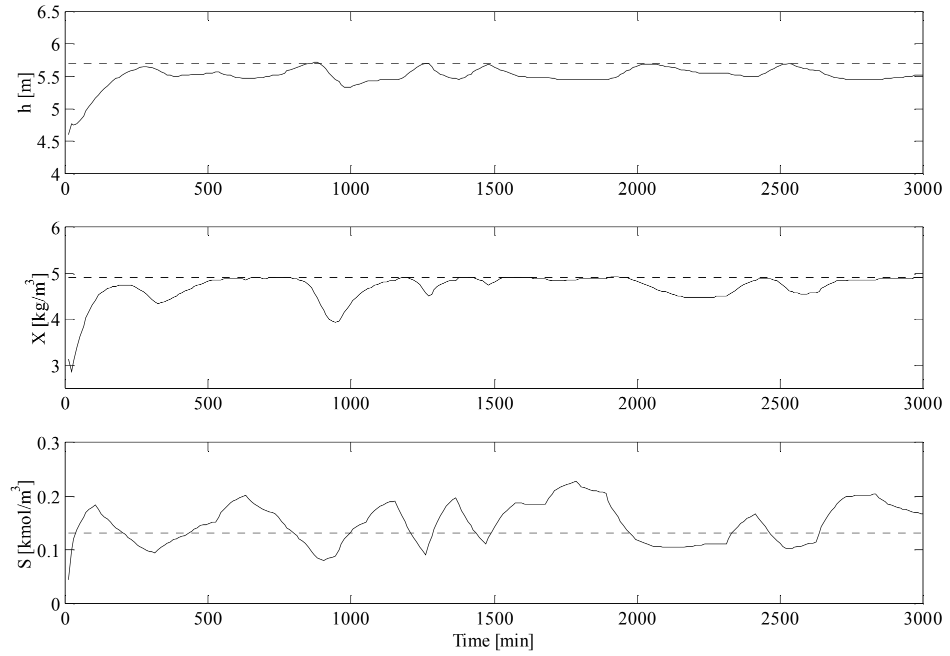


Figure 4-17 - Response of the state variables to gaussian noise on the S_F. (-----) Economic optimal steady state for $S_F = 0.3$ [kmol·m^{-3}].

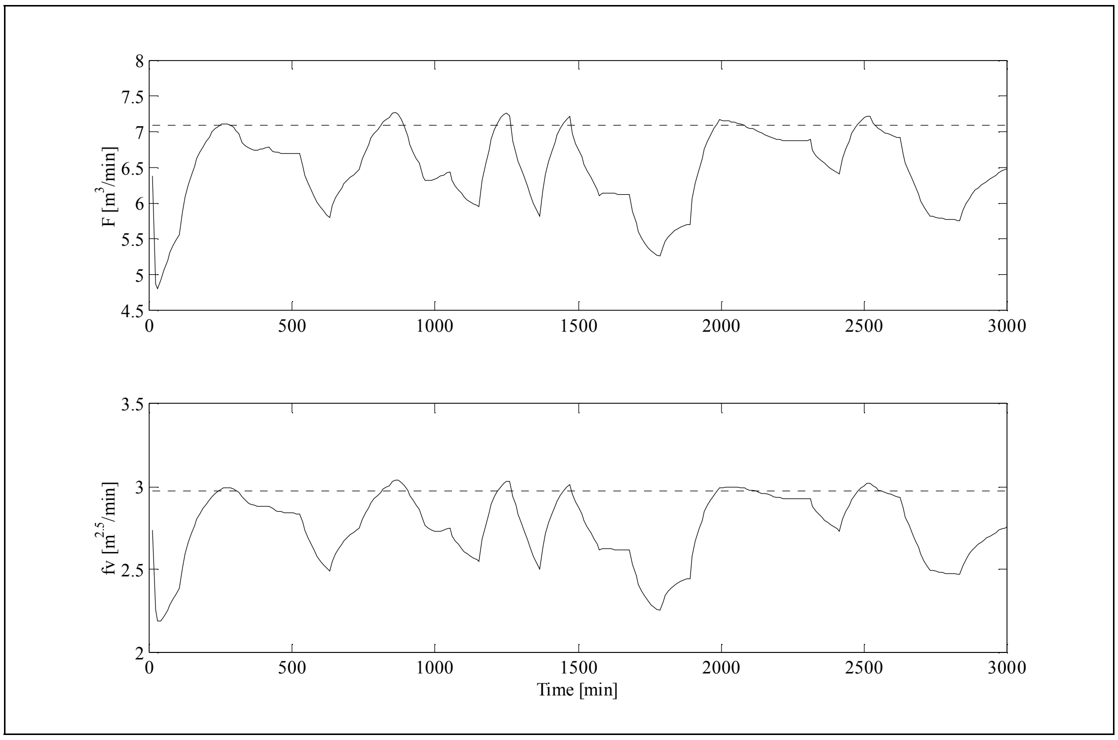


Figure 4-18 - The control action of the manipulated variables to gaussian noise on the S_F. (-----) Economic optimal steady state for S_F = 0.3 [kmol·m^{-3}].

4.7.1 Region change

Other than process disturbances such as S_F, disturbances in market conditions are also inevitable. Should the raw material cost of the concentrated substrate feed, S_C, change from 200 [\$/m^3] to 150 [\$/m^3] owing to prevailing market conditions, the economic optimum shifts dramatically as seen in Figure 4-19 to Figure 4-21. With $c_3 < 150$ [\$/m^3], it becomes profitable to use S_C, where previously the cost was prohibitive. Assuming model validity and sufficient dissolved oxygen can be supplied to sustain the high cell concentration (Figure 4-19), the operating profit almost doubles as seen in Figure 4-21. The two operating regions are located in vastly different regions of the state space. It becomes an inefficient task for a single neurocontroller to map such a large area of the state space to control actions. Typically such a strategy may be implemented using two optimal neurocontrollers that are switched from one to the other based on the cost of S_c. The generalisation and robustness of two such neurocontrollers allow for seamless transitions from one control strategy to the other.

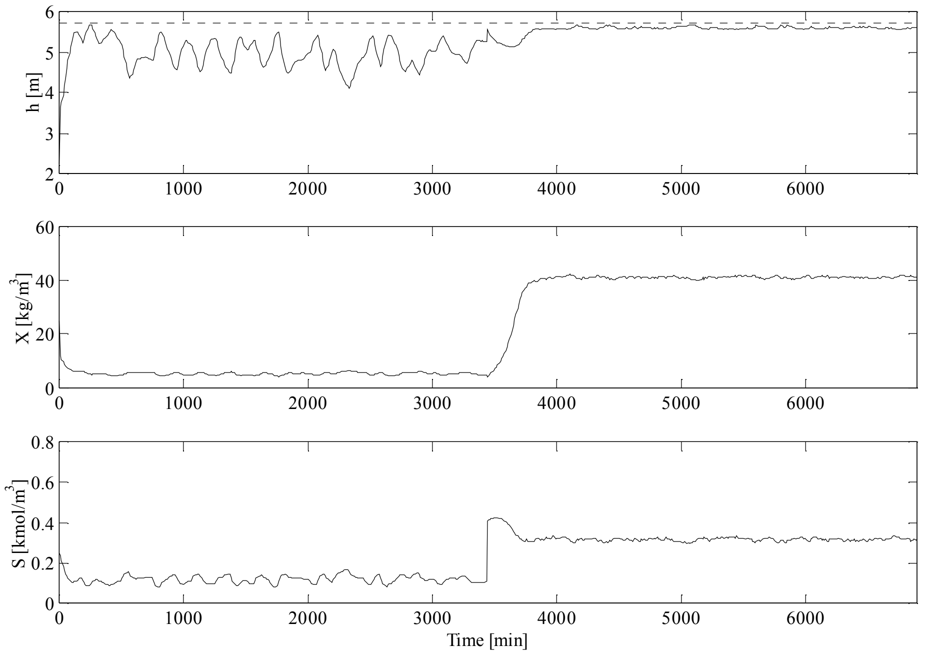


Figure 4-19 - State variable responses for a change in operating region necessitated by a change in the raw material cost of S_C from 200 [\$/m^3] to 150 [\$/m^3].

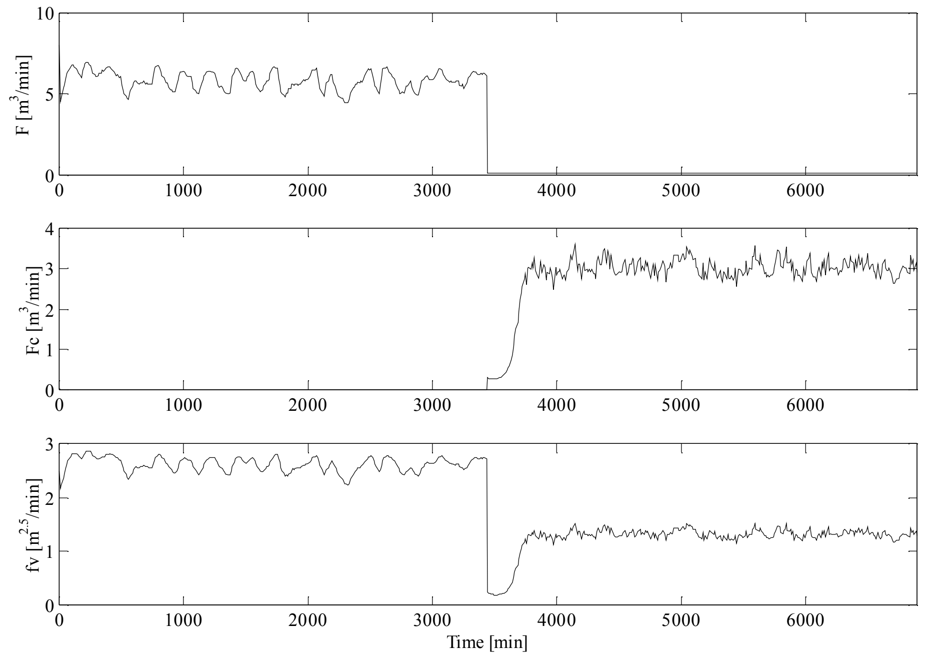


Figure 4-20 - Manipulated variable outputs for a change in operating region necessitated by a change in the raw material cost of S_C from 200 [\$/m^3] to 150 [\$/m^3].

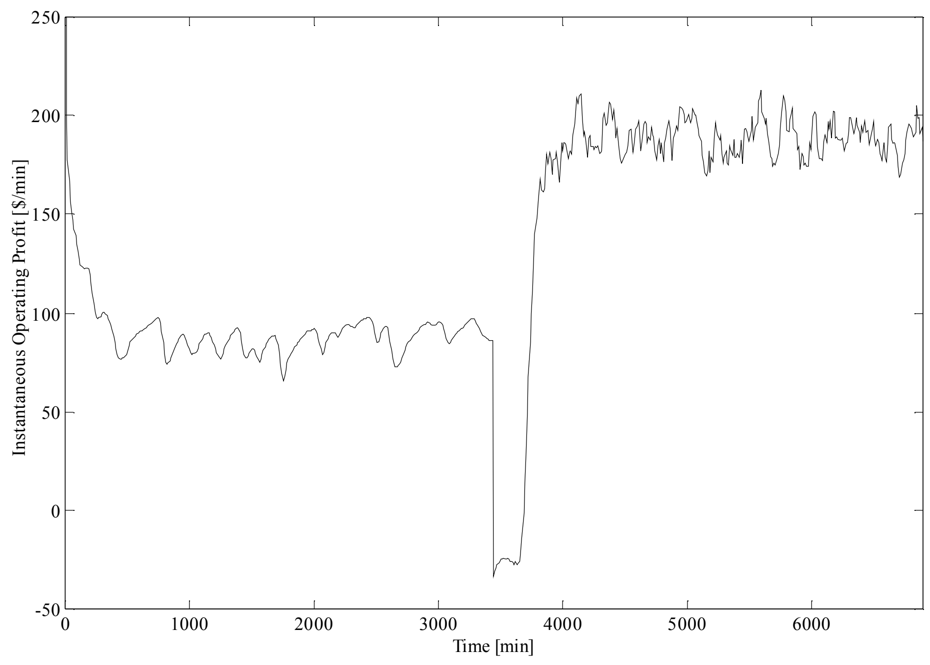


Figure 4-21 - A change in the raw material cost of S_c from 200 [$/m^3$] to 150 [$/m^3$] shift the operating region with the greatest economic return significantly.

4.7.2 Conclusions for SMNE neurocontroller development

Evolutionary reinforcement learning, in particular the SMNE algorithm, offers a robust search alternative for combined design and control optimisation. SMNE allows for flow sheet optimisation and optimal controller development based on economic considerations. The integrated design and controller development functions should significantly reduce the occurrences of over-design in the process industries. SMNE may thus considerably reduce the implementation of conservative design objectives. Also, the high degree of generalisation afforded by the developed non-linear neurocontrollers brings about more autonomous control for highly complex non-linear process environments. Efficient generalisation also allows for integration of several neurocontrollers into a single control strategy, thereby ensuring optimal operation over vast operating shifts in the state space brought about by market changes.

4.8 COMPARATIVE ABLATION BIOREACTOR STUDY

SMNE and SANE are highly effective global optimisation algorithms. Given the bioreactor benchmark as described in section 4.7, SMNE and SANE arrive at the same global optimum. Convergence speed is critical in large, real world problems where a single evaluation, though simulated, may consume significant processing time. As described in section 4.4, the SANE algorithm (section 4.3) has limitations in evolving neural networks. SMNE (section 4.5) overcomes these encoding limitations and augments a symbiotic EA with the synergy of a local search algorithm, PSO. Using the bioreactor benchmark as the dynamic environment, an ablation study demonstrates the contribution that each element in SMNE makes to convergence speed.

SMNE was compared to three other methods: (1) multi-start stochastic hill climbing (SHC) as a reduced model, augmented with an initial random search for suitable starting points, (2) the SANE algorithm as a symbiotic EA with a single population of neurons, and (3) the symbiotic sub-population EA used in SMNE without PSO. The neurocontrollers received the bioreactor's three sensor readings as inputs, and determined the output positions of the four valves. Twenty learning simulations, each for a total of 30 000 evaluations, were completed for each learning method. Each evaluation was initialised to a random process state and run for 300 sample periods. The fitness for each evaluation was calculated as:

$$f = \int_{0}^{300} t \cdot \phi(t) \cdot dt \tag{4-13}$$

where f is the fitness value and $\Phi(t)$ is the instantaneous profit at sample period, t. The statistical significance of the performance differences was measured using ANOVA analyses.

Table 4-7 - ANOVA analysis for ablation study.

ANOVA analysis	*P-Value*
Stochastic hill climbing vs. SANE	$8.9 \cdot 10^{-6}$
SANE vs. SMNE's symbiotic EA	$5.1 \cdot 10^{-4}$
Symbiotic EA vs. SMNE	$2.1 \cdot 10^{-3}$

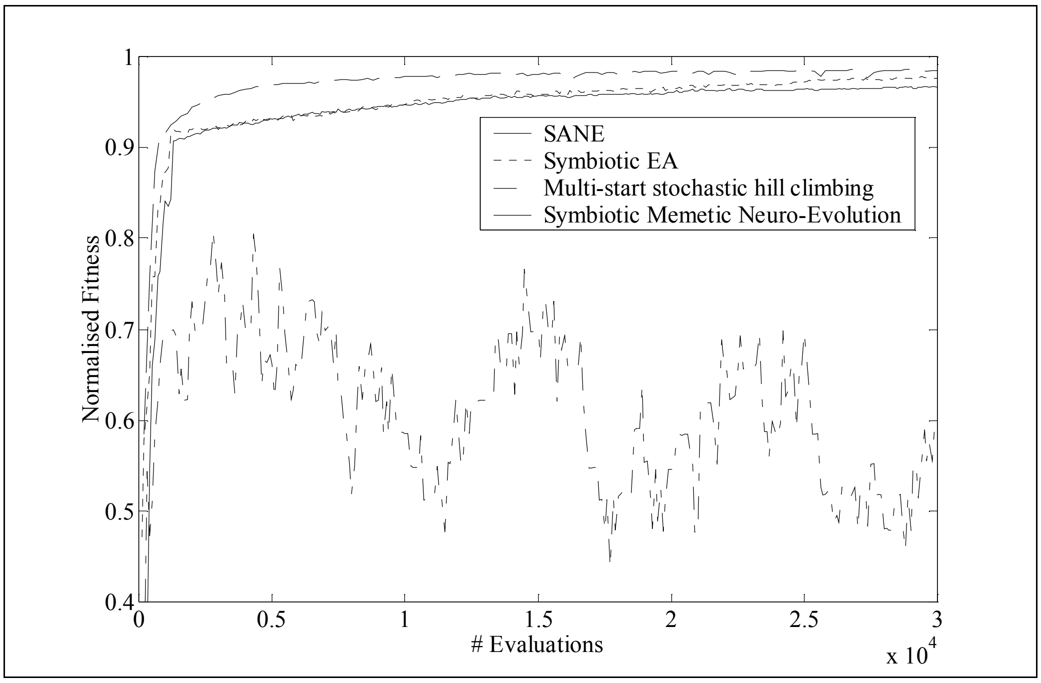


Figure 4-22 - Average normalised venture profit for the learning methodologies. A normalised venture profit of 1.0 represents the global optimum for the bioreactor control task.

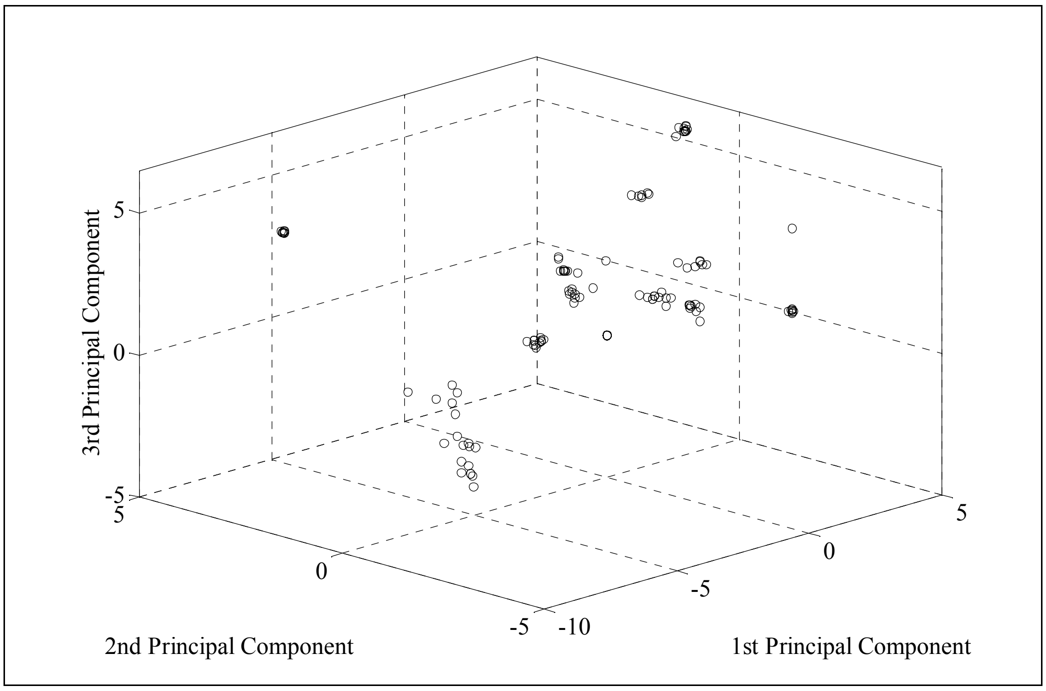


Figure 4-23 - Principal components of the weight vectors of each neuron weights in the elite network population.

The average normalised venture profit for each method is shown in Figure 4-22. The ANOVA results are tabulated in Table 4-7. Stochastic hill climbing could not learn an effective control strategy for the bioreactor (Figure 4-22). SHC could not reliably progress beyond the initial basin of attraction for any of the twenty simulation runs. This demonstrates the complexity of the bioreactor's dynamics and justifies using more complex algorithms in solving the control task.

The EA algorithms proved more successful in progressing towards the global optimum (venture profit = 1). Although SANE and SMNE's symbiotic EA are similar in implementation, SMNE's symbiotic EA deals more effectively with competing conventions and focuses the niching phenomenon. The ANOVA analysis indicates that the sub-population treatment is statistically superior (ANOVA $p < 0.01$) to the single neuron population SANE algorithm.

SMNE, with its local search refinement operator (PSO), both accelerated and produced a higher average venture profit than the symbiotic EA implementations alone (Figure 4-22). The ANOVA analysis (ANOVA $p < 0.01$) also indicates that the PSO treatment in SMNE is statistically superior to the symbiotic EA implementations alone.

Figure 4-23 presents a principal component analysis (first three principal components, explain 77 [%] of the variance). Each marker represents a neuron in the elite network population. The key elements of SMNE are illustrated: (1) the observed clusters illustrate the niching pressure induced by implicit fitness sharing, (2) genetic diversity has been maintained, allowing continued exploratory search, (3) each neuron cluster represents a swarm, which refines the promising sub-space regions identified by the symbiotic EA.

4.8.1 Discussion of ablation study

The ANOVA analyses (Table 4-7) prove that the synergy between evolution (symbiotic EA) and lifetime learning (PSO) in SMNE significantly enhances learning efficiency. This synergy relies on an effective balance between exploratory (genetic search) and exploitative (lifetime learning) search in the solution space. Implicit fitness sharing preserves this synergetic balance by maintaining genetic diversity through induced niching pressure. Genetic diversity allows a continued exploratory search, without which lifetime learning could not exploit the solution space effectively. Niching pressure bounds the solution space into distinct sub-space regions (clusters in Figure 4-23), which are partial solutions to the complete task. These sub-space regions are prime candidates for a local search (learning). The symbiotic EA's niching phenomenon thus aids learning by creating good conditions for lifetime learning (i.e. initial weights). Learning guides evolution by absolving it from exploring neighbouring solutions to the current evolutionary solutions. Evolution only needs to find appropriate regions in the solution space, rather than specific points in the solution space. A synergetic effect thus motivates the learning efficiency in the SMNE algorithm.

In the SMNE algorithm, evolution is learning at the level of the neuron population, while lifetime learning (PSO) is learning at the level of each individual neuron. The evolutionary task searches for cooperative neurons, while the learning task seeks to improve each neuron's partial solution to the complete task. The evolutionary and learning tasks are thus quite different. What the neurons are required to learn (i.e., the learning task) and which neurons are selected during evolution (i.e., evolutionary task), are indirectly related. In other words, the symbiotic EA selects neurons that cooperate successfully, while PSO optimises the partial solutions that each neuron represents. The evolutionary fitness surface and the learning fitness surface are correlated, i.e. superior neurons tend to have high fitness values on both the fitness surfaces (Nolfi et al., 1994). A superior neuron cooperates effectively and also represents a good partial solution to the control task. Effective synergy results once high correlations between the learning and the evolutionary fitness surfaces are found.

However, the fitness landscapes of the learning and evolutionary tasks are continuously changing, relative to one another, during evolution. This continuous change depends on the population's current location in the solution space. This suggests a dynamic correlation between the two fitness surfaces (Nolfi et al., 1994). Consider a novel neuron, with high learning task (i.e., partial solution) fitness. High partial solution fitness improves the likelihood of selection for genetic reproduction. Over several generations, the neuron is thus likely to obtain additional pointers from the network population. A greater number of neuron pointers translate to a higher cooperation (i.e., evolutionary) fitness. For effective synergy, a search for high dynamic correlation between the fitness surfaces must be maintained.

4.8.2 Conclusion

The Symbiotic Memetic Neuro-Evolution (SMNE) algorithm is effective at developing neurocontrollers for use in highly non-linear process environments. Implicit fitness sharing maintains genetic diversity. Implicit fitness sharing's niching pressure accelerates the evolutionary search for solution sub-spaces that may be exploited by local search. Particle swarm optimisation effectively absolves the EA from devoting its resources to local refinements. The synergy between the symbiotic EA and PSO accelerates learning from dynamic process models. SMNE's efficient learning translates to greater economic return for the process industries.

4.9 ADAPTIVE NEURAL NETWORKS

A process' operating point (i.e., the process state) determines the product purity and production rate. As in section 4.7, the operating point thus has an intrinsic economic value. Control engineers select fixed operating points (i.e., set points) based on economic value. Process changes, due to process disturbances and drifting dynamics, cause deviations from the set points, requiring corrective action. Optimal set points and effective corrective actions yield greater economic return. Typically, linear controllers (e.g., PID controllers) maintain the set points and provide corrective action to process changes (Seborg et al., 1989). For example, an exothermic reactor has an optimal operating temperature. This temperature determines the production rate that directly impacts on the economic return from the reactor. The control engineer selects this optimal temperature as a set point. A PID controller responds to process disturbances that affect the reactor temperature, by increasing or decreasing the cooling water flow rate, thereby maintaining the set point.

PID controllers typically utilise both fixed set points and fixed controller parameters. The PID controller parameters govern the corrective action (i.e., the control response) to process changes. There are three PID controller tuning parameters: proportional gain, integral and derivative. PID control's linear control structure is the industry standard, though not suited to non-linear processes that are common in the process industries. For non-linear processes, PID controller parameters are optimal only over a limited operating region. Process changes may cause the operating point to stray far from the set point, whereupon PID controllers may implement sub-optimal corrective actions. Detuning of the PID parameters may be necessary to maintain robust performance and ensure stability, resulting in a sacrifice of controller performance (Kavchak & Budman, 1999). Sub-optimal performance may be avoided only by adapting the controller parameters. As the set points largely determine the economic return, the set points must also adapt in response to process changes. Tracking the economic optimum therefore requires adapting both the controller parameters and the set points (Hrycej, 1997).

Even a basic control system comprised of a linear, time-invariant process and a linear feedback control law, becomes a high-dimensional, coupled, non-linear problem with the addition of on-line parameter tuning (Sanner & Slotine, 1992). For non-linear systems, a constant proportional gain for a PID controller may give unsatisfactory control responses and even result in instability. Even for linear processes, the system dynamics change over time due to, for example, catalyst decay and mechanical wear-and-tear. Different operating conditions should thus be matched to a set of PID controller gains (i.e., gain scheduling). Neural networks have been used in hybrid adaptive schemes. Megan & Cooper (1992) and Parlos et al. (2001) presented methodologies for using neural networks to map the current process conditions to a particular set of linear controller parameters. Essentially, these methods involve self-tuning PID controllers using neural networks. Megan & Cooper (1992) used a neural network, trained on patterns from ideal second order response data, to update PID tuning parameters. Similarly, Kumar and Guez (1991) used a neural network to estimate the second order model parameters based on the system response. Given the approximation of the plant within a linear second order model, standard pole placement techniques were used to calculate the controller parameters. This method has the advantage of being applicable to a wide range of dynamic systems without needing an explicit dynamic model.

Section 4.7 demonstrates that for complex process dynamics, sub-optimal economic performance may result even for minor disturbances to the process (Figure 4-16). A feedforward neurocontroller (i.e., static neural network) with constant weights presumes that both the process dynamics and the control objective are static over time. For real-world systems, this is almost never the case and although static neural networks generalise well to new process conditions, on-line adaptation of the weights preserves near-optimal control performance.

Numerous adaptive schemes have been proposed for neural networks in on-line process applications. The uncertain and changing nature of non-linear, time-varying processes necessitates on-line adaptation, thereby ensuring operation at the economic optimum. On-line adaptation of neural networks may prove problematic due to a lack of first principles knowledge, slow learning and rapid "forgetting" of past process knowledge. Slow learning is a function of the training algorithm. The backpropagation algorithm needs numerous passes through the training data to reduce the model error, which may be too slow for on-line application. More importantly, seldom-seen patterns are forgotten during adaptation as nodes are re-allocated to map more recent changes in the process dynamics. Typically, neural networks model the entire process, which presents a complex mapping (i.e., large neural networks) that is not suited to the time constraints and robustness criteria associated with on-line adaptation. Including neural networks within a fundamental model could reduce the learning burden (Ungar et al., 1990). Convergence and stability measures have been limited to specific classes of non-linear systems and are not generally applicable to numerous real-world processes. Ungar et al. (1990) concluded that based on the above arguments neural networks should not be adapted on-line. Rather, neural networks should be updated off-line with new process information and frequently updated.

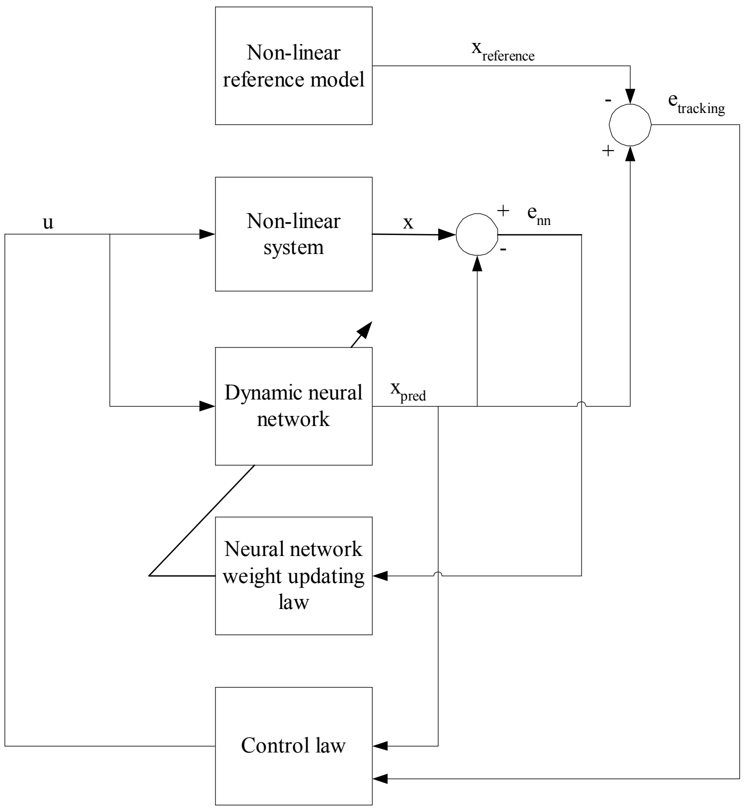


Figure 4-24 - Indirect control using adaptive neural networks (Poznyak et al., 1999).

Saerens and Soquet (1991) differentiated between generalised and specialised learning for neural network controllers. Generalised learning refers to learning from input-output plant models, where the learning task drives a dynamic system so that its state vector follows a desired trajectory. Effectively, this involves inverting the process dynamics and the controller is required to function as an optimal controller based on a chosen cost function (i.e., control objective). Specialised learning entails learning from the direct evaluation of the actual plant output as compared to the desired plant output. The network weights are changed based on the error signal between the actual and the desired value of the controlled variable. Dirion et al. (1996) noted that the learning rate for specialised learning based on back-propagation must be small, otherwise the control response becomes oscillatory and the neurocontroller becomes ineffective. The rate of specialised learning is thus not only limited by the type of learning algorithm, but may require slowing the chosen algorithm to prevent large changes to the controller structure. In turn, specialised learning may be categorised as either indirect or direct control. Poznyak et al. (1999) proposed an indirect control methodology as shown in Figure 4-24 in which the parameters of the neural network plant model are adapted on-line and the control law (e.g., model predictive control) is updated based on the updated dynamic model. For indirect adaptive control, any improvement in the estimation of plant dynamics should result in more accurate models and consequently for model-based control in better controller performance (Kavchek & Budman, 1999). Saerens and Soquet (1991) proposed a direct control methodology by directly adapting the weights of a neural network controller to reduce the output error. Unfortunately, the scheme by Saerens and Soquet (1991) is demonstrated using toy domains. Toy domains may relax conditions critical to efficient on-line adaptation in complex, real world domains.

On-line adaptive control (i.e., specialised learning) suffers from a need to maintain the persistent excitation (PE) condition. PE can be described as an observability criterion, which ensures that the state variables are observable (i.e., may be reconstructed) in the time series of the process variable (Ydstie, 1990). Therefore, for a model to be updated consistently and accurately, the state variables must be embedded in the input data time series. Interestingly, steady state off-set cannot provide the necessary dynamic information that adheres to persistent excitation, though in the indirect adaptive approach the model must be updated to eliminate off-set. The adaptive algorithm thus not only determines when the actual control response deteriorates far enough from the desired response for adaptation, but also needs to assess whether the current process signals contains information of the dominant state variables for adaptation. Direct adaptation may suffer less from this constraint, since there is no model integrity that must be maintained.

Krishnapura and Jutan (2000) noted that neural network models are typically not parsimonious and therefore an adaptive control scheme needs to update a large number of weights. Updating a large number of weights at each sampling instance is prone to error. Krishnapura and Jutan (2000) proposed using a small neural network structure (i.e., two neurons) modelled on the conventional PID controller, making on-line adaptation by simple gradient methods viable. This adaptive approach was demonstrated in noise-free simulation studies, preventing assessment of this gradient-based method in noisy process environments. Narenda and Balakrishnan (1997) proposed using switching rules to cater for changing process conditions. Multiple models, with few weights, map different operating regions in which the control system may need to function. If the plant is sufficiently close to any particular model, the model may be used to choose the controller. Multiple models are used to determine both when and to which controller the control system should switch, as dictated by the process conditions. Further, avoiding a large number of weights for adaptation, Pottmann & Henson (1997) demonstrated that radial basis function (RBF) networks have favourable adaptation properties. The localised mapping of RBF networks imply that only active nodes need to be adapted, since new information only applies to the firing neurons. This reduces the number of weights that require adaptation considerably.

In any adaptive control methodology, the task reduces to finding a stable algorithm for adjusting the weights of the network. A brute-force correction of the controller parameters based on the gradient of the output error, particularly for noisy input signals, could result in instability even for linear systems. To avoid such instabilities, neurocontrollers are often trained off-line to first provide, as a minimum, stabilising non-linear control. A simple approach is to find the identity mapping that approximates the inverse dynamics of the non-linear system (see 4.1.1). Direct inverse control may result in large control actions unsuited to many dynamic systems. The off-line training is thus frequently posed as an optimisation problem using a reference model or cost function (see 4.1.1). Stabilising or near-optimal performance may be further improved by on-line adaptation of the neural network weights in response to process changes. Effective generalisation and adaptability during process changes are essential to tracking a process' economic optimum. Robust search techniques are required for effective on-line adaptation of neurocontroller weights.

Section 4.10 outlines basic notions in conventional adaptive control, which remain relevant to an advanced adaptive control schemes. Section 4.11 introduces an adaptive neurocontrol strategy, Adaptive Neural Swarming (ANS). A highly non-linear bioreactor benchmark (section 4.7) is used in the control simulation. The bioreactor's dynamic behaviour is changed continuously, which shifts the operating point with maximum economic return. In section 4.12, ANS adapts an existing neurocontroller's weights to reap greater economic return from the changing bioreactor process. ANS emerges as an effective tool for adapting existing neural network strategies, resulting in enhanced performance.

4.10 ADAPTIVE CONTROL METHODS

Control design requires a dynamic process model. Optimal control design is possible only if the process model is accurate. However, the model and the actual process are invariably mismatched. Also, exact knowledge of possible process changes is seldom available for control design purposes. Despite these shortcomings, robust control remains a control requirement. Generalisation is the ability of a controller to deliver near optimal performance, despite limited process knowledge during its design.

Generalisation may provide robust control, but optimal control is rarely ensured during the control design process. The designed controller frequently requires on-line refinements to the controller parameters and set points. Improved generalisation is difficult to impart on-line, as it involves reconciling past (i.e., design) and current process information into a single control strategy. For example, catalyst decay may cause the optimal temperature of a reactor to change over time. In contrast, adaptation changes controller parameters giving precedence to on-line process information. However, degraded performance may result should past process conditions return. A balance must thus be maintained between retaining generalisation imparted during design, while allowing adaptation to exploit changes in the process conditions (Hrycej, 1997).

On-line process information contains inaccuracies due to sensor noise and short-lived disturbances. Adapting controller parameters based on imperfect process information involves operational risk. The process may become unstable. On-line adaptation to control parameters faces numerous challenges: (1) Balancing the use of past and present process information, (2) supervising process stability, (3) implementing emergency procedures should the process become unsafe, due to on-line adaptation (Hrycej, 1997).

The following two sub-sections illustrate the aims of conventional methods for adapting controller parameters (section 4.10.1) and process set points (section 4.10.2). ANS has the same aims, though its methodology is dissimilar.

4.10.1 Conventional adaptive control

An adaptive linear controller maintains a specified control response (i.e., corrective action) around a set point during process changes. For non-linear processes, a set of PID controller parameters can only maintain the specified control response for a limited range of process conditions. Process changes in non-linear processes may cause the control response to become oscillatory around the set point, as illustrated in Figure 4-25a. Adaptive linear control tunes the PID controller parameters, which corrects the oscillatory response in Figure 4-25a to the specified response in Figure 4-25b. Conventional adaptive control relies on on-line process modelling (i.e., Model Reference Adaptive Control) and heuristic methods (i.e., Ziegler-Nichols) for adapting controller parameters (Ghanadan, 1990). Likewise, ANS must also ensure that a specified control response is maintained.

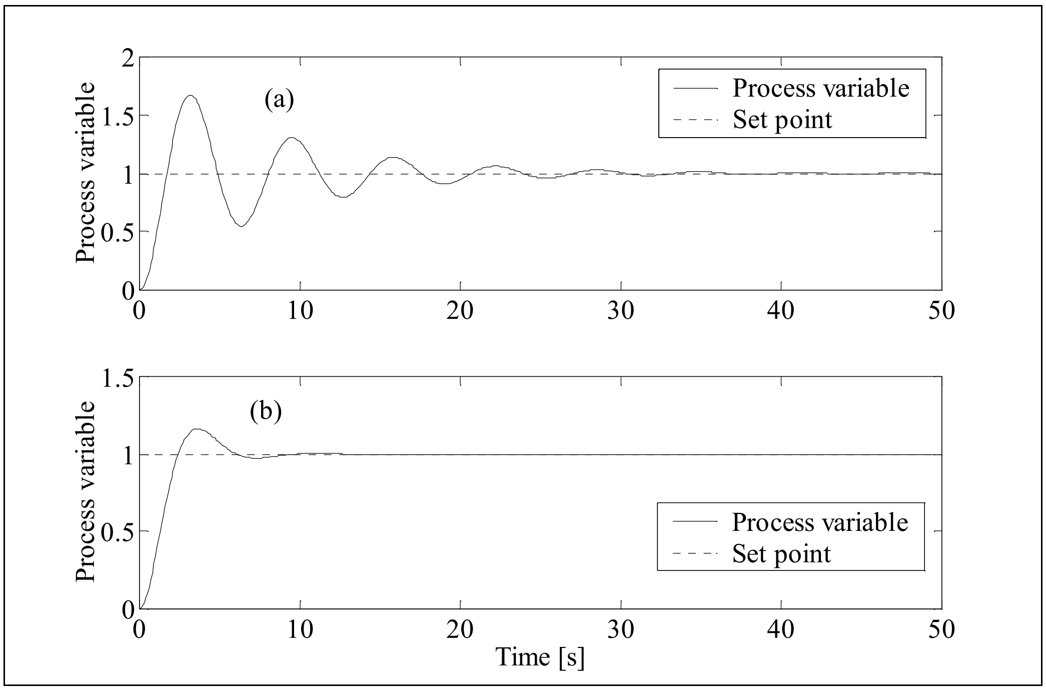


Figure 4-25 - Objective of linear adaptive control. An oscillatory control response around the set point (a) is changed to a specified control response (b). The specified control response settles sooner on the set point.

4.10.2 Evolutionary Operation

Adaptive control does not change the set points that largely determine the economic return. Set points are selected during design based on an optimisation of the dynamic model equations. The optimisation considers both economic return and controllability. However, process changes during operation may make the current set points economically sub-optimal.

Evolutionary operation (EVOP) challenges the use of constant set points in a continuously changing process. EVOP monitors the process and improves operation by changing the set points towards the economic optimum. EVOP makes a number of small set point changes that do not disrupt production. However, the set point changes need to be sufficiently large to discover potential improvements in the operating point. EVOP uses an experimental design to determine the number of set point change experiments. Pattern search methods use the experimental results to determine whether and in which direction the set points should be changed (Walters, 1991).

Consider Figure 4-26, which graphs the economic return of a process that has two process variables. The contour lines represent operating points with similar economic returns. The circular marker represents the current set point, which is economically sub-optimal. The set points for both process variables should be reduced for optimal economic return. EVOP conducts a number of set point change experiments (represented by square markers) in the neighbourhood of the current set point. The economic return for each set point experiment is determined. In Figure 4-26, three experiments have greater economic return than the current set point. EVOP adjusts the current set point in the direction of greater economic return. The process is repeated until optimal set points are found (Walters, 1991).

EVOP does not adapt the PID controller parameters for each of the set point experiments. As discussed in section 4.10.1, using the same controller parameters for all the set point experiments may give oscillatory responses. Poor control responses impact negatively the accurate determination of economic returns.

Adaptive control and EVOP may be combined in a two-step methodology to track a changing economic optimum. EVOP selects a number of set point experiments. An adaptive control method establishes a specified control response for each set point experiment. The economic evaluations for each experiment will consequently be comparable, whereupon EVOP adjusts the current set point. This cumbersome two-step process is repeated until the optimal set point is found. Ideally, a single on-line experiment (evaluation) should provide information on both the economic return and the control response.

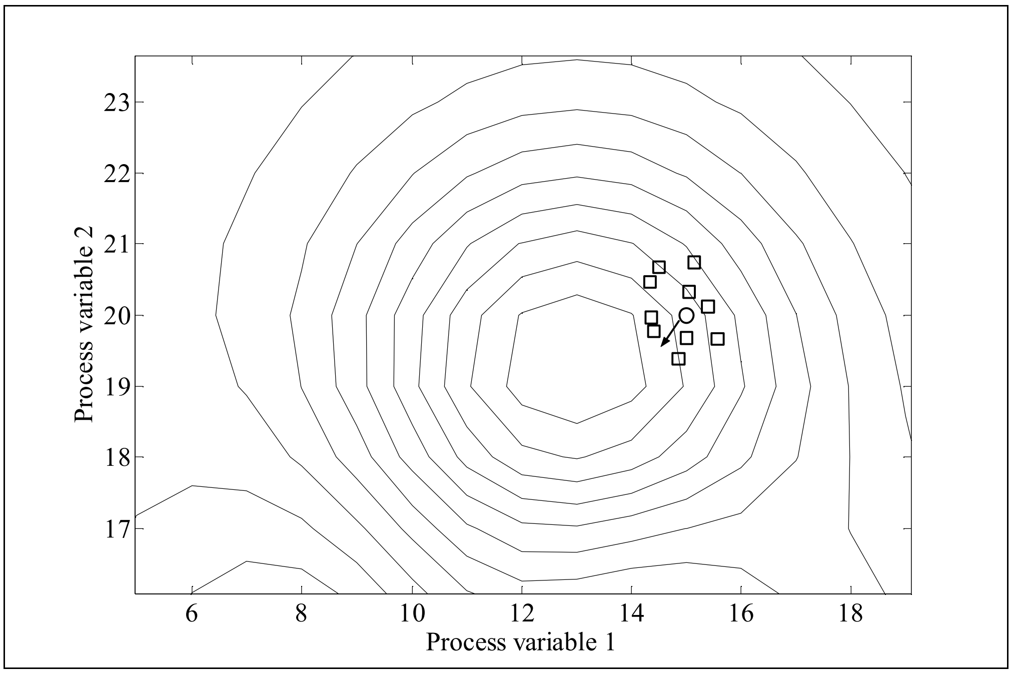


Figure 4-26 - EVOP for a process with two process variables. The current set point (circular marker) is moved along the arrow's trajectory based on the economic return of each set point experiment (square markers). The process operation is thus improved.

4.11 ADAPTIVE NEURAL SWARMING

This section describes Adaptive Neural Swarming (ANS), which combines adaptive control and EVOP into a single comprehensive step. In ANS, both the economic return and the control response are combined into a single feedback signal. A local PSO uses this sparse reinforcement information to adapt the weights of existing neural network controllers towards greater economic return in response to a changing process.

4.11.1 Neural network structures

Neurocontrollers may originate from various sources. Neural networks may be trained to mimic the control actions of existing PID controllers, thereby distributing the PID functionality over several neurons. Existing fuzzy logic systems may be converted to equivalent neural network architectures (Jong & Sun, 1993). Neurocontrollers are also developed utilising evolutionary reinforcement learning techniques (Conradie et al., 2000). Neural networks possess characteristics that are beneficial to an adaptive scheme, such as generalisation and graceful degradation.

For non-linear systems, Chen and Khalik (1995) confirmed the importance of extensive off-line training based on a prototype (even partial) model of the dynamic system. Off-line training provides a good starting point when the network is adapted on-line. Although the neural network may only provide robust or stabilising control, such a network may have initial parameters that are within the domain of attraction.

The ability of neural networks to generalise from insufficient or partial information is particularly useful for noisy input data and provides a computing architecture for real-world adaptive control. Also, once a PID controller is adapted, the small number of control parameters prohibits effective generalisation to past process conditions. Neural network controllers are collections of neurons, with each neuron specifying the weights from the input layer (process states) to output layer (control actions). Neurocontroller parameters are the neural network weights. A neurocontroller that is equivalent to a PID controller, has additional degrees of freedom, owing to a larger number of controller parameters. During adaptation, a neural network's distributed functionality preserves greater generalisation to past process conditions. The need for effective generalisation justifies the use of neural networks.

Neural networks also exhibit graceful degradation. Graceful degradation allows small changes to the weights, without causing catastrophic control performance loss (S'euim & Clay, 1990; Ydstie, 1990). Process stability is preserved during adaptation.

These neural network characteristics are relied upon in a reinforcement learning framework, described below, to provide process stability and continued generalisation.

4.11.2 Reinforcement learning

Reinforcement learning (RL) automates the acquisition of on-line performance (i.e., feedback) information and the adaptation process. RL uses on-line performance evaluations to guide adaptation (section 3.2).

ANS maintains a population of possible neurocontroller solutions that serve as RL evaluations, similar to EVOP experiments. Each neurocontroller is evaluated individually over a number of sensor sample periods while interacting with a dynamic process as in Figure 3-1. Initially, the process may be at an arbitrary operating point (state, s_t). The neurocontroller observes the current process operating point at sample, t, and selects a control action, a_t. The control action changes the operating point to s_{t+1}. A reward, r_t, is assigned based on the economic value of this new operating point. The objective is to maximise the total reward over a series of control actions, while maintaining a specified control response. An optimisation algorithm adapts the neural network weights based on the reward feedback from each evaluation.

ANS treats the population of neurocontrollers as a swarm, using a local particle swarm optimisation for adapting the weights of each neurocontroller.

4.11.3 Particle swarm optimisation

PSO is loosely based on the social behaviour of flocks of birds. A population of individuals is updated based on feedback evaluations, gathered from the collective experience of the swarm individuals (Shi & Eberhart, 1999). Equations 4-14 and 4-15 determine the velocity and position of the swarm in the solution space:

$$v_{id} := \omega \cdot v_{id} + c_1 \cdot rand() \cdot (p_{id} - x_{id}) + c_2 \cdot rand() \cdot (p_{gd} - x_{id}) \qquad \textbf{(4-14)}$$

$$x_{id} := x_{id} + v_{id} \qquad \textbf{(4-15)}$$

where each particle, i, moves through the solution space with dimension, d. Each particles velocity vector, v_{id}, is dynamically adjusted according to the particle's own best experience, p_{id}, and that of the current best particle, p_{gd}, in the swarm. These two knowledge components are blended with each particle's current velocity vector to determine the next position of the particle as per equation 4-15 (Shi and Eberhart, 1999).

The best swarm particle is a beacon to a region of the solution space that may contain better optimisation solutions. Each particle searches the solution space along its unique trajectory for better solutions. Should a better solution be found, the new best swarm particle moves the swarm in a new direction. The momentum in each particle's current velocity provides some protection against convergence to a local optimum (Shi and Eberhart, 1999).

PSO has been utilised in tracking changing optima in function optimisation problems (Carlisle and Dozier, 2001; Angeline, 1997). PSO's success in these artificial domains motivates its use in complex real-world problems.

4.11.4 On-line optimisation using ANS

ANS uses a local PSO search as the optimisation algorithm within a reinforcement learning framework. ANS thereby tracks the shifting economic optimum resulting from a changing process. Practical considerations for on-line application relate to the (1) selection of swarm size, (2) swarm initialisation, (3) appropriate PSO parameters and (4) duration of an RL evaluation.

Each on-line experiment is time and resource intensive, since no control improvements are possible during the evaluation phase. The number of reinforcement learning evaluations per PSO adaptation must therefore be minimal. However, the dimensionality of the control task constrains the minimum number of evaluations. More process information (i.e., more evaluations) is required during the evaluation phase, as the dimensionality of the control task increases. Otherwise, effective adaptation based on on-line feedback is not possible. Each neuron in a neurocontroller represents a partial solution to the control task. The number of neuron weights reflects the dimensionality of such partial solutions. For example, to effectively adapt neurons with 12 weights, an absolute minimum of 12 evaluations is required. The number of swarm neurocontrollers *(n)* is thereby selected based on the dimensionality of the control task, as reflected by the number of neuron weights.

In ANS, each swarm particle is an altered version of an existing neurocontroller. The initial swarm consists of the original (i.e., existing) neurocontroller and *(n-1)* altered neurocontrollers. Each altered neurocontroller is initialised with a small gaussian deviation from the existing neurocontroller weights. The maximum weight deviation is 3 [%] from the each original weight, thereby altering the control policy only marginally. A neurocontroller swarm is thus initialised in a local region of the network weight solution space. This slight weight alteration determines the direction in which the swarm should move, without negatively effecting production and inducing process instability. On-line evaluation (experimentation) is thus limited to neighbouring solutions of an existing solution.

Each swarm neurocontroller is evaluated on-line for a limited number of sensor samples. A process' time constant is defined as the process response time to a step change in a control action. The process' time constant determines the number of sensor samples used in each evaluation. Equation 4-16 is the fitness evaluation that serves as feedback of each swarm neurocontroller's economic return:

$$Fitness = \int_{t_1}^{t_2} t \cdot \phi(t) \cdot dt - Penalty \tag{4-16}$$

where the evaluation is conducted for the number of samples between t_1 and t_2 and $\Phi(t)$ is the instantaneous profit at time t.

A higher $\Phi(t)$ for each sample reflects a higher economic return, which increases the fitness value. ANS thus searches for improved economic return. Equation 4-16 also dictates the specified control response. An ITAE (integral-time-absolute-error) control response has minimal oscillation, which is suited to numerous process control applications. Maximising the integral results in an ITAE control response. The fitness evaluation thus contains information regarding both the economic return and the desired control response. Also, should hard operating constraints exist for the process, a penalty is assigned should such operating constraints be approached during adaptation. This penalty reduces the fitness and solutions are therefore pursued only within the search boundaries.

An exploitative search preserves generalisation and reduces the risk of inducing process instability. A local (i.e., exploitative) PSO search is implemented by selecting a small inertia weight (ω = 0.4) and the parameters c_1 and c_2 equal to 0.5 (conventionally 2.0) in equation 4-14. Each neurocontroller, i, adapts each weight, x_{id}, at position d in accordance with equation 4-15.

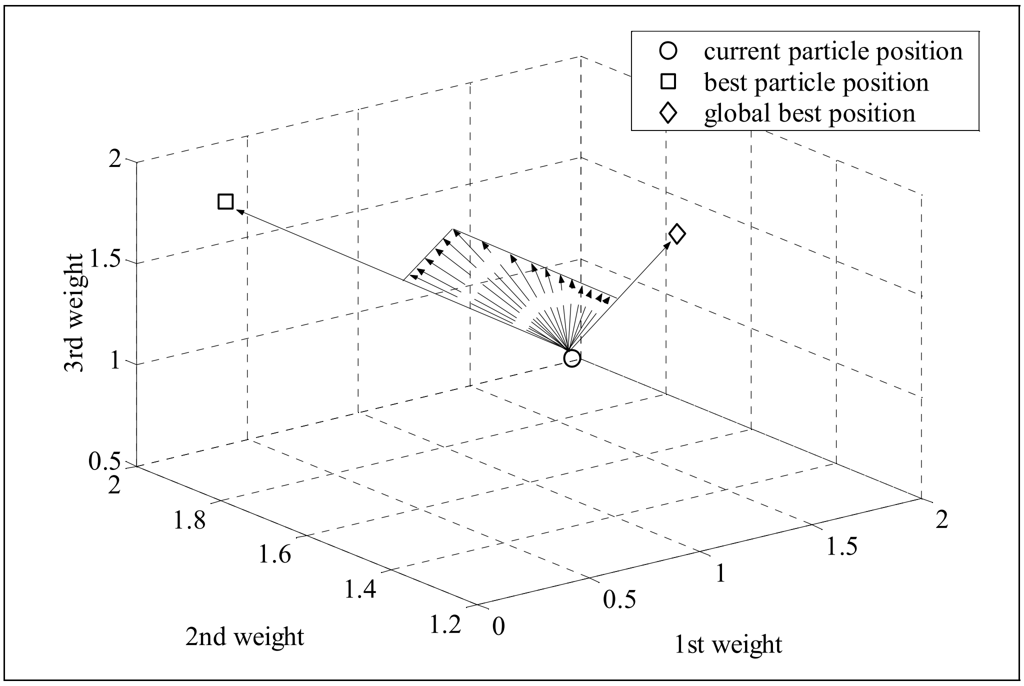


Figure 4-27 - Possible adaptation trajectories of a weight vector based on the swarm's experience. The possible final position after adaptation lies in the plane formed by the arrow lines. The search is exploitative.

A neurocontroller may move only in a limited number of trajectories based on the swarm's experience. Consider a neurocontroller comprised of one neuron with 3 weights with no initial velocity. In Figure 4-27, the circular marker represents the current weight vector. The dashed arrow lines illustrate the possible adaptation trajectories. These trajectories are determined by the global best neurocontroller (square marker) and the neurocontroller's own best experience (diamond marker). These limited trajectories make the search exploitative and are relevant to the optimisation objectives, since the directions are determined by the swarm's collective experience (Shi and Eberhart, 1999).

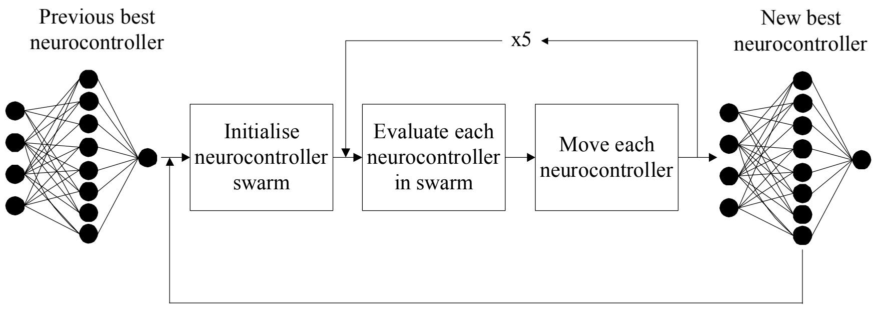

Previous best neurocontroller
x5
New best neurocontroller
Initialise neurocontroller swarm
Evaluate each neurocontroller in swarm
Move each neurocontroller

Figure 4-28 - Adaptive neural swarming flow diagram. An effective neurocontroller is initialised into a swarm and adapted based on the evaluation of the swarm.

The local PSO search is run for five iterations, as illustrated in Figure 4-28. The swarm is then re-initialised around the new best neurocontroller. Re-initialisation starts a new search in the neighbouring solution space of the new best neurocontroller. The search thus continues outside the solution space of the prior initialisations.

ANS was tested in a real-world bioreactor case study. Section 4.12 illustrates ANS' ability to adapt the neurocontroller weights towards greater economic return.

4.12 ADAPTIVE CONTROL OF THE BIOREACTOR BENCHMARK

As described in section 4.7, a bioreactor is a continuous stirred tank fermenter. It contains a biomass of micro-organisms that grow by consuming a nutrient substrate. The liquid substrate is fed continuously into the reactor, which also determines the reactor's liquid level (i.e., hold-up). The biomass is sold as the product. The bioreactor's dynamic behaviour is highly complex, non-linear and varies unpredictably. Also, the bioreactor process is difficult to quantify, due to unreliable biosensors and long sampling periods (Brengel and Seider, 1992).
Furthermore, the maximum bioreactor liquid level is a hard operating constraint. Should operation exceed the maximum level, the bioreactor is shut down and must then be restarted at great operational cost. However, the maximum instantaneous profit increases as operation approaches the hard level constraint. A trade-off between safety and the maximum economic return is required (Brengel and Seider, 1992).

The operating objective is to maximise the venture profit of the process on-line in response to process changes. This entails tracking the operating point with the maximum venture profit and ensuring acceptable control responses. The bioreactor may be simulated accurately and as such constitutes a benchmark for testing new adaptive methodologies without risking unsafe operation.

4.12.1 Experimental set-up

Typical process changes were simulated to mimic real-world bioreactor operation. The bioreactor's model was changed significantly by reducing the cell mass growth K (Figure 4-29a) and increasing the substrate feed concentration S_F. The increased (i.e., off-set) S_F is also disturbed with a gaussian distribution (Figure 4-29b). In addition, the biosensors were inaccurate with a gaussian distribution around the correct reading.

Process search limits ensured that the process operation did not exceed the operation constraints. An adaptation scheme should never induce process shutdown by searching for operating points that are unsafe. The reactor level must remain below a high level alarm, which is a safety margin before bioreactor shutdown is initiated.

Contrary to equation 4-12, the high level alarm was set at 5.95 [m] and bioreactor shutdown at 6.2 [m].

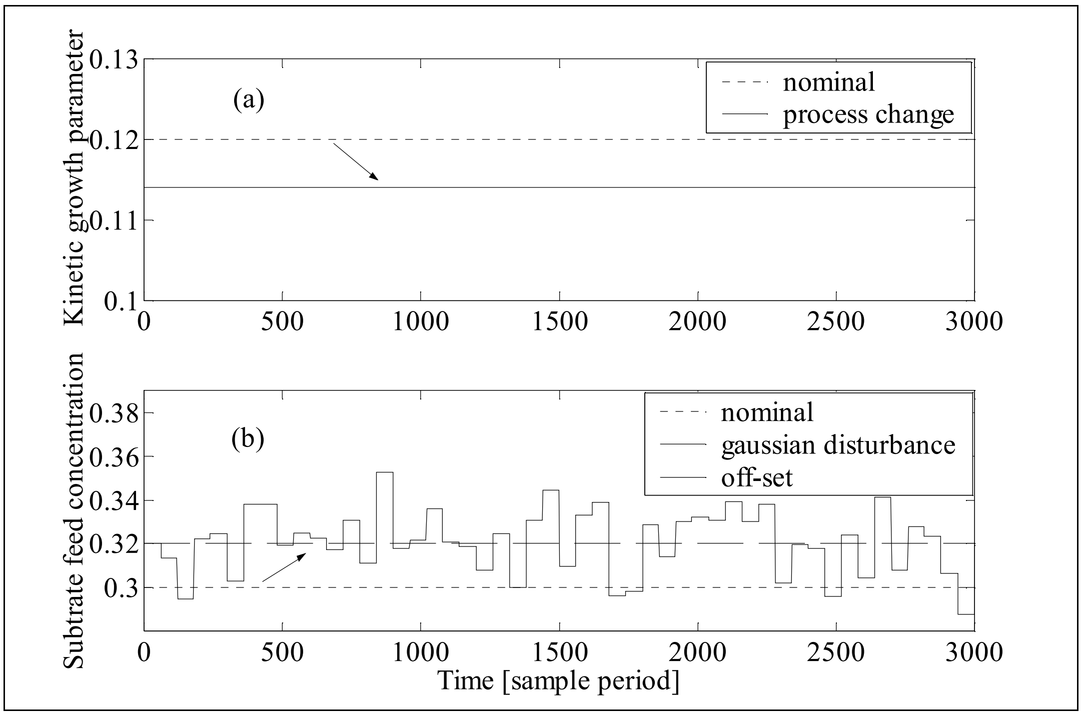


Figure 4-29 - Process changes to the bioreactor. The arrows show the changes from the nominal process conditions (dashed line) for the growth parameter (a) and the substrate feed (b). The process is changed significantly.

An optimal neurocontroller, with 12 neurons comprised of 7 weights each, was developed for the nominal process conditions using the SMNE algorithm (section 4.3). As discussed in section 4.11, ANS utilised this original neurocontroller to initialise a swarm of 10 neurocontrollers and each swarm neurocontroller was evaluated on-line over 20 sample periods. The inaccurate sensors and randomly changing process conditions make obtaining accurate feedback (i.e., evaluations) for ANS difficult. The ten evaluations, though not based on precise information, determined the direction and velocity of the neurocontroller swarm.

4.12.2 Results

4.12.2.1 Adaptation efficiency of ANS

Figure 4-30 presents the instantaneous profit (IP) for the original neurocontroller and the ANS neurocontroller over a hundred day operating period. Figure 4-30 illustrates the effect of the process changes on the IP. The average instantaneous profit for the original neurocontroller was 55 [$/min]. As shown in Table 4-8, this is well below the optimal profit of 96 [$/min] expected during design for the nominal process conditions. The original neurocontroller's IP is reduced due to sub-optimal generalisation to the process changes, though it was able to keep the process stable.

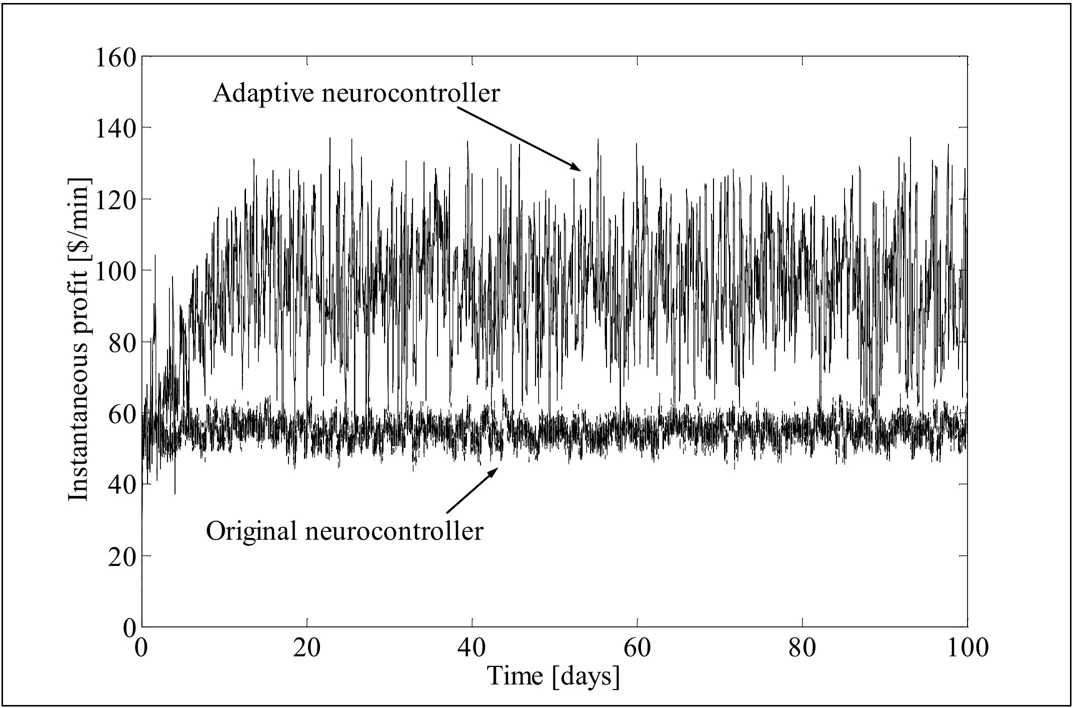


Figure 4-30 - Instantaneous profit for the original and ANS neurocontrollers over 100 days of on-line operation. The adaptive neurocontroller garners greater economic return from the changing process than the original neurocontroller.

Table 4-8 - Maximum IP for changing process conditions.

Process condition	*Maximum profit[$/min]*
Nominal process conditions	96
K reduced, Off-set S_F	106
K reduced, Minimum S_F deviation	69
K reduced, Maximum S_F deviation	130

The original neurocontroller incurs an economic opportunity cost. Improved performance over 55 [$/min] is attainable with ANS. The average increase in S_F (i.e. off-set) presents an opportunity for greater venture profit. ANS achieves a substantially increased average profit of 94 [$/min] (Figure 4-30), which is only slightly below the attainable 106 [$/min] possible for the increased $\mathrm{S_F}$ (Table 4-8).

As seen in Figure 4-30, the ANS neurocontroller has a larger IP standard deviation than the original neurocontroller. ANS tracks the optimal IP that is due to the gaussian disturbance in S_F. For high S_F values over extended periods (Figure 4-29b, between samples 2000-2250), an IP of 120 [$/min] was attained, though a maximum of 130 [$/min] is attainable (Table 4-8). For unusually low S_F values over extended periods, the swarm attained a minimal profit of 60 [$/min]. The optimal profit for this unfavourable process condition is 69 [$/min] (Table 4-8). ANS thus approximates the changing optimal IP. A small difference remains, because S_F changes substantially over time periods that are too short for the swarm to adapt completely. The swarm is thus essentially tracking the moving average of S_F. Nevertheless, the IP for ANS control exceeds the highest IP for the original neurocontroller at all times (Figure 4-30). ANS offers considerable benefits over the generalisation offered by the original neurocontroller.

4.12.2.2 Avoiding Hard Process Constraints

Figure 4-31 illustrates the swarm's ability to avoid the process search limits. Recall that the IP increases as the bioreactor level increases. The swarm neurocontrollers thus searched for control policies that increased the bioreactor level. Consequently, the swarm moved towards the high level alarm during on-line operation. The high level alarm of 5.95 [m] should never be exceeded; preserving the safety margin before bioreactor shutdown. A neurocontroller's fitness was penalised severely for exceeding the high level alarm. Such a penalised fitness was always lower than the fitness of a neurocontrol policy that remained within the search boundaries. Neurocontrollers with a penalised fitness, no longer guided the swarm and the swarm moved away from the high level alarm. In Figure 4-31 at 3000 sample periods, the trend line indicates a move away from the high level alarm. Shutdown at a reactor level of 6.2 [m] was thus safely avoided in ANS' on-line search.

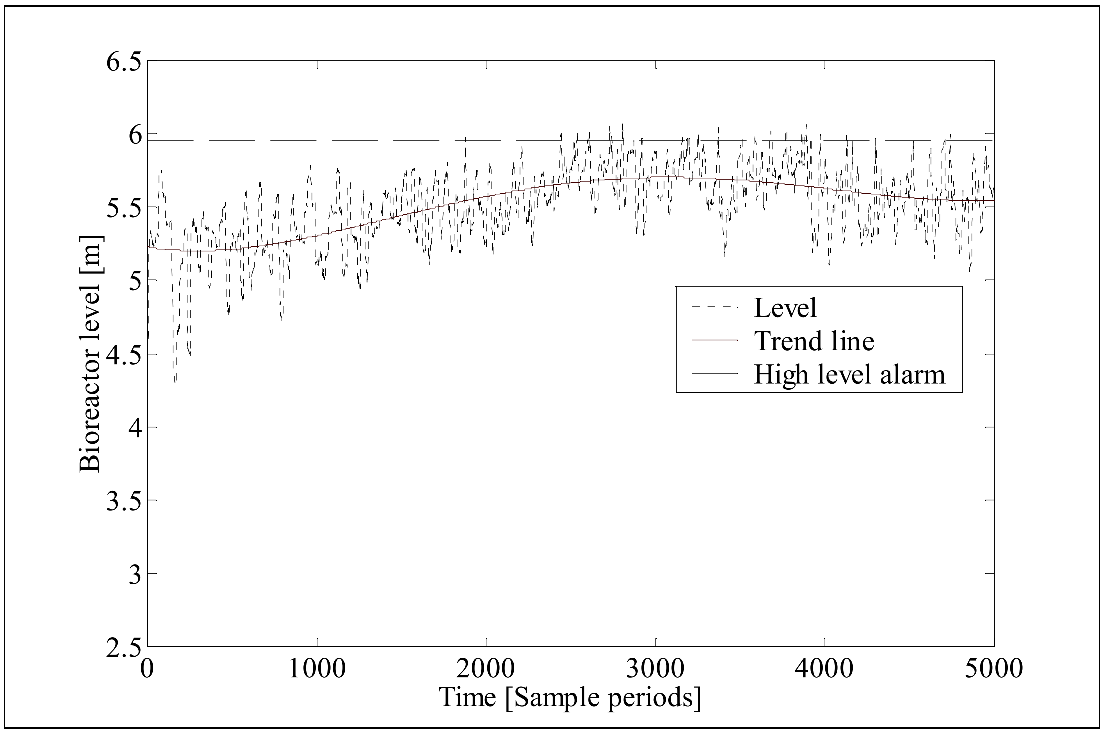


Figure 4-31 - Avoiding the hard level constraint. The trend line (solid) illustrates the swarm moving away from high level alarm set at 5.95 [m]. Process shutdown is thereby avoided.

4.12.2.3 Neuron Weight Adaptations

Each neuron in a neurocontroller has a particular functionality that is a partial solution to the control task. A neuron's weight vector determines its functionality. The changes to a neuron's weight vector during adaptation, provides insight into how its functionality changed in response to the changing process conditions. Principal component analysis allows visualisation of neuron weight vectors and therefore neuron functionality.

Figure 4-32 is a principal component plot of the weight vector of each neuron in the swarm's current best neurocontroller. After each adaptation, all the neuron weight vectors for the best swarm neurocontroller were plotted in Figure 4-32 as circular markers. The markers thus represent the history of adapted neuron functionalities.

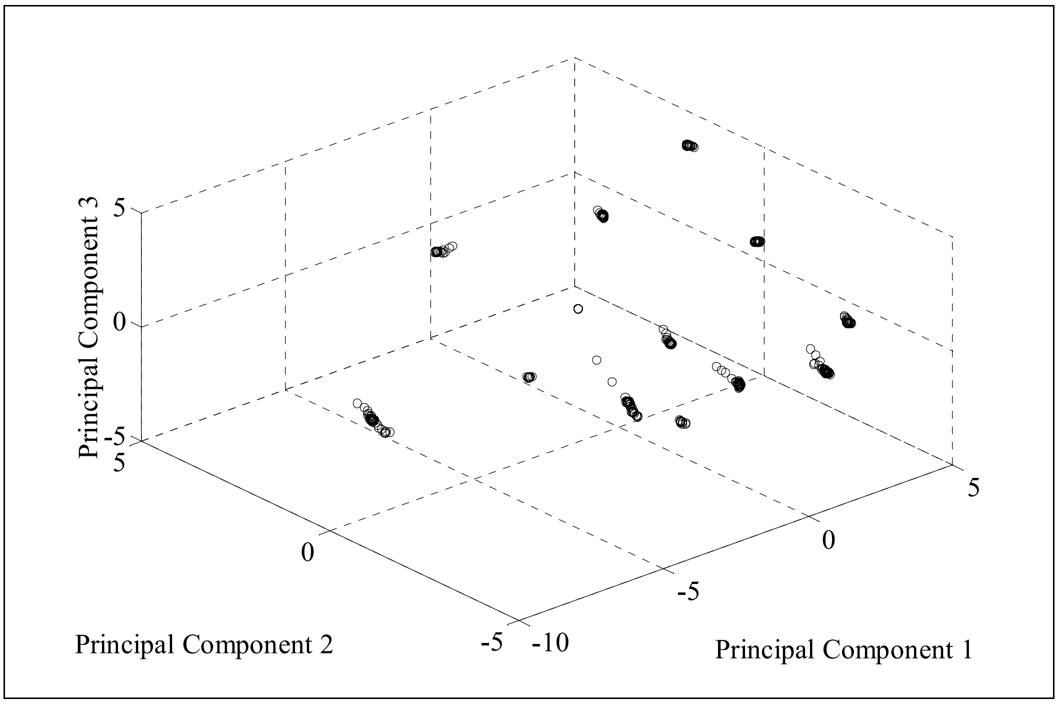


Figure 4-32 - Principle component analysis of neuron functionality (85% variance explained). Circular markers represent neuron weight vectors. Each cluster represents the change in neuron functionality due to adaptation. The extent of each neuron's adaptation is determined by the reigning process changes.

In Figure 4-32, the clusters indicate the different neuron functionalities that solve the control task. A cluster that is distributed over a larger region of the neuron weight space, had undergone a greater degree of on-line adaptation to its functionality. The extent of each neuron's adaptation is determined by the reigning process changes.

4.12.3 Discussion and future work

ANS' exploitative search preserves the existing neurocontroller's generalisation. For the bioreactor, adaptation failure (i.e., shutdown) never occurs during extensive implementation. Also, instability is never induced in the control response. The bounded nature of each neuron cluster in Figure 4-32 provides insight into how ANS preserves generalisation. Each neurocontroller retains memory of its best position (equation 4-14) during the five iterations between initialisations. As the fitness landscape changes, the fitness value of a neurocontroller's best position is no longer valid. A neurocontroller's best position rather serves as an example of where previous good solutions have been found. Memory of past neurocontroller positions biases the search in the direction of good past solutions. This memory function preserves generalisation by considering both past and current process information in the search. Re-initialisation, which clears the swarm's memory, limits prolonged bias to past solutions. Without limiting memory of past solutions, a drifting optimum would be difficult to track.

ANS' search for optimal control policies in a changing process works as follows. Process changes affect each neuron's functionality differently. Some neurons consequently no longer contribute to optimal economic return. The functionality of such a neuron needs to be updated, while retaining information in its weight structure that is still valid.

Consider a neuron weight that is optimal once adapted to a fixed value, despite continued process changes. Such fixed weights correspond to process conditions that remain constant (e.g., fixed growth parameter). As described in section 4.11, the possible directions for adaptation are limited to the positional experiences of all the swarm neurocontrollers. In ANS, the swarm neurocontrollers align along such a fixed weight, preventing (as per equation 4-14) the swarm from moving along that particular weight dimension. After several ANS iterations, only weights still relevant to improving the IP are implicitly changed. Re-occurring process changes (e.g., S_F) govern which specific neuron weights are continuously changed to track the economic return. The dimensionality of the search is thus somewhat reduced. Figure 4-33 is a copy of Figure 4-32, except that adaptation trajectories are emphasised by drawing arrow lines through the clusters. Each neuron functionality (cluster) moves along a fixed trajectory in response to re-occurring process changes. ANS establishes these trajectories implicitly and exploits this swarm knowledge for greater economic return.

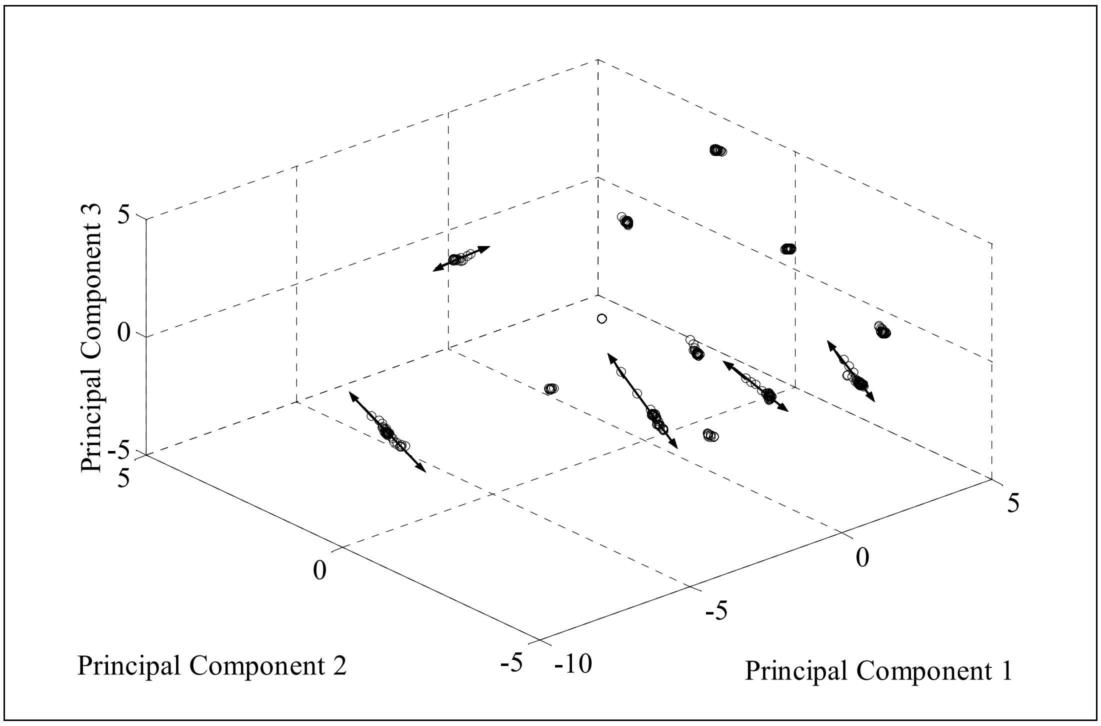


Figure 4-33 - Arrow lines indicate the trajectories of neuron functionalities in response to common (re-occurring) process changes such as S_F. ANS implicitly takes advantage of common process changes, which facilitates effective adaptation.

Future work will explicitly identify neuron functionalities that require adaptation. Such explicit knowledge may be used to further speed adaptation using fewer on-line evaluations. As ANS is a robust means for adapting neurocontrollers, it will be tested in other complex domains such as robotics and gaming.

Although neurocontrollers generalise their control actions in a changing process, such generalisation, though robust, may be economically sub-optimal. Adaptive Neural Swarming augments neurocontroller weights on-line, thereby garnering greater economic return from the changing process. ANS balances the need to adapt with the need to preserve generalisation. ANS also effectively avoids hard operating constraints during its on-line search. ANS implicitly identifies re-occurring process changes and uses this knowledge to speed adaptation. ANS is therefore a robust general tool for adapting of neural network controllers on-line. The greater economic return for the bioreactor case study suggests that neurocontrollers developed with SMNE would benefit significantly by implementing Adaptive Neural Swarming.

4.13 CONCLUDING REMARKS

Chapter 4 describes direct inverse control and model predictive control as two principal approaches to using neural networks in control architectures. Evolutionary methodologies to neurocontroller development have included single chromosome encodings, co-adaptive and cooperative evolutionary approaches. Cooperative evolutionary approaches have proven superior to single chromosome encodings, owing to more efficient credit assignment, maintained genetic diversity and a parallel search for partial solutions. SANE is a second generation cooperative EA for evolving neurocontrollers via implicit fitness sharing and the preservation of effective neuron combinations. SMNE is a third generation algorithm that improves on the SANE algorithm by incorporating the synergy provided by a local search combined with a global search (i.e., a memetic algorithm). The synergy between a local PSO algorithm and a symbiotic EA smoothes the fitness landscape, allowing for accelerated convergence to the global optimum. Similar to SANE, SMNE maintains genetic diversity via implicit fitness sharing. An ablation study gives statistical significance to the enhanced learning speed of SMNE over SANE for a simulated bioreactor benchmark. Although evolutionary algorithms such as SANE and SMNE impart significant generalisation to process uncertainty, adapting a neurocontroller on-line may improve economic performance significantly. ANS adapts neurocontrollers on-line using a local PSO implementation, appreciably improving economic return. The SMNE and ANS algorithms provide a comprehensive tool-set for developing intelligent, non-linear controllers for the process industries. The SMNE and ANS methodologies stand in sharp contrast to the tools offered by conventional plant-wide control approaches highlighted in chapter 2.

5 NEUROCONTROL OF BALL MILL GRINDING CIRCUITS

OBJECTIVES OF CHAPTER 5

- Illustrate the ERL plant-wide controller design approach applied to a high dimensionality, first principles non-linear model of a closed loop grinding circuit.
- Highlight the benefits of SMNE as opposed to current grinding control methodologies.
- Show effective set point tracking and rejection of common disturbances.
- Establish ERL techniques, such as SANE and SMNE, as a plant-wide control technique for the mineral processing industry.

5.1 INTRODUCTION

The simulation bioreactor benchmark in chapter 4 has only three state variables and four manipulated variables. Though the bioreactor benchmark has highly non-linear dynamics, the control problem remains a single unit operation with low dimensionality. The bioreactor benchmark, though multi-input multi-output, is not ideal for demonstrating whether a controller development algorithm is prone to the curse of dimensionality or significant process interactions. Mineral processing plants have interacting, non-linear dynamics and are typically comprised of numerous state variables (i.e., high dimensionality).

Mineral processing operations aim to concentrate a raw ore for metal extraction. First, valuable minerals are liberated from the ore matrix by grinding (e.g. ball mills) and classification (e.g. hydrocyclones) processes. The liberated minerals are consequently concentrated according to chemical or physical properties in flotation (i.e., surface chemistry) or magnetic separation. The majority of grinding circuits employ the principle of a closed loop consisting of a mill and a classifier (e.g., hydrocyclone). Ore is partly milled and fed to the classifier where the finer material is split off and the coarser material is recycled to the mill for further milling. A slurry sump, which accepts the mill discharge, acts as a buffer to fluctuations in the incoming flow from the mill.

The overall control objective is to produce a concentrate that maximises the venture profit of the concentrator. The role of the comminution circuit in the overall control objective is optimal liberation of valuable minerals, which may be exploited by the concentration processes. Liberation is not measurable on-line. Therefore, the production objective seeks to maintain a particle size distribution deemed optimal for subsequent concentration (Hodouin et al., 2001).

Another operational concern lies in the energy inefficiency of ore size reduction. With less than 10 [%] of the electrical power input contributing to grinding of the ore, the total operating cost of a grinding facility may contribute more than 50 [%] of the overall cost of mineral processing (Lo et al., 1996). To minimise operating cost, it is thus pertinent that the ore feed rate (i.e., throughput) remains in close proximity to the maximum design specification. This desirable operating state is constrained by the need to meet the particle size specification dictated by the flotation circuit. It is therefore essential that grinding circuits are subject to effective control in which both an optimal particle size distribution is provided to downstream operations and optimal utilisation of electrical energy is assured (Rajamani & Herbst, 1991).

Disturbances are prevalent during grinding circuit operation, viz. ore hardness changes, ore feed rate changes and feed particle size variations. The resulting sub-optimal particle size distribution affects downstream operations adversely. Variations in ore hardness and feed size variations may reduce the mill throughput drastically by perturbing the mass flow rate of the hydrocyclone overflow and particle size distribution continuously. In order to counteract the effects of these disturbances, a basic circuit usually has a minimum of two manipulated variables, viz. the fresh solids feed rate to the circuit and the dilution water to the sump. The most common process-manipulated variable pairings are illustrated in Table 5-1 (Rajamani & Herbst, 1991).

Table 5-1 - Common Single-Input-Single-Output (SISO) control loop pairings.

Pairing	*Controlled variable*		*Manipulated variable*
I	Particle size hydrocyclone overflow	$\rightarrow$	Sump water dilution rate
	Hydrocyclone feed rate	$\rightarrow$	Fresh solids feed (& dilution rate)
II	Particle size hydrocyclone overflow	$\rightarrow$	Fresh solids feed (& dilution rate)
	Hydrocyclone feed rate	$\rightarrow$	Sump water dilution rate

A high degree of controller interaction between the control pairings in Table 5-1 is evident. For pairing I, the dynamic response to an upward step in the product size, m_{OF}, set point illustrates this interaction. This set point change initially causes the first control loop to increase the sump dilution rate, Q_{Wsp}. An increase in the dilution rate to the sump causes an increase in the discharge flow rate, Q_{SP}, and a decrease in the solid concentration of the feed to the cyclone, $C_{S,SP}$, both of which have the initial effect of causing a finer classification. Although resulting in a finer classification, an increase in cyclone feed rate, Q_{SP}, at approximately constant solids concentration implies that a larger portion of the solids are classified to the underflow. The resulting larger mass flow rate of coarse material to the mill, M_{UF}, reduces the mill's ability to grind the material as fine as prior to the set point change. Hence the discharge from the mill slowly becomes coarser. Once this coarser particle distribution reaches the cyclone, the cyclone overflow particle size distribution becomes coarser and the particle size distribution to the overflow may return to a similar state as to before the step change in dilution rate, Q_{Wsp}. This open-loop response is illustrated in Figure 5-1 (Barker & Hulbert, 1983).

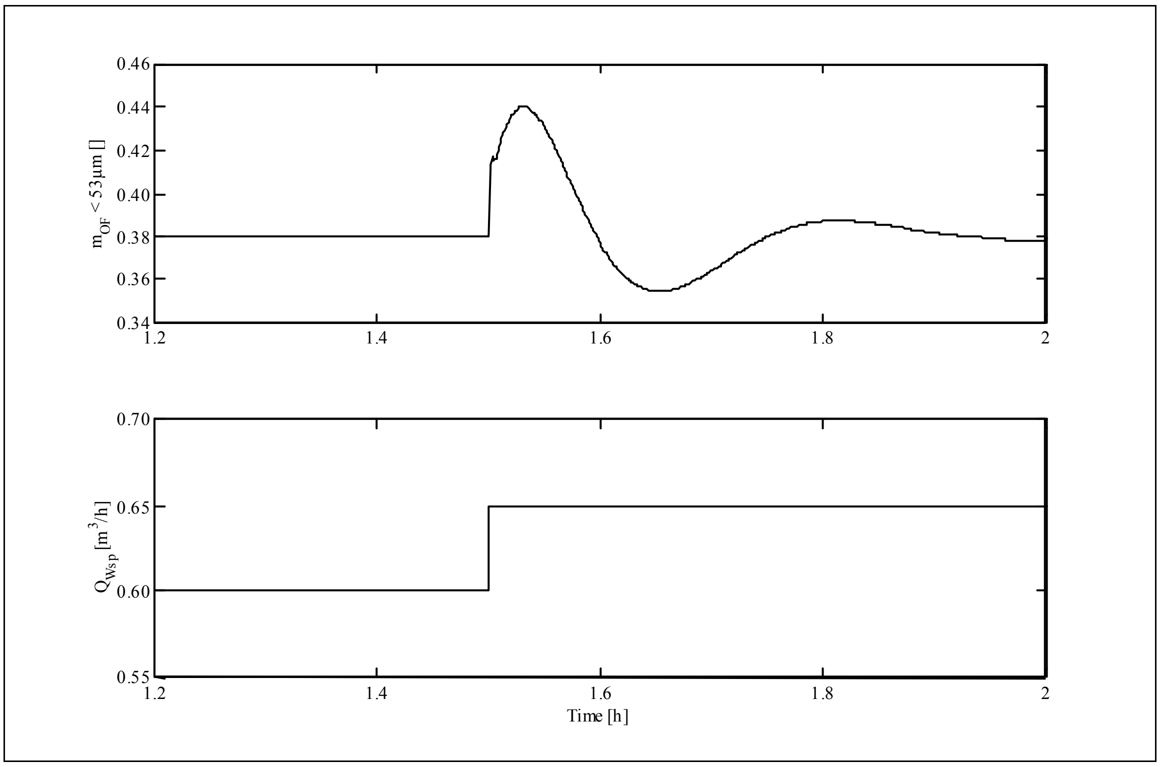


Figure 5-1 - Particle size response (m_{OF}, top) in hydrocyclone overflow in response to an open loop step change in dilution rate (Q_{Wsp}, bottom).

The increase in the hydrocyclone feed rate, Q_{Wsp}, initiated by the first control loop consequently causes the second control loop in pairing I to decrease the fresh feed to the mill, M_{FF} (reducing the mill production rate). This is primarily a response to the mill's inability to cope with the larger amount of coarse particles in the mill. The long-term change in product size (Figure 5-2) is thus mainly a result of a change in fresh solids feed rate, but this control configuration (pairing I) initially results in an upset to the cyclone in order to attain the set point change. Continuous interaction between the two control loops may lead to prolonged upsets to the product specification, before the new set point is finally reached. Pairing II similarly results in significant controller interaction (Barker & Hulbert, 1983).

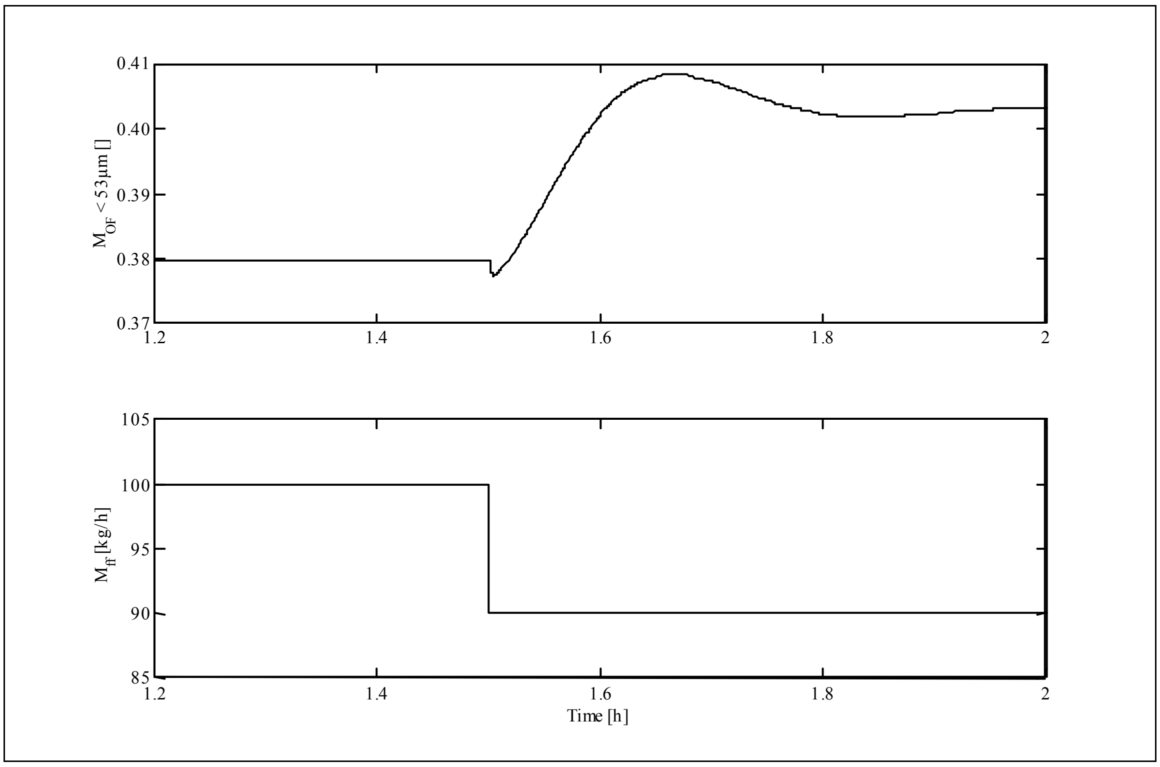


Figure 5-2 - Particle size response (m_{OF}, top) in the hydrocyclone overflow, after an open loop step change in fresh solids feed rate (M_{FF}, bottom).

The time constant of the hydrocyclone is negligible compared to any of the other time constants in the grinding circuit, necessitating precise hydrocyclone control to prevent short-term fluctuations from affecting the product. As particle size instrumentation is prone to time delay, any form of control based on the measurement of particle size may be ineffective to high frequency disturbances to the hydrocyclone. As the feed flow to the cyclone has a significant effect on the behaviour of the cyclone and thus on the behaviour of the rest of the circuit, the sump discharge flow rate should be as steady as possible. Many control strategies have regarded the sump level as a separate entity from the mill control, in that, the sump level is controlled separately by a single-input-single-output (SISO) controller. This reduces the rigour of the controller design methodology, by ensuring that the remainder of the circuit is open loop stable. However, stringent SISO level control results in continuous variation of the flow rate to the hydrocyclone. High performance sump level control, i.e., control that contributes to the control objectives of the entire circuit, should ensure a steady flow to the cyclone and thus reduce the likelihood of short-term fluctuations in the product size distribution. In view of the severe interaction between controlled and manipulated variables (Table 5-1), better control may be achieved by incorporating mill rotation speed into the control law. Mill speed has been proposed as a less interactive manipulated variable. Mill speed may be considered ideal as it directly affects the grinding kinetics of the mill and therefore should eliminate the interactions caused by control through flow rate manipulation. The basic premise is that a build-up of coarse material in the mill should result in an increase in mill speed. Large variable speed drive motors would allow for continuous adjustment in mill speed (Herbst et al., 1983). As a minimum requirement, a decoupled SISO arrangement or a multi-input multi-output (MIMO) controller design is thus required to eliminate negative controller interaction (Barker & Hulbert, 1983).

Lo et al. (1996) demonstrated the benefits of advanced control and design optimisation in reducing unit production costs thereby increasing profits. Both steady state and dynamic optimisation are proposed. Steady state optimisation evaluates the design and operating variables that impact the specific energy consumption (kWh) at a given product specification. Lo et al. (1996) demonstrated that appropriate grinding circuit modifications may result from combined experimental and simulation work. Batch grinding tests and simulation revealed that a reduction in the recharge ball size (i.e., design variable) and lower percentage solids (i.e., operating variable) increased the throughput from 10-13 [%] in an open-circuit industrial facility. This increased the revenue by $16.5 million per annum for an investment of $200 000. In another industrial application, a model-based expert system, together with an ore hardness estimator, increased the mill throughput by approximately 7 [%]. This corresponded to increased revenue of $8.4 million per annum for an investment of $250 000. As grinding circuits are often the process bottleneck, small increases in efficiency have a dramatic performance impact.

The objective of this chapter is to demonstrate the potential of evolutionary reinforcement learning, in particular the SMNE algorithm (section 4.6), for the development of non-linear (neuro)controllers for ball mill grinding circuits. The effective elimination of process interaction, which has plagued multi-loop SISO designs, is shown. The high performance of the developed neurocontroller is demonstrated for a wide variety of disturbances, viz. ore hardness and feed particle size distribution changes. As shown, this novel approach to the design of high dimensionality, non-linear controllers can incorporate all possible process and manipulated variables (including mill speed) into an optimal control law (i.e. neural network), which effectively deals with the control challenges posed by ball mill grinding circuits.

5.2 CURRENT CONTROL TECHNIQUES

Although grinding circuits exhibit non-linear dynamic behaviour, controller design has largely been investigated from a linear controller perspective (e.g. PID control). Linear controller design techniques (modern and classical) require the use of linear process models. As rigorous grinding circuit models are necessarily non-linear, the non-linear models need to be linearised in the vicinity of a predetermined economically optimal operating region. Should the state space be highly non-linear in this region of the state space, the developed linear controller may not remain robust (or severely degrade in performance) once operation strays significantly from the region where the linearisation applies. Nevertheless, linear MIMO control strategies have been applied with significant success in industrial grinding circuits.

5.2.1 Linear multivariable control design

Rajamani & Herbst (1991b) implemented a multi-loop SISO feedback strategy for an experimental pilot-scale grinding circuit. The sump level and percentage solids in the fresh feed were controlled separately with PI controllers, thereby assuming that these process variables do not contribute to the overall control objective. A variable speed pump controlled the sump level through PI control. The percentage solids in the fresh feed was controlled with a PI ratio controller, maintaining the fresh feed water in proportion to the fresh ore feed, thereby assuming that 60 [%] (m/m) solids in the fresh feed is optimal. The exclusion of these controlled variables from the multi-loop SISO design, limits the integration advantages provided by a more plant-wide control approach. In a multivariable system, PID controllers are limited by a fixed structure and controller pairing uncertainties, which require an effective tuning method. In a 2x2 control system the tuning of one controller impacts the tuning of the other controller. The controller design thus focuses on the control system structure (i.e., pairing I or pairing II in Table 5-1) and the controller parameter tuning. Figure 5-3 illustrates the implementation.

Herbst & Rajamani (1979) compared the control strategies presented by pairing I and pairing II. Recall that the response of the overflow product size, m_{OF}, to changes in fresh feed rate (M_{FF}) is delayed, owing to the lag introduced by the mill and the sump. The particle size response in the cyclone overflow responds rapidly to sump water addition ($G_{p11}(s)$ in Figure 5-3), owing to the rapid classification in the cyclone to changes in cyclone feed percentage solids, f_v. Similarly, the mill throughput response is rapid with respect to the solids fresh feed rate ($G_{p22}(s)$ in Figure 5-3). For this reason, pairing I was confirmed superior to pairing II. Conventionally, the proportional and integral parameters of the PI controllers are determined by trial-and-error, which requires no modelling exercise. Trial-and-error tuning is prone to oscillatory responses, as limited consideration is given to decoupling the two controllers. The impact of $G_{p12}(s)$ and $G_{p21}(s)$ in a system without decoupling is pronounced (Figure 5-3). Herbst & Rajamani (1991b) determined the controller parameters for the multi-loop SISO system using "simulation tuning". A simulation model of the pilot grinding circuit was used to maximise an objective function that aimed to maximise mill throughput while minimising the errors from the set points. A Box-Wilson search determined the best parameter tunings. Nevertheless, oscillations were observed in the particle size passing 44 [µm] in the pilot plant implementation, due to the process interaction described in section 5.1. Detuning of the simulation-tuned controllers was also necessary to compensate for inappropriate control actions to measurement noise.

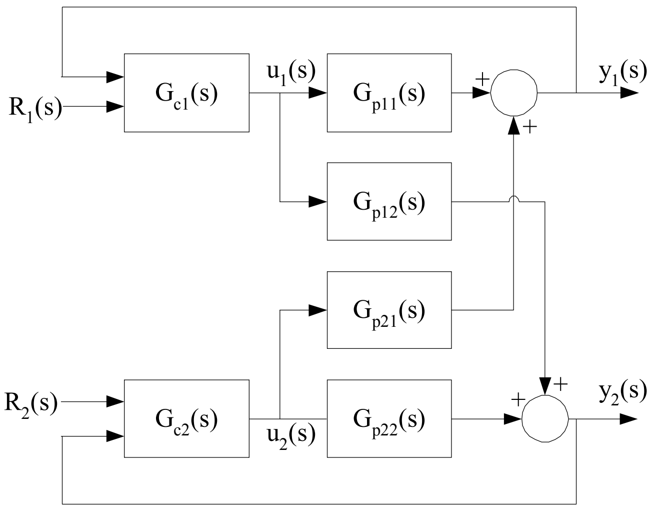


Figure 5-3 - Distributed SISO control without decoupling.

Multivariable decoupling strategies have proven highly effective in overcoming controller interaction. Decoupling strategies aim to augment the output of distributed PI SISO controllers (i.e., PI controllers) as in Figure 5-4. Inverse Nyquist Array (INA) multivariable control has proven highly successful at overcoming controller interaction in closed milling circuits. Consequently, INA has been applied in numerous industrial applications (Hulbert et al., 1980; Gossman and Buncombe, 1983; Barker and Hulbert, 1983; Jämsä et al., 1983; Hulbert et al., 1990) Hulbert et al. (1990) implemented a multivariable controller using INA methods. INA is an interactive design method in the frequency domain with the objective of decoupling the controller interactions. The linear INA distributed multi-loop SISO controller was implemented in a run-of-mine (ROM) milling unit. Hulbert et al. (1990) demonstrated the effective use of a continuous particle-size monitor (PSM) in the industrial control application. After obtaining sufficient process data from dynamic tests, a linear dynamic model was constructed with the product size (PSM), mill load (MLOAD), sump level (SLEV) and cyclone feed density (CFD) as the process variables. The manipulated variables were the sump dilution feed rate, the fresh solids feed rate, the slurry feed rate to the cyclone and the water feed rate to the mill inlet. The control of the CFD was omitted and the INA controller was developed to control MLOAD, SLEV and PSM. The water feed to the mill inlet was disregarded as a manipulated variable in the INA design. The overflow particle size was paired with the sump water addition rate, the mill load with the fresh feed rate and the sump level with the cyclone feed rate. The outputs of the three PI controllers were intermediate variables fed to the INA matrix compensator. The matrix compensator was designed so that the corrective action to one controlled variable changes all the manipulated variables to some degree, but does not change the other controlled variables significantly. Despite the non-linear behaviour of the milling circuit, the 3x3 INA controller proved highly effective. The INA controller reduced the standard deviation in the particle size specification of the cyclone overflow and increased the mill throughput. This decoupled approach demonstrated the benefit of multivariable control for milling circuits, though the application was linear. Effective decoupling of the process variables was achieved and accepted into continuous industrial implementation.

The success of a decoupling scheme for multi-loop SISO control is highly dependent on the accuracy of the process models. A rigorous identification program contributes markedly to the successful implementation of such controllers. Hulbert et al. (1990) obtained perturbation data for model identification from 34 tests over a period of two months. Perfect decoupling may be difficult to achieve, though partial decoupling may be beneficial to the control objectives. Foss (1973) emphasised that there is frequently no compelling argument to impose decoupled servo responses on processes, particularly as the main objective is rejecting disturbances that reduce the mill throughput. Also, interactions between process variables may enhance control, provided dynamic information in an adequate modelling exercise is exploited by an appropriate control theory.

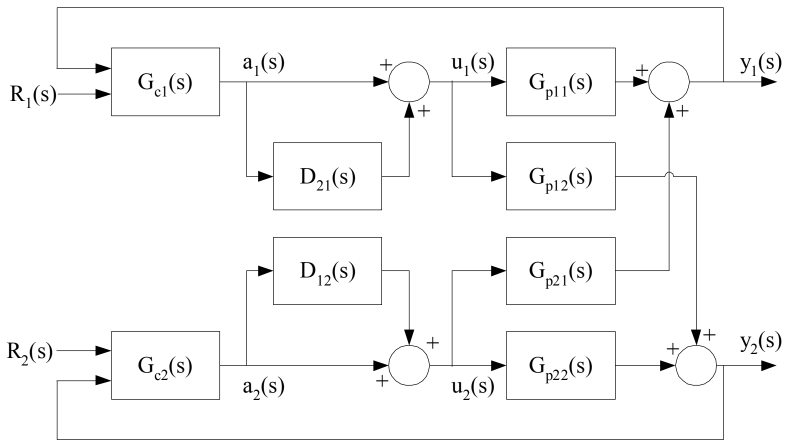


Figure 5-4 - Distributed SISO controllers with decoupling.

A more implicit decoupling or exploitation of interactive controlled variables results through the use of state space methodologies (Ylinen et al., 1983; Rajamani & Herbst, 1991b). Modern control is a state space methodology that offers a complete multivariate (MIMO) approach to mill controller design (Rajamani & Herbst, 1991b). The MIMO design had particle size in the cyclone overflow and the mill throughput as controlled variables, with the sump dilution rate and the solids fresh feed as manipulated variables. No prior knowledge of appropriate controller pairings is required in a state space design. Modern control determines a manipulated variable's control action based on all the state variables in a linear model, as opposed to feedback control that uses a single error input. Rajamani and Herbst (1991a) developed a rigorous non-linear dynamic model of a pilot scale ball mill circuit. To use the rigorous non-linear model in a modern control framework, the model was linearised. The linearisation is applicable to a narrow operating range around the set points, which also limits the operating region in which the linear control law is optimal in the non-linear process. Full advantage could thus not be taken of the available non-linear model. The control system is complicated by unmeasured state variables, which required an observer (state estimator) in the control block. Furthermore, a pure integrator must generally be added to account for process / model mismatch. However, the modern control dealt effectively with process interaction. Rajamani & Herbst (1991b) confirmed experimentally that an optimal control strategy performed significantly better than an optimised multi-SISO design with smooth settling to set point changes (i.e., minimal oscillation) and required no detuning of controller parameters.

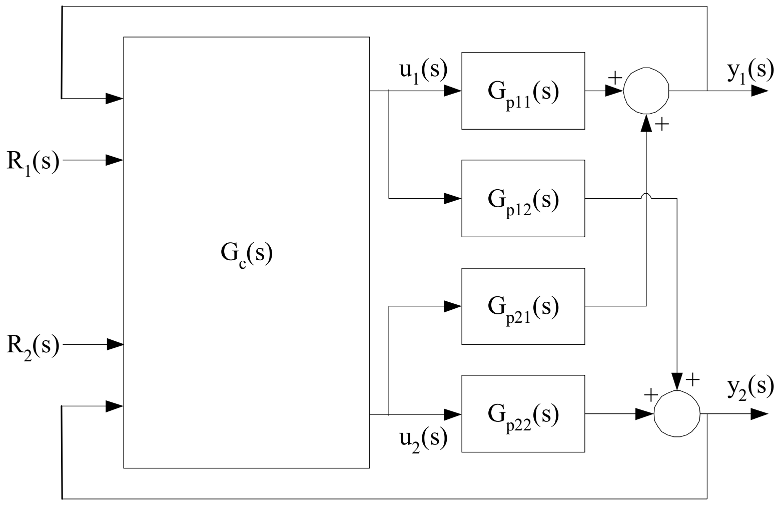


Figure 5-5 - Complete multivariable controller.

Desbiens et al. (1994) used an adaptive distributed general predictive control (GPC) design in a simulation study for grinding circuit control. The implemented predictive controller was linear. A PI controller controlled the sump level and a proportional controller controlled the water addition to the rod mill, both independent of the GPC scheme. Historical data were used to obtain dynamic process information and construct linear discrete polynomial models. One GPC controller paired the cyclone overflow particle size with the sump water addition. The other GPC controller paired the circulation load (i.e., throughput) with the fresh ore feed rate. At each sampling period, the model parameters were re-estimated and a sequence of multi-step control actions were calculated which minimised the cost functions. The first control action was implemented and the process repeated at the next sample. The model parameter adaptation was the key to the control system, though the control system's distributed nature complicated the model identification for each GPC controller. The adaptive GPC scheme had 39 [%] lower summed squared error than a fixed GPC implementation for particle size set point tracking. The adaptive GPC scheme also improved the tracking error of the circulation load by 8 [%]. However, identification was difficult during external disturbances to the process. Also, a multivariable (not distributed) GPC scheme has advantages due to insensitivity to controller pairings and the internal elimination of process interaction.

Pomereau et al. (2000) compared the performance of decentralised PID controllers without explicit decoupling (i.e., co-tuning), a partially decoupled PID control scheme, internal model control and adaptive linear GPC. Pomereau et al. (2000) found that distributed controllers performed comparably with the multivariable controllers, provided the coupling of the process was taken into consideration and the pairings were selected appropriately. The adaptive GPC had superior performance to the fixed controllers in the event of process / model mismatch.

Linear multi-loop SISO and MIMO control may be further improved through non-linear identification and controller development that offer substantial opportunities for operation over a wider region of the state space. Non-linear controllers cope with a wider variety of disturbances and process uncertainties, making non-linear controller design based on non-linear process models highly desirable.

5.2.2 Non-linear control

The non-linear nature of grinding circuits makes process identification difficult. Furthermore, determining the structure of a useful model is non-trivial. Neural networks overcome the identification uncertainties presented by input-output plant data. Neural networks offer a non-linear, self-organising structure for developing non-linear models with minimal user input. Flament et al. (1993) gathered process data from a rigorous grinding simulator for training feedforward neural networks. Both non-linear model predictive control and direct inverse control was considered. An adaptation scheme for the neural network weights also improved settling performance. However, the implementation adds one MISO non-linear controller to an otherwise distributed, linear control objective. The fresh feed dilution rate, the sump level and the fresh solids feed was controlled by PI controllers. The non-linear control scheme is limited to controlling only the particle size in the cyclone overflow, with no consideration to maximising the mill throughput. The circulating load and the particle size in the cyclone overflow were the process variables, with the sump dilution rate as the only manipulated variable. This neurocontrol implementation makes a decoupling arrangement with the other PI controllers difficult. Significant process interaction is assumed to exist with the PI controller for mill throughput as discussed in section 5.1.

Cho et al. (1997) also applied neural network based model predictive control (MPC) to SAG mill control. A neural network model and a linear state space model were derived from sampled plant data. The control study was implemented in a simulated environment. Linear MPC and LQC (linear quadratic) control were compared to a non-linear MPC scheme with the neural network as model. The manipulated variables were SAG speed, SAG solids feed rate and sump pump speed, with the overflow particle size as only controlled variable. The control objective had no consideration for maximising the mill throughput. The non-linear MPC had faster simulated settling times than both the linear MPC and LQC schemes. This was attributed to the advantages of using non-linear model identification.

Martin and McGarel (2001) applied a non-linear MPC scheme to a cement plant's closed ball mill circuit. The significant dead-time between a change in the fresh feed to a response in particle size was cited as justification for MPC. A neural network model was created from logsheet data, which provided non-linear gains for non-linear MPC. On-line application revealed that non-linear MPC increased production, improved cement consistency and quality and provided effective feed disturbance rejection, though only simulated results were presented.

Flament et al. (1997) and Desbiens et al. (1997) used a rigorous dynamic grinding circuit model, based on population balances, to evaluate numerous control strategies. After model parameter estimation from experimental data, the simulator conformed to the dynamic responses of the plant. Controlled variable pairings were selected using a relative dynamic gain technique (i.e., frequency domain). Decoupled controller tuning was resolved using a frequency based method. Though the dynamic simulator offered non-linear simulation, the controller development was confined to linear analysis.

The following section details the non-linear phenomenological model, used by the SMNE algorithm in entirety to develop a plant-wide neurocontrol strategy.

5.3 GRINDING CIRCUIT SIMULATION MODEL

5.3.1 Pilot plant simulation model

Rajamani & Herbst (1991a) developed a dynamic model for a pilot scale grinding circuit (Figure 5-6). The simulation model is based on a ball mill with an internal diameter of 0.76 [m] and a length of 0.46 [m]. The simulation parameters reflect a ball load of 345 kg, which corresponds to a 40 [%] mill filling. The mill discharge is fed to a sump with a 0.3 [m] diameter and a 0.8 [m] height. The 0.075 [m] hydrocyclone classifies the feed from the sump.

For the purposes of this control study it is assumed that sensors are available to measure the mass flow rate for the fresh ore feed (limestone with a density of 2500 kg/m^3 and particle size distribution -1680 μm), and the solids concentration in both the sump discharge and the cyclone overflow streams. Also, volumetric flow metering is provided in the cyclone overflow stream. A particle size analyser provides the fraction passing 53 [μm] with a sample interval of 2 [min]. The solids concentration and volumetric flow sensors give an indication of the mass flow rates of the solids and water.

The model equations (Equations 5-1 to 5-17) in the following sub-sections were solved simultaneously using Euler's method.

5.3.2 Ball mill model

The population balance concept may be applied to the particle breakage processes occurring in a ball mill. A linear, size-discretised model for breakage kinetics may be derived by selecting n size intervals into which the particulate material may be divided, with d_1 being the maximum size and d_n the minimum size. The *ith* interval is thus bounded by d_i above and d_{i+1} below. The mass balance for the mass fraction in each size interval, $m_i(t)$, may consequently be expressed in equation 5-1. In this study, 13 size intervals were considered from -2380 [μm] to -37 [μm] with size intervals of $\sqrt{2}$, leading to 13 differential equations with the general form of equation 5-1 (Rajamani & Herbst, 1991).

$$\frac{dH(t)\cdot m_{MP,i}(t)}{dt} = M_{UF}\cdot m_{UF,i} + M_{FF}\cdot m_{FF,i} - M_{MP}\cdot m_{MP,i}(t) \\ - S_i\cdot H(t)\cdot m_{MP,i}(t) + \sum_{j=1}^{i-1} b_{ij}\cdot S_j\cdot H(t)\cdot m_{MP,j}(t) \tag{5-1}$$

where *H(t)* is the total particulate mass hold-up in the mill. M_{UF}, the underflow mass feed rate from the hydrocyclone, and M_{FF}, the fresh ore mass feed rate, cumulatively represent the total feed, M_{MF}, to the ball mill (Figure 5-6). The mixing of the two feed streams, M_{FF} and M_{UF}, is assumed to be perfect before the total feed enters the mill. The size-discretised selection function, S_i, is the rate at which material is ground out of the *ith* size interval. The size-discretised breakage function, *bij*, represents the fractions of the ore in the size interval *j* that is broken into the following smaller size intervals. The ball mill is modelled as uniformly mixed, which was found to be a fair assumption based on residence time tracer tests (Rajamani & Herbst, 1991a).

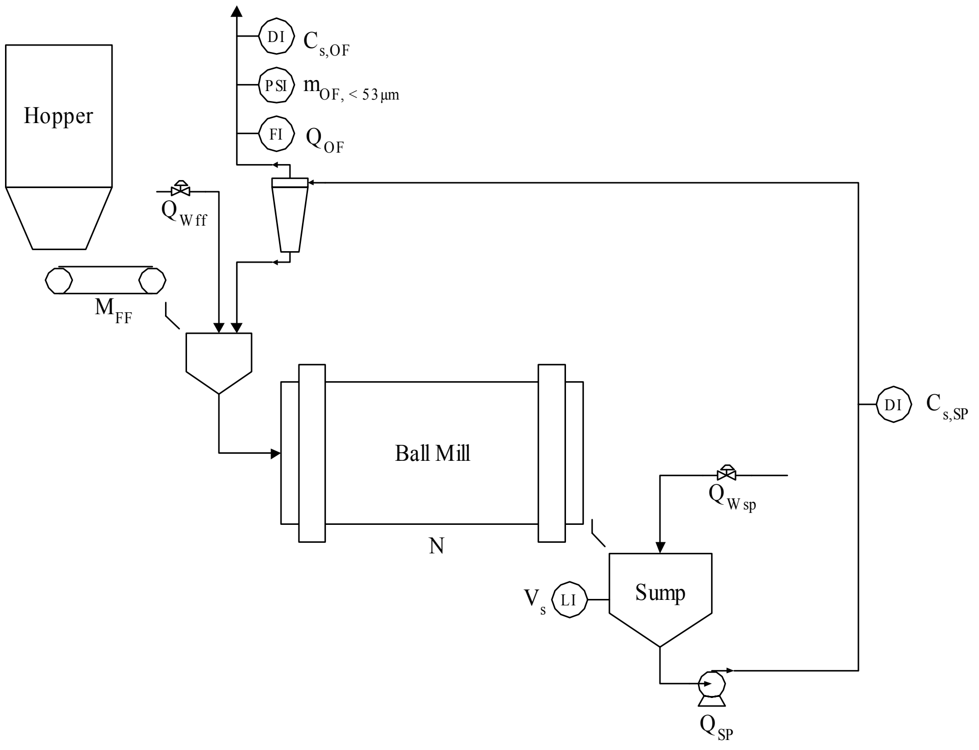


Figure 5-6 - Grinding circuit flow diagram as modelled by Rajamani & Herbst (1991).

For overflow mills, the volume of slurry in the mill is reasonably constant over a wide range of operating conditions. However, the effective slurry volume, V_M, is expected to vary as the fraction of mill critical speed, *N*, varies. This steady state relationship is expressed by equation 5-2 (Herbst et al., 1983). It is assumed that the time constant for the dynamics relationship between mill volume and mill rotation speed is small.

$$V_M = 0.0899 \cdot N^2 - 0.0818 \cdot N + 0.0482 \tag{5-2}$$

For the computation of mill hold-up, the solids concentration (mass of solids per unit volume of slurry), $C_{S,\,MP}$, is computed utilising Equations 5-3 to 5-4. The volumetric feed rate to the mill is assumed to equal the volumetric discharge rate at all times.

$$H(t) = V_M \cdot C_{S,MP} \tag{5-3}$$

$$\frac{dC_{S,MP}}{dt} = \frac{Q_{MF}}{V_M} \cdot \left(C_{S,MF} - C_{S,MP} \right) \tag{5-4}$$

Selection functions for a particular material are proportional to the specific power drawn by the mill as described in equation 5-5. The power, *P*, drawn by the mill is determined by the Bond power correlation (equation 5-6), which is influenced by the fraction of mill critical speed, *N*. The specific selection function, S_i^E, is dependent on the fineness of the product in the mill. However, in the pilot plant model developed by Rajamani & Herbst (1991), a single set of selection functions described by equation 5-7 was deemed applicable in the fresh solids feed range 90-136 [kg/h]. The fresh solids feed rate was limited to this range in this control study to ensure model validity. To good approximation, the cumulative form of the breakage function, B_{ij}, may be described by equation 5-8. The feed size to the mill is linearly transformed to the mill discharge particle size distribution, which is dictated to by the selection and breakage functions (Rajamani & Herbst, 1991a).

$$S_i = S_i^{\,E} \cdot \left(\frac{P}{H} \right) \tag{5-5}$$

$$P = M_{balls} \cdot \left[3.1 \cdot D^{0.3} \cdot (3.2 - 3 \cdot M_b) \cdot N \cdot \left(1 - \frac{0.1}{\left(2^{9-10 \cdot N}\right)} \right) - 1.13 \right] \tag{5-6}$$

$$S_i^{\,E} = 10.6 \cdot \left(\sqrt{d_i \cdot d_{i+1}} \cdot \sqrt{d_1 \cdot d_2} \right)^{1.427} \tag{5-7}$$

$$B_{ij} = 0.31 \cdot \left(\frac{d_i}{d_{j+1}} \right)^{0.48} + 0.69 \cdot \left(\frac{d_i}{d_{j+1}} \right)^{2.8} \tag{5-8}$$

5.3.3 Sump model

An agitator suspends the slurry in the sump. Along with the assumption of uniform mixing in the sump, no particle size changes are assumed to occur due to abrasion. Equations 5-9 to 5-11 provide the dynamic behaviour of the sump, where M_{SP} is the sump discharge mass flow rate, Q_{SP} is the volumetric discharge rate of slurry, Q_{Wsp} is the volumetric dilution rate and $m_{SP,i}$ is the fraction of size interval i in the sump. Equation 5-9 is the mass balance for the particle size intervals and equation 5-10 represents the change in sump volume, V_s, during operation (simple mixing rule applies). The overall mass balance for the sump is described by equation 5-11.

$$\frac{dm_{SP,i}}{dt} = M_{MP} \cdot m_{MP,i} - M_{SP} \cdot m_{SP,i} \tag{5-9}$$

$$\frac{dV_S}{dt} = Q_{MP} + Q_{Wsp} - Q_{SP} \tag{5-10}$$

$$\frac{d}{dt}\left(V_S \cdot C_{S,SP}\right) = Q_{MP} \cdot C_{S,MP} - Q_{SP} \cdot C_{S,SP} \tag{5-11}$$

5.3.4 Hydrocyclone model

Rajamani & Herbst (1991a) deemed the use of a dynamic model for the hydrocyclone unnecessary. The dynamic response of the hydrocyclone is assumed orders of magnitude faster than the other time constants in the grinding circuit. An empirical model was utilised as described by the model equations 5-12 to 5-17. The water flow rate in the overflow, *WOF*, is determined by the water feed rate to the hydrocyclone, *WF*, as prescribed by the water split equations 5-12 and 5-13. The particle size, d_{50}, at which 50 [%] of the solids are classified to the overflow and 50 [%] of the solids are classified to the underflow, is described by equation 5-14. In equation 5-14, Q_{SP}, is the slurry volumetric feed rate to the hydrocyclone and, f_v, is the volume fraction of solids in the slurry feed. To account for short circuiting of fines to the underflow, R_f, is defined in equation 5-15. R_f is used with the corrected efficiency curve, Y_i (eq. 5-16), to calculate, E_i (eq. 5-17), which represents the fraction of particles in size interval i classified to the underflow.

$$WOF = 1.363 \cdot WF - 10.75 \qquad \text{for WF} < 21.4 \text{ [kg/min]} \tag{5-12}$$

$$WOF = 0.837 \cdot WF + 0.35 \qquad \text{for WF} > 21.4 \text{ [kg/min]} \tag{5-13}$$

$$\log_e d50 = 3.616 - 15.006 \cdot 10^{-2} \cdot Q_{SP} + 20 \cdot f_v \tag{5-14}$$

$$R_f = 0.818 - 0.7932 \cdot \frac{(WF - WOF)}{WF} \tag{5-15}$$

$$Y_i = 1 - e^{\left[-0.6931 \cdot \left(\frac{d_i}{d_{50}}\right)^{1.6}\right]} \qquad \textbf{(5-16)}$$

$$E_i = Y_i \cdot (1 - R_f) + R_f \qquad \textbf{(5-17)}$$

Rajamani & Herbst (1991a) determined the 8 empirical coefficients in equations 5-12 to 5-17 over the cyclone feed range 15-30 [L/min] with a feed solids volume percentage ranging from 30-50 [%]. To ensure model validity these constraints were adhered to during simulation.

5.4 NEUROCONTROLLER DEVELOPMENT AND PERFORMANCE

5.4.1 SMNE implementation

The rigorous non-linear model in section 5.3 was used in the SMNE algorithm (section 4.6) to develop a feed-forward neurocontroller with 6 input nodes (i.e., process variables), 12 hidden nodes and 5 output nodes (i.e., manipulated variables). The 6 inputs nodes and the 5 output nodes are listed in Table 5-2.

Table 5-2. Description of neurocontroller inputs and outputs.

Input nodes	*Output nodes*
Product mass fraction < 53 [µm], m_{OF}	Solids fresh feed mass flow rate, M_{ff}
Product mass fraction < 53 [µm] set point	Fresh feed water dilution addition rate, Q_{Wff}
Sump volume, V_s	Sump water dilution addition rate, Q_{Wsp}
Product volumetric flow rate, Q_{OF}	Sump volumetric discharge rate, Q_{SP}
Sump solids concentration, $Cs_{,SP}$	Fraction of mill critical speed, N
Product solids concentration $Cs_{,OF}$	

The operating range constraints for the manipulated variables (i.e., output nodes) in Table 5-3 are listed in Table 5-4.

SMNE was required to minimise the fitness function in equation 5-18 (i.e., maximise reward) at each time step. A neurocontroller is thus able to attain reward based on the effectiveness with which it tracks the set point, while maintaining the highest possible production rate. Note that economic considerations such as operating cost are not considered in the neurocontroller design, but could be added if available.

$$Fitness = \left| f_{OF,53\mu m} - f_{setpoint} \right| \cdot \left| 1 - \frac{C_{s,OF} \cdot Q_{OF}}{1200} \right| \qquad \textbf{(5-18)}$$

Table 5-3 - Operating ranges of manipulated variables.

Manipulated variable (Output nodes)	*Operating range constraint*	
Solids fresh feed mass flow rate, M_{ff}	90 - 136	$[kg \cdot h^{-1}]$
Fresh feed water dilution addition rate, Q_{Wff}	0.1 - 2.0	$[m^3 \cdot h^{-1}]$
Sump water dilution addition rate, Q_{Wsp}	0.1 - 3.0	$[m^3 \cdot h^{-1}]$
Sump volumetric discharge rate, Q_{SP}	0.1 - 2.0	$[m^3 \cdot h^{-1}]$
Fraction of mill critical speed , N	0.3 - 0.8	[-]

During each learning trial, the initial condition for the state variables is initialised with a gaussian distribution around the mean values listed in Table 5-4. Initialisation serves as a starting point for the dynamic optimisation conducted by the SMNE algorithm.

Table 5-4 - Mean initial conditions of the grinding circuit state variables.

Size interval [µm]	*Solids fresh feed mass fraction [-]*	*Mill mass fraction [-]*	*Sump mass fraction [-]*
2380 - 1680	0.150	0.07692	0.07692
1680 - 1190	0.150	0.07692	0.07692
1190 - 841	0.150	0.07692	0.07692
841 - 595	0.200	0.07692	0.07692
595 - 420	0.200	0.07692	0.07692
420 - 297	0.033	0.07692	0.07692
297 - 210	0.033	0.07692	0.07692
210 - 149	0.033	0.07692	0.07692
149 - 105	0.010	0.07692	0.07692
105 - 74	0.010	0.07692	0.07692
74 - 53	0.010	0.07692	0.07692
53 - 37	0.010	0.07692	0.07692
-37	0.010	0.07692	0.07692

Process variable	*Initial condition*	*Unit*
H	36	[kg]
V_s	0.0028275	$[m^3]$
$C_{s, SP}$	350	$[kg \cdot m^{-3}]$
Manipulated variable	*Initial condition*	*Unit*
M_{ff}	100	$[kg \cdot m^3]$
N	0.75	[-]
Q_{Wff}	0.005	$[m^3 \cdot h^{-1}]$
Q_{SP}	1	$[m^3 \cdot h^{-1}]$
Q_{Wsp}	0.6	$[m^3 \cdot h^{-1}]$

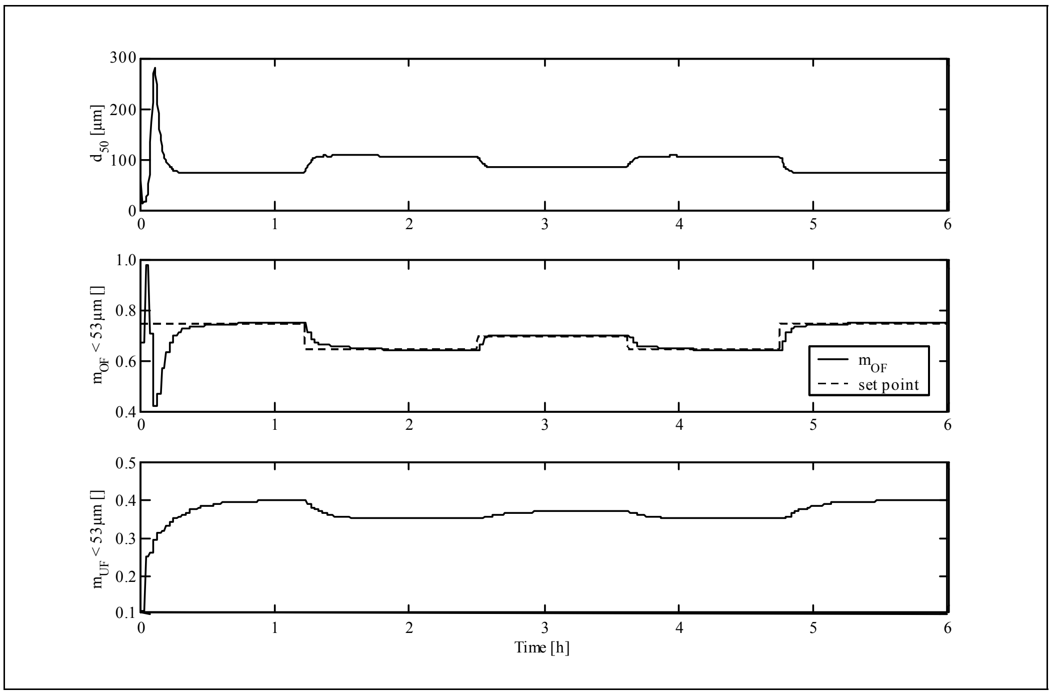


Figure 5-7 - Set point change responses of the hydrocyclone process variables without the presence of disturbances to the grinding circuit (learning environment).

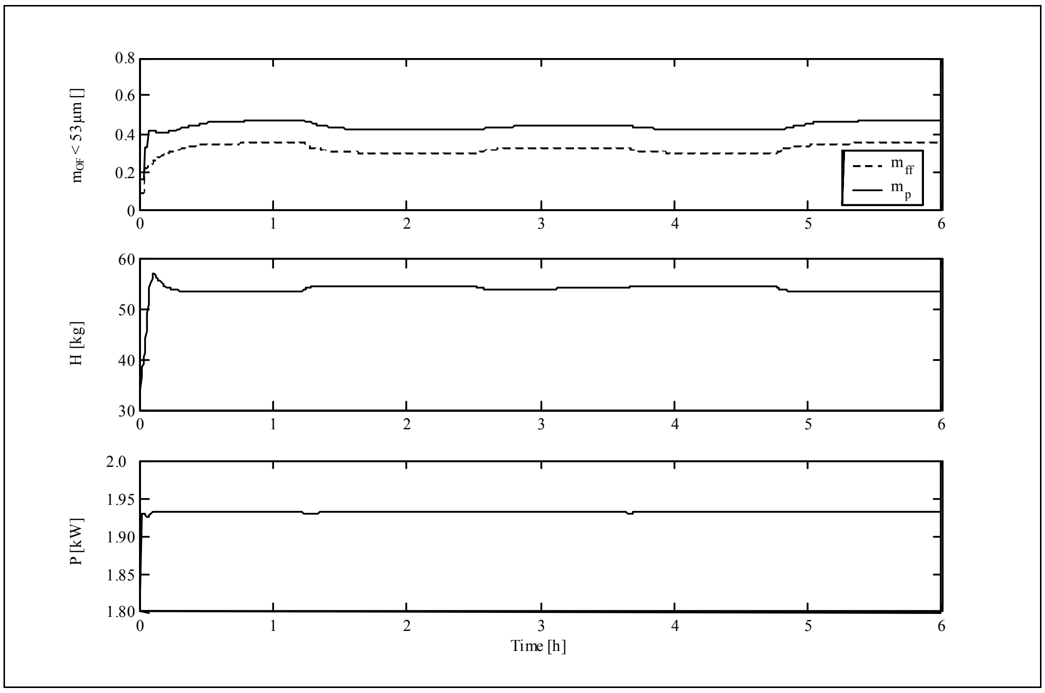


Figure 5-8 - Set point change responses of the ball mill process variables without the presence of disturbances to the grinding circuit (learning environment).

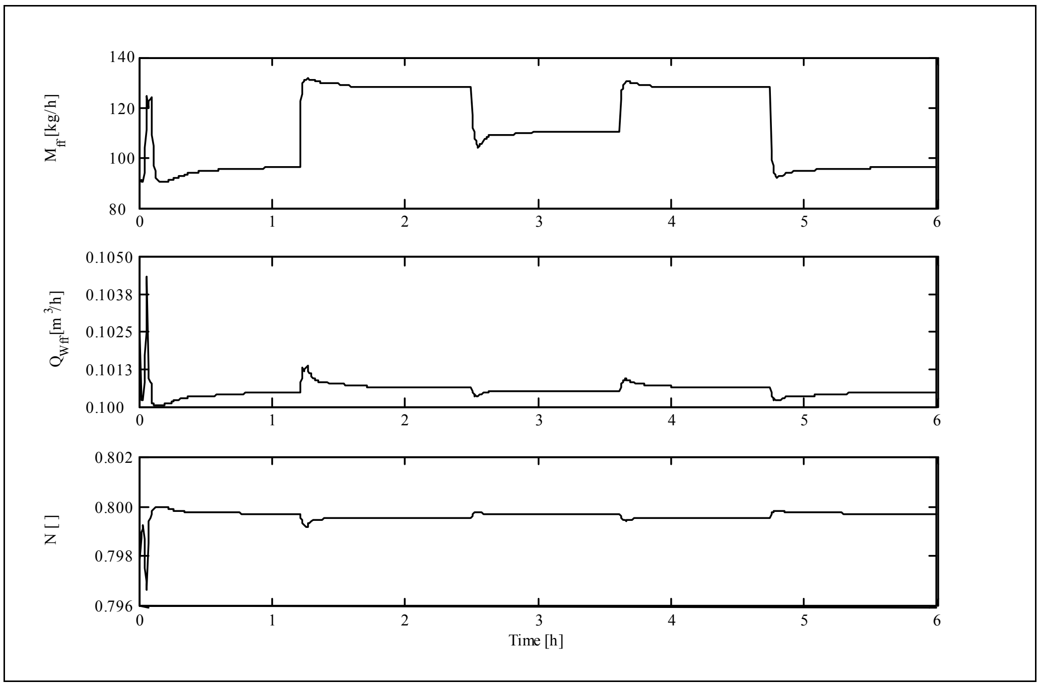


Figure 5-9 - Ball mill manipulated variable control actions based on the observed inputs to the neurocontroller (learning environment).

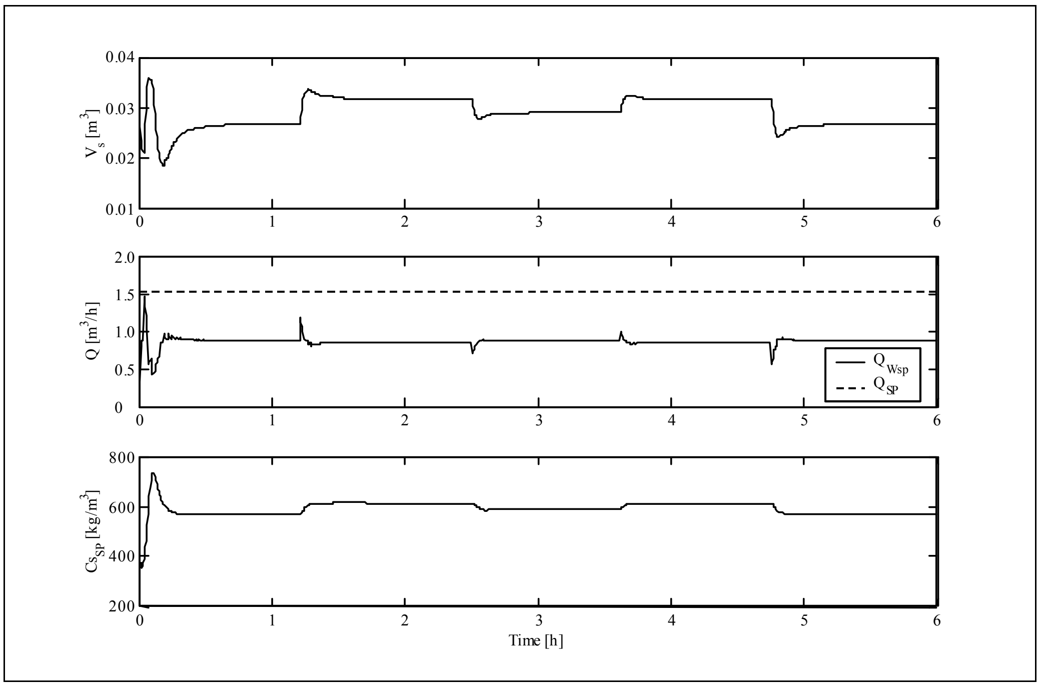


Figure 5-10 - Set point change responses of the sump process variables and the associated control actions (learning environment).

5.4.2 Learned behaviour - set point changes

Figure 5-7 to Figure 5-10 illustrate the learned behaviour acquired by the developed neurocontroller. With the reinforcement learning methodology (section 3.2), learning occurs without explicitly showing the neurocontroller *how* the task is to be performed. Instead, the reinforcement learning framework is concerned with *what* is to be accomplished. The learning algorithm (SMNE) explores the simulated environment and implicitly discovers *how* the task should be performed. No prior analysis of the simulated environment, for example possible controller pairings or likely control strategies, was provided to SMNE. The learned behaviour is purely an indication of what could be gauged from cause-effect interactions with the simulated environment. For example, consider the step change in set point from $m_{OF,\,53\,\mu m} = 0.65$ [-] to $m_{OF,\,53\,\mu m} = 0.75$ [-] between 4 - 6 [h]. As may be seen in Figure 5-7b, the set point change resulted in an overdamped response in the controlled variable with negligible steady state offset. As the SMNE design methodology does not impose a specific dynamic response in the controlled variable, the overdamped response reflects the dynamic response most suited to gaining the maximum reward from the dynamic environment. The controlled variable's dynamic response is thus implicitly chosen based on the system dynamics.

To establish this change in set point, the neurocontroller made a number of changes to the available manipulated variables. The mill solids fresh feed (M_{ff}), as illustrated in Figure 5-9a, was reduced significantly to allow for finer grinding of the mill charge by increasing the circulation load (ratio of mill feed rate to the fresh feed rate). The fresh feed dilution (Q_{Wff}) was not changed significantly (Figure 5-9b), which had the effect of reducing the solids concentration in the mill feed. This had the added effect of reducing the total mill hold-up (H) as seen in Figure 5-8b. With regard to equation 5-5 this resulted in the selection function, S_i, increasing due to a reduction in the mill hold-up, which increased the breakage rate. The mill speed could also be used to regulate the mill power and thus increase the selection function (equation 5-5). As the fraction of the mill critical speed (N) had been included as a possible manipulated variable, notably N was not considered by the neurocontroller for the mill control (Figure 5-9c). The value of N was primarily maintained at the maximum allowable value of 0.8 [-] (Table 5-3). Although the manipulated variable, N, is promising for reducing SISO controller interaction, the requirement of maintaining maximum mill throughput negates the effective use of N. The acquisition of variable speed drive inverters is thus unnecessary for this control strategy. Assumed that the increase in S_i is desirable, SMNE found that the reduction in H was more beneficial than an increase in mill power draw (Figure 5-8c), P, to the overall control strategy. The higher circulation load to the mill naturally allowed for finer grinding of the mill content as seen by the upward step change in $m_{p,53\,\mu m}$ in Figure 5-7a.

For the set point change between 4 – 6 [h], the sump volume, V_s, reduced significantly (Figure 5-10a). Sump level control (i.e., sump volume) is frequently regarded as a separate entity to the circuit control strategy. This generally implies that the sump level is maintained at a constant level throughout the grinding circuit operation. As the sump level control has been included in the more plant wide approach of SMNE, this control action is of interest. In Figure 5-10b it is evident that the sump discharge, Q_{SP}, and dilution water feed rates, Q_{Wsp}, were maintained at a more or less constant setting. The reduction in sump volume was thus as a result of the reduced mill throughput that accommodated the product specification set point change. Allowing the sump level to float within its lower and upper limits has favourable implications for control of the hydrocyclone. The constant Q_{SP} in Figure 5-10b means that the impact of hydrocyclone feed rate on classification, as dictated by equation 5-14, has been eliminated from consideration. The hydrocyclone classification control was thus limited to manipulating the solids volume fraction (f_v in equation 5-14) in the hydrocyclone feed rate. The reduction in $C_{s,SP}$ in Figure 5-10c effectively resulted in the classification becoming finer and had the desired effect of reducing d_{50} in Figure 5-7a. A larger portion of the fine hydrocyclone feed was thus also classified to the underflow to accommodate the set point change (Figure 5-7c).

The SISO controller interaction discussed in section 5.1 is thus not observed. A positive set point change in the product specification caused no interaction between the solids fresh feed rate and the sump discharge rate. SMNE has thus allowed for highly effective implicit elimination of controller interaction as a result of a more plant wide control approach.

5.4.3 Feed particle size and ore hardness disturbances

It is significant that the learned behaviour, illustrated in the previous section, is the neurocontroller's response to the simulated environment without the presence of sensor noise or disturbances of any kind. However, real world grinding circuits are plagued by disturbances. The neurocontroller's ability to effectively generalise its learned behaviour in the presence of disturbances, viz. feed particle size changes and ore hardness changes, is a measure of controller autonomy in dealing with uncertainty in its operating environment.

Figure 5-11 to Figure 5-14 illustrate the neurocontroller's ability to generalise in the presence of feed particle size disturbances. This disturbance is introduced as a twofold increase in the mass fractions of the four largest size intervals in the fresh mill feed (Table 5-4). The particle size disturbance is consequently changed randomly every 12 [min]. This disturbance frequency is deemed sufficient to allow for the complete development of transient responses. The impact on set point changes is illustrated in

Figure 5-11b. The process variable tracked the set point reasonably well considering the large disturbance. Robust performance is maintained despite needing to simultaneously make set point changes and deal with the disturbance in the feed. The solids feed rate, as with no disturbances, appears to be the primary manipulated variable for maintaining effective controller performance (Figure 5-13a).

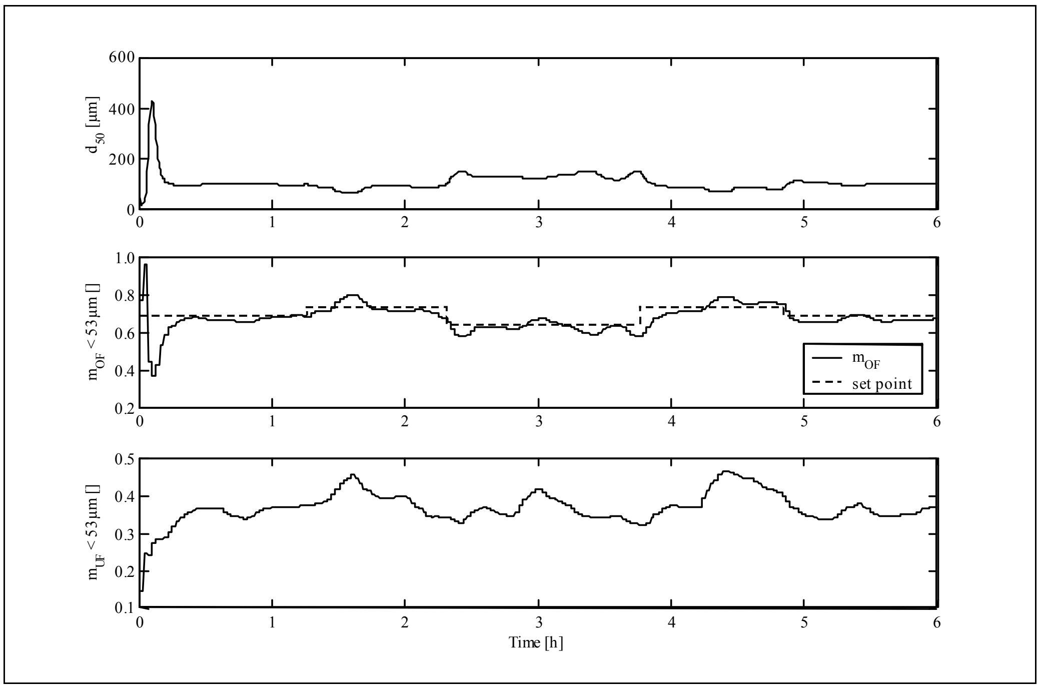


Figure 5-11 - Set point change responses of the hydrocyclone process variables in the presence of particle size disturbances in the fresh solids feed.

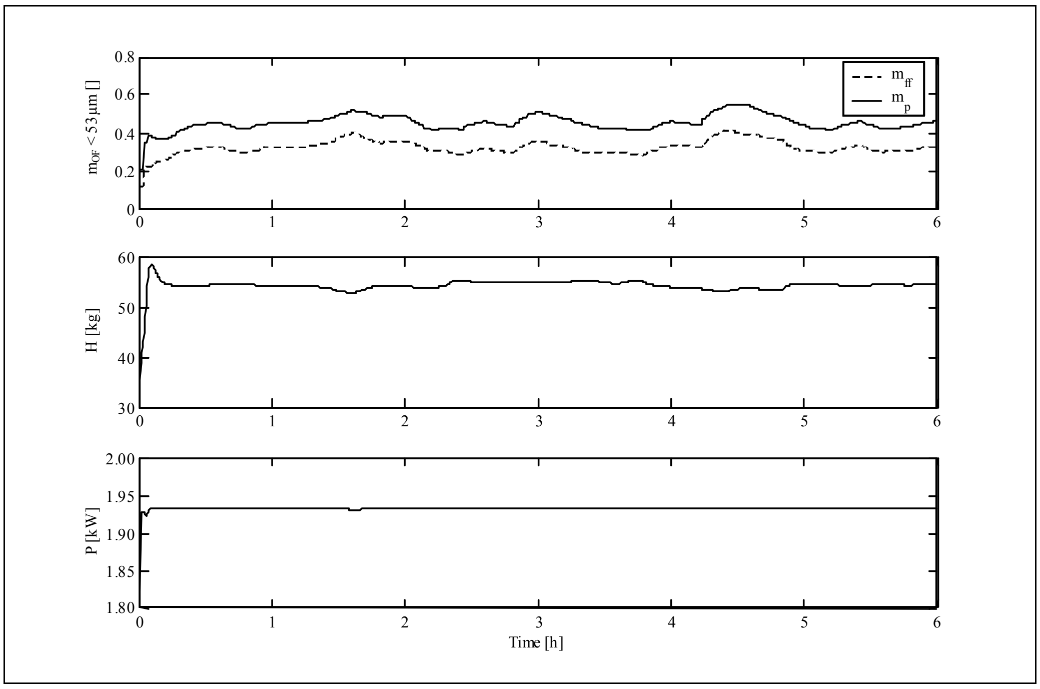


Figure 5-12 - Set point change responses of the ball mill process variables in the presence of particle size disturbances in the fresh solids feed.

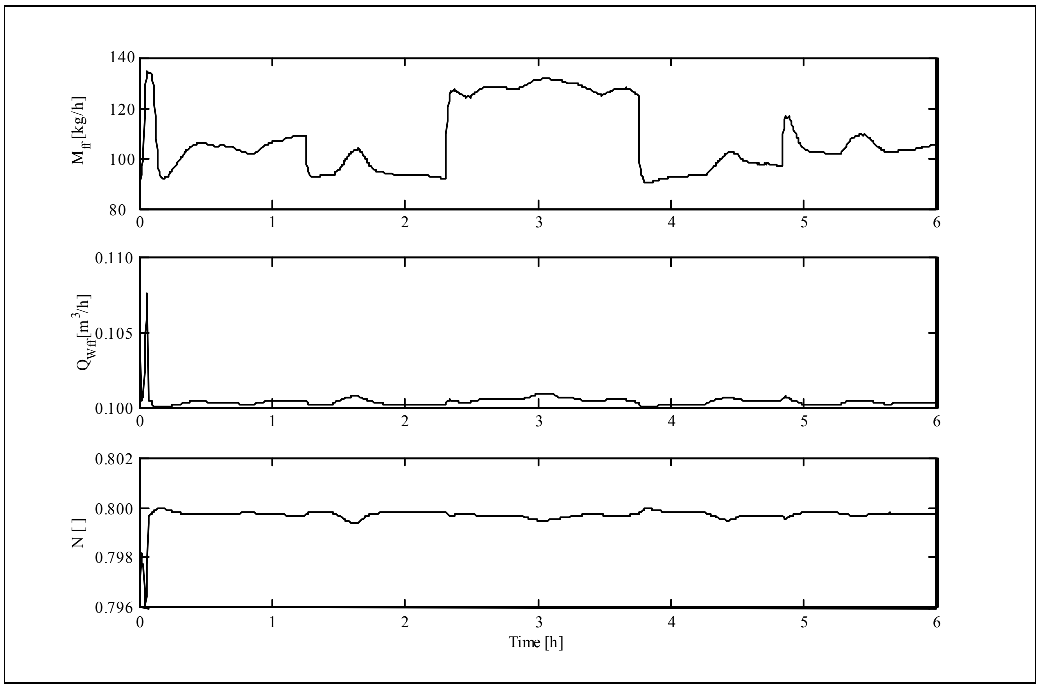


Figure 5-13 - Ball mill manipulated variable control actions based on the observed inputs to the neurocontroller (with particle size disturbances in fresh solids feed).

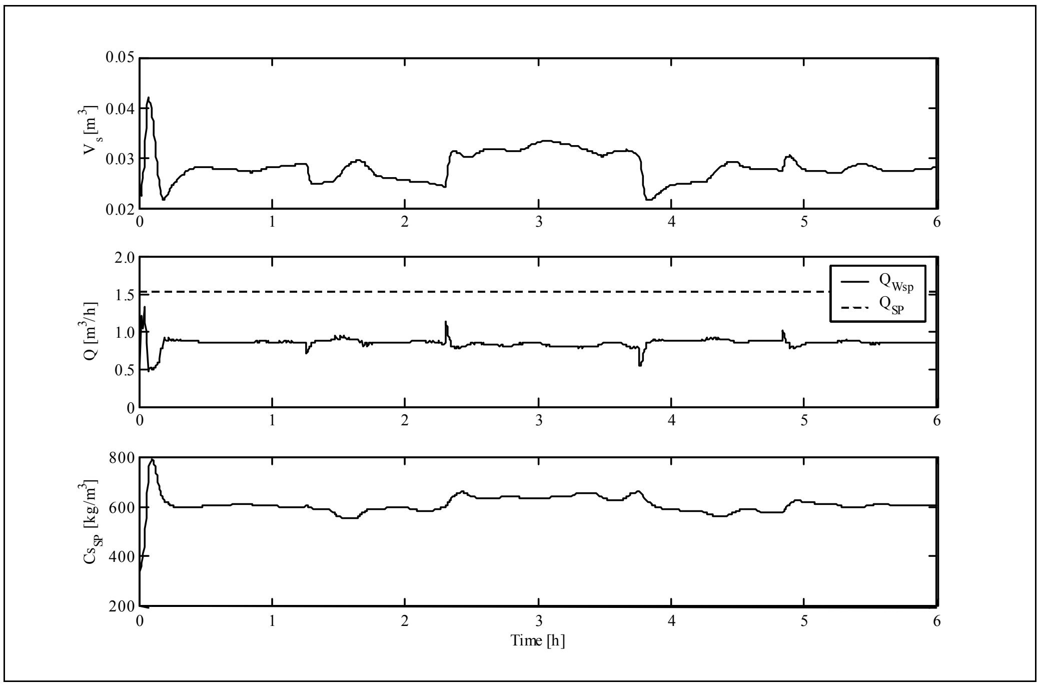


Figure 5-14 - Set point change responses of the sump process variables and the associated control actions (with particle size disturbances in fresh solids feed).

Another common grinding circuit disturbance is changes in ore hardness. The neurocontroller's response to this disturbance is illustrated in Figure 5-15 to Figure 5-18. The ore hardness disturbance is modelled by changing the nominal selection function (equation 5-5), S_i, randomly by a percentage at 12 [min] intervals. The change in hardness is indicated as a percentage change from the nominal value, as illustrated in Figure 5-16d. Despite this being a significant change in ore hardness over unrealistically short periods of time, the performance of the neurocontroller in dealing with this disturbance is extremely satisfactory (Figure 5-15b).

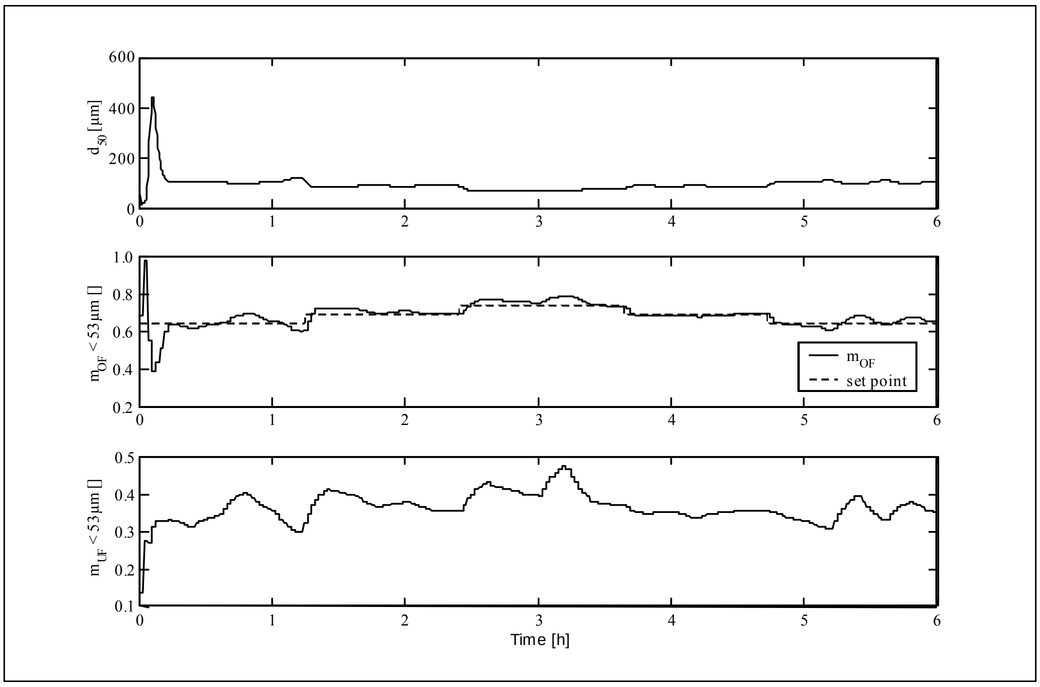


Figure 5-15 - Set point change responses of the hydrocyclone process variables in the presence of ore hardness disturbances.

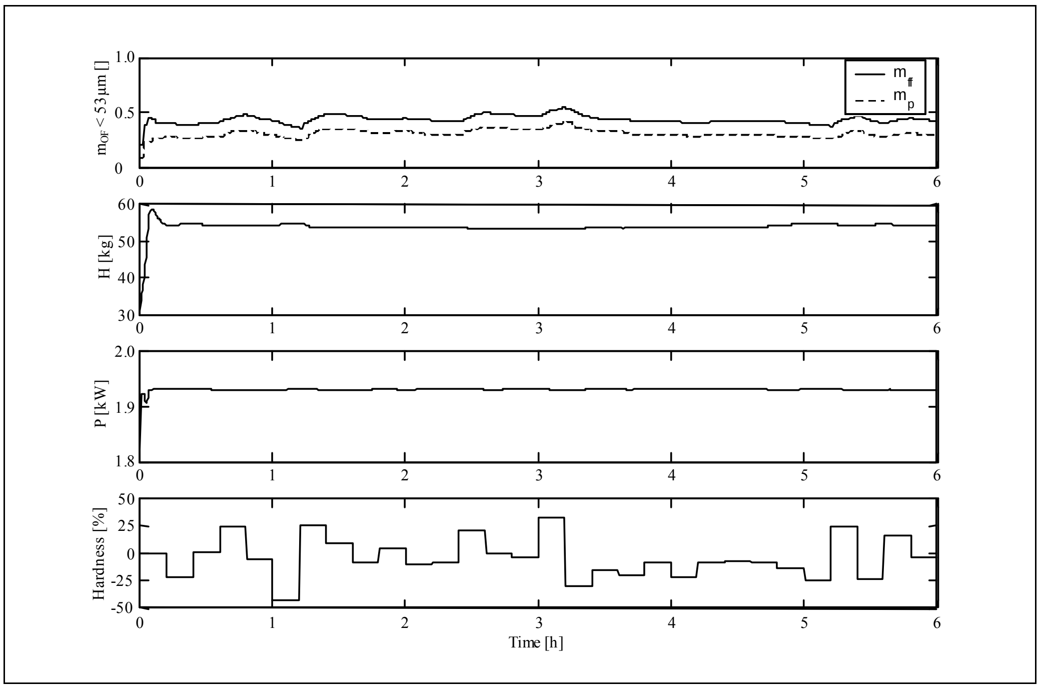


Figure 5-16 - Set point change responses of the ball mill process variables in the presence of ore hardness disturbances.

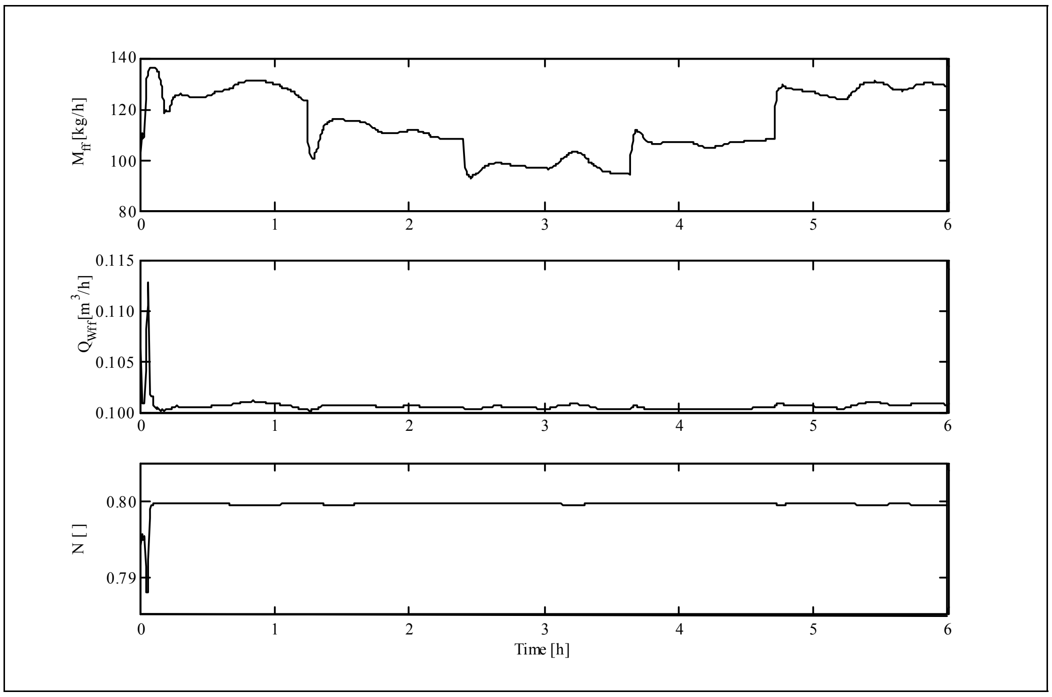


Figure 5-17 - Ball mill manipulated variable control actions based on the observed inputs to the neurocontroller (with particle size disturbances in fresh solids feed).

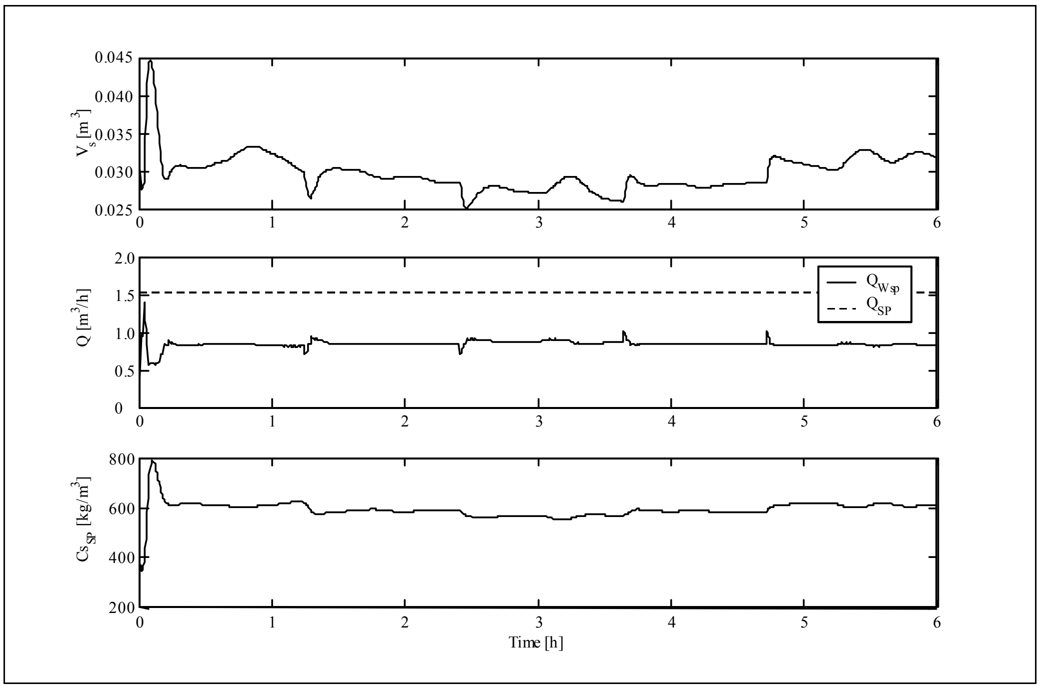


Figure 5-18 - Set point change responses of the sump process variables and the associated control actions (with ore hardness disturbances).

5.5 CONCLUSIONS

Evolutionary reinforcement learning offers significant opportunities for developing non-linear controllers (neurocontrollers) for mineral processing plants. The grinding circuit has a high dimensionality (i.e., numerous state variables). SMNE identifies and maps the dominant state variables to the input node patterns (i.e., process variables) during learning from cause-effect relationships. Cause-effect dynamic learning effectively reduced the dimensionality of the control problem. The SMNE algorithm was able to implicitly learn to eliminate controller interactions owing to cause-effect learning. The robust and highly autonomous control provided by the neurocontroller was demonstrated for both particle size disturbances in the fresh mill feed and for ore hardness variations. This effective neurocontroller behaviour is an indication of the ability of the SMNE algorithm to impart a beneficial degree of generalisation to the neurocontroller during the learning process, which allows for superior control in the face of process uncertainty (unmeasured disturbances). A more plant-wide approach to controller design based on the complete non-linear process model is achievable through the use of evolutionary reinforcement learning.

5.6 SYMBOLS

Symbol	*Description*	*Unit*
a_t	control action at time, t	[-]
bij	discrete breakage function	[-]
B_{ij}	cumulative breakage function	[-]
Cs	solids concentration	[kg·m^{-3}]
d	process disturbances	[-]
d_{50}	cut size for hydrocyclone	[µm]
d_i	mesh opening with size interval 2	[µm]
D	mill diameter	[ft]
E_i	fraction of solids in *i* reporting to underflow	[-]
f_v	volume fraction solids in cyclone slurry feed	[-]
H	total particulate mass hold-up in the mill	[kg]
m_i	mass fraction of material in size interval i	[-]
M	solids mass flow rate	[kg·h^{-1}]
M_b	fraction of mill loaded with balls	[-]
M_{balls}	mass of balls in mill	[short tons]
N	fraction mill critical speed	[-]
P	net mill power draw	[kW]
Q	volumetric flow rate	[m^3·h^{-1}]
r_t	reward at time, t	[-]

R_f	short circuiting of fines to cyclone underflow	[-]
s_t	process state at time, t	[-]
S_i	selection function	[h^{-1}]
S_i^E	specific selection function	[$t\cdot kWh^{-1}$]
t	time	[h]
u	manipulated variable control actions	[-]
V	vessel volume	[m^3]
WF	water flow rate in cyclone feed stream	[$m^3\cdot h^{-1}$]
WOF	water flow rate in cyclone overflow	[$m^3\cdot h^{-1}$]
y_p	process variables	[-]
Y_i	correction for entrainment	[-]
z^{-1}	Z-transform (time delay)	[-]

Subscript

Symbol	*Description*
i	size interval
n	size interval -37µm
FF	solids fresh feed
M	mill
MF	mill feed
MP	mill product
OF	cyclone overflow
S	sump
SF	sump dilution stream
SP	sump discharge stream
UF	cyclone underflow
Wff	fresh feed dilution water
Wsp	sump dilution water

6 DYNAMIC MODELLING

OBJECTIVES OF CHAPTER 6

- Describe an empirical rapid model development approach, singular spectrum analysis (SSA).
- Illustrate the SSA methodology for modelling the dynamics of an industrial lysine bioreactor from historical plant input-output data.

6.1 DYNAMIC MODELLING APPROACHES

The availability of a rigorous fundamental dynamic model, as in chapter 5, is rare. Although a phenomenological structure exists for parameter estimation via data reconciliation methods, an inappropriate model structure may make such a fit impractical.

In industrial practice it is seldom technically or economically feasible to construct detailed first principles models. The industrial success of MPC is largely attributed to the availability of commercial software for developing linear dynamic models from plant tests. No established method exists for identifying empirical non-linear models, due to the complexity of non-linear systems. Consequently, non-linear models are frequently based on fundamental models, derived from energy and material conservation laws (Henson, 1998; Morari and Lee, 1999).

In system identification, the prime concern is to find a suitable model structure wherein a good model fit may be obtained. Prior knowledge of physical relationships should be used in the model structure. Identifying concrete model relationships makes the model development task complex. Wherever possible, the problem should be reduced to parameter estimation, thereby making the system identification task less rigorous. Three model types exist (1) white-box models, (2) grey-box models and (3) black-box models. White-box models include fundamental models, where knowledge and physical insight describes the process behaviour perfectly. Grey-box models are sub-divided into (a) physical modelling and (b) semi-physical modelling. Grey-box physical modelling pertains to cases where a model structure is known, but certain parameters need to be estimated from experimental data. Grey-box semi-physical modelling suggests that certain measured input signals are used for constructing unstructured models such as neural networks. Black-box modelling implies that no prior physical insight is available and is typically grounded in unstructured regression models (Sjöberg et al., 1995).

6.1.1 Fundamental Models

Fundamental dynamic models derive from transient material, energy and momentum balances that are relevant to the process. Without spatial variations, the general form of non-linear phenomenological models is:

$$\dot{\bar{x}} = f(\bar{x},\bar{u}) \qquad \textbf{(6-1)}$$

$$0 = g(\bar{x},\bar{u}) \qquad \textbf{(6-2)}$$

$$\bar{y} = h(\bar{x},\bar{u}) \qquad \textbf{(6-3)}$$

where the ordinary differential equations (*f*), the algebraic equations (*g*) and the measurement equations (*h*) are functions of the n-dimensional state variable vector $\bar{x}$ and the m-dimensional manipulated variable vector $\bar{u}$. $\bar{y}$ is the a p-dimensional observability vector dictated by the available sensor measurements. Fundamental models have several advantages over non-linear empirical models. The constrained structure and limited parameters require less process data for development. Particularly, model parameters may be estimated from laboratory experiments as opposed to time-consuming plant experiments needed for empirical modelling. Also, fundamental models extrapolate well to operating regions not represented in the data set used for model parameterisation, provided the underlying assumptions remain valid. Effective extrapolation is important where processes have a wide operating region. Commercial dynamic simulators, such as AspenPlus (from Aspen Technologies) and Hysys (Hyprotech), provide for rapid development of rigorous non-linear dynamic models. However, these fundamental models must be validated with plant or laboratory data that reflect typical operating conditions. On-line validation involves placing the model in predictive mode in parallel with the process, whereupon large deviations between plant measurements and the model predictions may necessitate further modelling effort (Henson, 1998).

6.1.2 Empirical models

Numerous processes elude fundamental modelling, owing to a lack of process knowledge and suitable fundamental equation frameworks. Although fundamental modelling is conceivable for a large number of chemical and mineral processing operations, mechanistic model-based approaches may prove unreliable in their predictions. Many micro-phenomena that occur in complex processes (e.g., flotation) are poorly understood and identifying the large number of parameters within a reasonable fundamental framework may require a vast number of dedicated experiments (Amirthalingam & Lee, 1997). Empirical non-linear models must consequently be developed from dynamic plant input-output data. Discrete-time non-linear models include: (1) Hammerstein and Weiner models, (2) Volterra models, (3) polynomial ARMAX models and (4) artificial neural network models. Such non-linear input-output models have the general form:

$$y(k) = F\lfloor y(k-1),\ldots,y(k-n_y),u(k-1),\ldots,u(k-n_u),e(k),\ldots,e(k-n_e+1)\rfloor \qquad \textbf{(6-4)}$$

where F is a non-linear mapping, k the sample index, y the measured variable, u the manipulated variable and e the noise input. The number of past samples used is denoted by n. State space representations of the input-output data are also feasible. Typically multi-variable processes are modelled by a multi-input, single-output model for each measured variable (Henson & Seborg, 1995).

Non-linear identification entails the task listed in Table 6-1.

Table 6-1 - Non-linear system identification (Henson, 1998).

1. *Input sequence design - selection of appropriate measured and manipulated variables that impact y(k).*
2. *Structure selection - selection of input parameters n_y, n_u and n_e.*
3. *Noise modelling - estimation of the noise input e(k).*
4. *Parameter estimation - estimation of the model (i.e., mapping) parameters.*
5. *Model validation - prediction comparisons with plant data not used during training.*

Most of the tasks listed in Table 6-1 remain open-ended problems. The relative suitability of either Hammerstein, Weiner, ARMAX and neural network models for a given problem is not formally defined. An important issue is the design of plant tests that provide sufficient excitation for determining the dynamics without impacting production. The optimal number of delayed inputs n_y, n_u and n_e are difficult to determine, requiring complex techniques such as false nearest neighbours. A promising method for non-linear multi-variable identification is to determine state representations through appropriate projection (i.e., linear or non-linear) of input-output time-series data. Also, most practical problems require multi-input multi-output dynamic models. Combining MISO process models into a multi-variable model may not be effective. Non-linear models will inherently contain modelling errors in their process approximation, which need to be quantified in the controller design and analysis to ensure robust control (Morari and Lee, 1999).

Empirical non-linear models offer advantages over fundamental models, in that empirical dynamic models do not require detailed process understanding. Artificial neural networks are the most popular framework (Henson, 1998).

6.1.3 Hybrid models

Hybrid non-linear models combine the fundamental and empirical approach, which allows exploitation of the advantages of both methods. Typically, hybrid models use empirical models to estimate unknown functions in fundamental models, such as reaction rates in a fundamental reactor model. Hybrid models constrain the underlying physics within a fundamental framework, while estimating complex sub-systems using empirical approaches (Henson, 1998).

6.1.4 Batch modelling

Though fundamental, empirical and hybrid models have found application in control applications for continuous processes, few applications of model-based control to batch processing exist (Morari & Lee, 1999). Control objectives in batch processes are most frequently posed as tracking problems for time-varying reference trajectories over a defined batch time. During a batch, the process variables traverse wide regions of the state space that are characterised by varying degrees of non-linearity. The batch trajectory may only be in a particular region of the state space for a limited time period, limiting the representative data that may be collected for modelling the regional process dynamics. For effective modelling, a representative data set larger than for continuous process modelling is typically required. Favourably, batch processing is repetitive and hence process information may be exploited in a framework that allows the use of past batch data along with real-time data in the control system. Previous batches may be incorporated through state estimation within the predictive control computation. Run-to-run learning is fundamental to batch optimisation and control (Morari & Lee, 1999).

Developing an accurate batch model is thus considerably more difficult than for continuous operation. Owing to the high probability of an inaccurate dynamic model, a model-based control system is likely to have significant tracking error. The modelling exercise needs to include additional identification rigour and the modelling technique must maximise process knowledge from limited process information.

6.2 COMPLEXITY OF NON-LINEAR MODEL DEVELOPMENT

Modelling the complexities of non-linear processes from first principles often necessitates a large time investment that escalates costs. Modelling costs typically account for over 75 [%] of the expenditure incurred in an implementation of advanced control. Reaching tractable solutions to the modelling effort frequently requires relaxing the modelling rigour by introducing simplifying assumptions. A first principles model may not only be costly but also subject to inaccuracies. Poor model accuracy, owing to highly non-linear dynamics, may limit implementation to a narrow operating range. Without sufficient process models, advanced control methodologies have limited scope for successful implementation. Even with accurate process models, conventional advanced control strategies prove complex to implement (Hussain, 1999). Although the percentage of process units that require advanced control is small, these key units may account for up to 40 [%] of the gross revenue regenerated in USA process industries (Harris & Palazoglu, 1998). A technique that allows generality of the model structure (i.e., thereby facilitating rapid and less costly development) and high accuracy in expressing the process non-linearities is highly sought (Willis et al., 1992).

Research on non-linear system identification has focused on SISO systems, while most control challenges of practical importance are MIMO systems. In most cases, model regression uses SISO and MISO methodologies. Since the models are fitted separately, these methods prove less useful for MIMO control system development. Correlation among different outputs are not captured or exploited in the model identification process. A MIMO identification algorithm fits a single model with all the outputs simultaneously, normally in the form of a combined stochastic/ deterministic system. Thereby, correlation between outputs is incorporated and typically leads to improved identification of the deterministic part of the model (Morari & Lee, 1999).

Intelligent process control has been a reaction to the inability of conventional non-linear control to meet the needs of industrial control challenges (Stephanopoulos and Han, 1996). As neural networks approximate the complex dynamic behaviour of non-linear systems effectively, the use of neural networks in non-linear model identification for non-linear control becomes attractive. As neural networks learn by example, the development of useful process models from plant input-output data, becomes cost effective. Several neural network control strategies have been applied for a wide range of control objectives on-line (Hussain, 1999). However, no established method exists for constructing a non-linear model from plant test data (Morari & Lee, 1999).

The most extensive use of neural networks in control applications has been in a model predictive control (MPC) guise. Non-linear MPC entails a non-linear optimiser, which uses an on-line process model to determine future control actions. The process model is included in the control loop as in Figure 6-1. The non-linear optimiser uses the process model to calculate a number of optimal control actions over a prediction horizon. This involves calculating a control action vector, $\overline{\upsilon}$, which optimises a cost function containing the process vector, $\overline{\psi^m}$ with reference to the desired process state, ψ^{SP}. The first predicted control action in the prediction horizon $\upsilon(t+1)$ is applied to the plant, which changes the process state to $\psi^p(t+1)$. MPC uses an error signal, *e*, to improve the process model by reducing the plant/model mismatch. NMPC's success is highly dependent on the quality of the process model. Both experimental-scale and industrial implementations have used neural network models in MPC control loops. These include control of an industrial polypropylene reactor, an industrial multi-pass packed bed reactor, a laboratory scale distillation column and temperature control of a 175 [kW] experimental furnace system (Hussain, 1999). These control applications are single-input single-output (SISO) control systems.

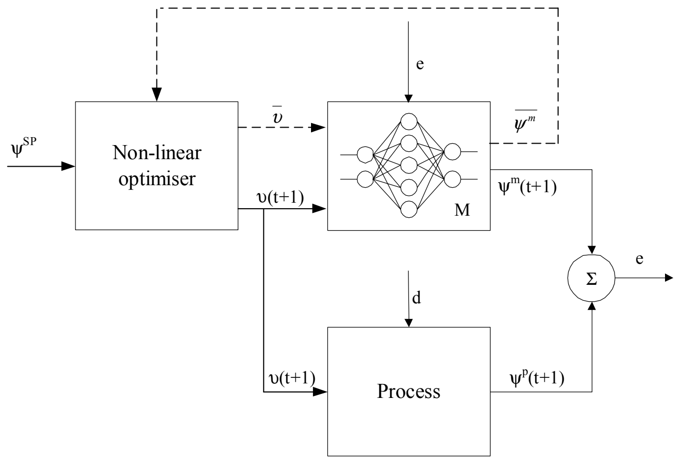


Figure 6-1. Control structure for non-linear model predictive control.

A simpler alternative to NMPC is direct inverse control (DIC). DIC inverts a neural network model, which acts directly as the controller (Figure 6-2). The controller inputs may include several current and past process variables (ψ^P), set points (ψ^{SP}) and past control actions (υ). Thereby, the controller determines the control action, $\upsilon(t+1)$, which along with disturbances *d* changes the process state to $\psi^P(t+1)$. Direct inverse controllers have been utilised in laboratory scale applications for single-input single-output (SISO) temperature control (Hussain, 1999).

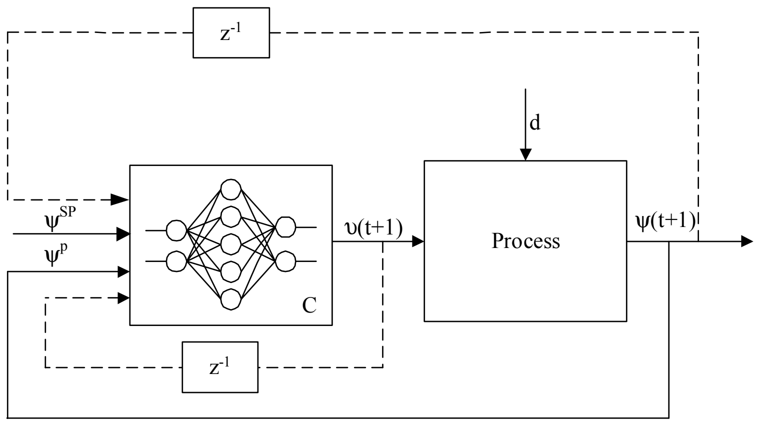


Figure 6-2. Control structure for direct inverse control.

This chapter demonstrates a non-linear identification methodology Singular Spectrum Analysis (SSA). SSA identifies the state variables that created the time series of a given process variable and constructs a dynamic neural network model using principal component analysis (PCA). SSA models may consequently be used by the SMNE algorithm for neurocontroller development. The following section describes the SSA methodology. Historical input-output data from an industrial fed-batch bioreactor demonstrates the SSA methodology as a case study.

6.3 RAPID MODEL DEVELOPMENT IN A NEURAL NETWORK PARADIGM

6.3.1 Semi-empirical dynamic modelling

Dynamic modelling entails determining those variables that describe a physical system's dynamic response and how these variables are interrelated. The ultimate goal of any dynamic modelling exercise is to accurately predict the process conditions of a real plant at some time in the future. Such models prove invaluable in the implementation of advanced process control strategies. As discussed in section 6.1, phenomenological modelling is concerned with models based on first principles, whilst empirical modelling uses input-output data for creating representations of a real system's dynamics. Modelling from first principles is desirable, as it generally provides wider applicability and greater understanding of the process. However, phenomenological modelling is far more time-consuming and therefore more expensive than empirical modelling that relies on warehoused plant data. Ideally, the gap between phenomenological and empirical modelling should be bridged. This may be accomplished by constructing pseudo-phenomenological models from input-output data.

A promising approach for non-linear multivariable system identification involves defining states from input-output data through appropriate non-linear projection, thereby building a state-space model (Morari & Lee, 1999). A deterministic system may be described by differential equations comprised of state variables and the manipulated variables that impact on the state variables. The state variables, ξ_i, represent the minimum number of variables required to fully describe the dynamic system. The vector of state variables represents the phase space for the dynamic system. The dimension of the state space is equal to the number of differential equations. Manipulated variables, υ_i, act upon the dynamic system and determine the future values of the state variables. The dynamic system may be comprised of f ordinary differential equations with state vector $\overline{\xi}(t) = [\xi_1(t), \xi_2(t), \ldots, \xi_f(t)]$ and is assumed differentiable over the full range of operation, so that each state variable may be described by (Abarbanel et al., 1993):

$$\frac{d\xi_i(t)}{dt} = f(\overline{\xi}) + g(\overline{\upsilon}) \tag{6-5}$$

$$\psi_k(t) = h(\xi_k) \tag{6-6}$$

where f is a function of the state vector, $\overline{\xi}$, and g is a function of the manipulated variable vector, $\overline{\upsilon}$. The process variables, ψ_i, are related to the state variables by the function transformation h (Abarbanel et al., 1993). A continuous process typically has a single steady state attractor associated with the chosen operating point. Such steady state attractors may be either point attractors, chaotic (i.e., non-linear attractors) or periodic attractors with long oscillation periods. Consider the bioreactor flow sheet optimisation study presented in section 4.7. With no recycle or concentrated substrate feed at constant bioreactor level, a bifurcation analysis reveals that numerous open-loop steady states exist depending on the residence time. Figure 6-3 shows the bifurcation analysis, i.e. the possible open-loop steady state conditions for the optimised bioreactor flow sheet. At a Damkohler number of ±1.5 [-], the steady state attractor is open-loop stable. A stable attractor is also referred to as a point attractor as shown in Figure 6-4 for 3 state variables in phase space. When the Da = ± 1.0 [-], the steady state attractor exhibits multiplicity. Multiplicity implies that the phase space attractor nearest the initial condition determines the open loop steady state response. The attractor may consequently be either open-loop stable or unstable. An unstable or chaotic attractor is shown in Figure 6-5. As the Damkohler number is further reduced, the steady state open-loop response can no longer be stable, regardless of the initial condition, and unstable attractors and periodic attractors become the possible attractors depending on the initial condition.

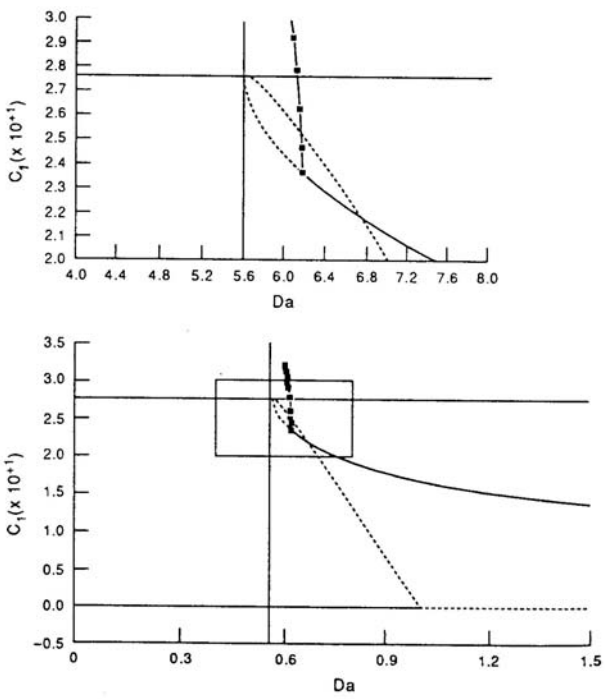


Figure 6-3 - Bifurcation analysis for the optimised bioreactor flow sheet. C$_1$ represents the dimensionless biomass concentration and Da the Damkohler number (i.e., dimensionless residence time). Solid lines represent stable open-loop steady state attractors. Dashed lines represent unstable or chaotic open-loop steady state attractors. Marked solid lines represent periodic attractors with long periods.

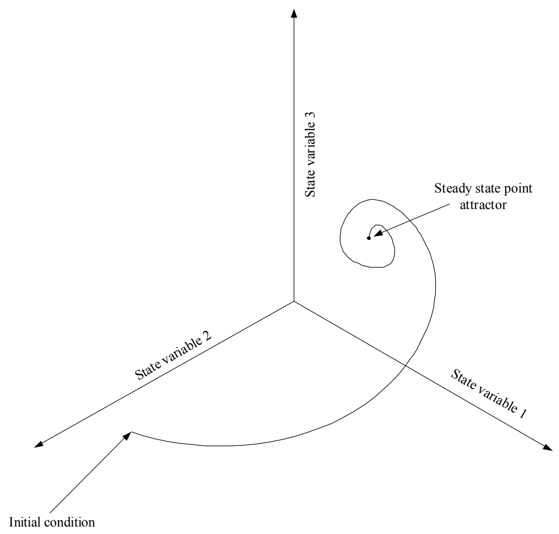


Figure 6-4 - Point attractor for a 3-dimensional state space (i.e., 3 state variables).

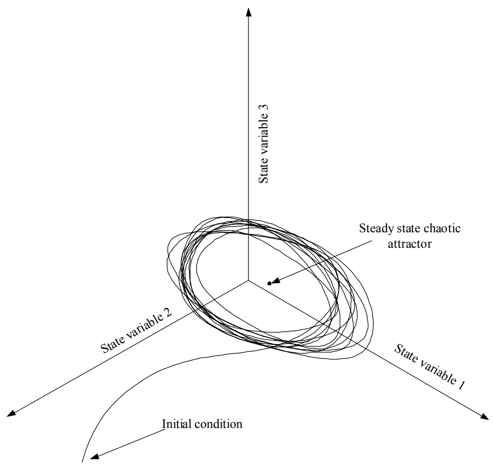


Figure 6-5 - Chaotic attractor for a 3-dimensional state space (i.e., 3 state variables).

Since the experimental pilot-plant case study in this work pertains to batch distillation (chapter 7), the differential equations for a unit volume of structured packing in a batch rectifier serve as example. For an arbitrary volume of column packing the governing mass balance, component balance and enthalpy balance yields (Distefano, 1968):

$$\frac{dH_j}{dt} = V_{j+1} + L_{j-1} - V_j - L_j \tag{6-7}$$

$$\frac{d\left(H_j x_{i,j}\right)}{dt} = V_{j+1} \cdot y_{i,j+1} + L_{j-1} \cdot x_{i,j-1} - V_j \cdot y_{i,j} - L_j \cdot x_{i,j} \tag{6-8}$$

$$\frac{d\left(H_j \cdot I_j\right)}{dt} = V_{j+1} \cdot J_{j+1} + L_{j-1} \cdot I_{j-1} - V_j \cdot J_j - L_j \cdot I_j \tag{6-9}$$

where H is the liquid hold-up, V is the vapour flow rate and L the liquid flow rate. The state variables x_{ij} and y_{ij} are the mass fractions of each component, i, at column location, j, in the liquid and vapour phases respectively. Since the mathematical model involves heat balances, liquid and vapour enthalpies are given by $I_j = f(x_j, T_j)$ and $J_j = f(y_j, T_j)$. Assuming constant volume hold-up, the hold-up mass per volume of structured packing may be described by the algebraic equation, $H_j = G_j \cdot \rho_j$ with the liquid density $\rho_j = f(x_j, T_j, P_j)$ and G_j the volume hold-up. The rate of interface mass transfer, with the vapour phase rate limiting, may be given by (Distefano, 1968):

$$\frac{dy_{i,j}}{dt} = k_y \cdot a \cdot S \cdot \left(y^* - y\right) \tag{6-10}$$

$$\frac{dx_{i,j}}{dt} = k_y \cdot a \cdot S \cdot \left(y - y^*\right) \tag{6-11}$$

where k_y is the overall mass transfer coefficient with $k_y = f(V, L,$ packing geometry), a is the interfacial area per unit volume of the structured packing and S is the cross-sectional area of the column. The vapour-liquid equilibrium relationship may be given by (Distefano, 1968):

$$y^*_{i,j} = K_{i,j} \cdot x_{i,j} \tag{6-12}$$

where $K_{i,j} = f(x_j, T_j, P_j)$ determined by an equation of state. Figure 6-6 illustrates the unit volume of structured packing with the governing variables.

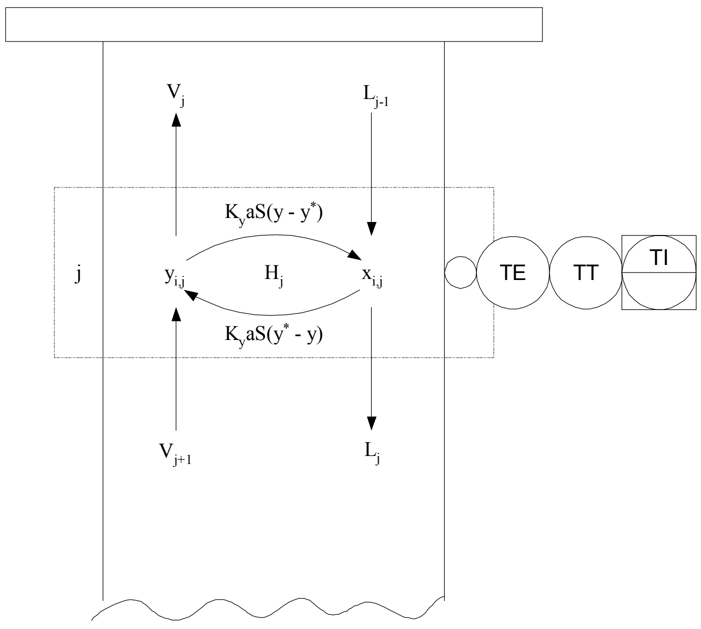


Figure 6-6. Control volume of a structured-packing distillation column near the top of the column.

For the batch distillation column section in Figure 6-6, the dominant state variables are the variables of the differential equations, that describe $x_{i,j}$ and $y_{i,j}$. These state variables are rarely measured in batch distillation, owing to the high cost of composition analysers. For a binary or pseudo-binary system, a temperature sensor serves as inferential indicator for composition using the temperature dependency of equation 6-12 (assuming constant pressure). Such a temperature measurement along the column is thus an example of a process variable. Assuming a constant heat duty, the manipulated variable for the column is the liquid reflux rate, *L*. When such a system is sampled discretely, the dynamic representation for each state variable becomes (Abarbanel et al., 1993):

$$\bar{\xi}(n) = \bar{\xi}(t_0 + n \cdot \tau_s) + \bar{\upsilon}(t_0 + n \cdot \tau_s) \quad \textbf{(6-13)}$$

$$\xi_k(n+1) = F\left[\bar{\xi}(n)\right] + G\left[\bar{\upsilon}(n)\right] \quad \textbf{(6-14)}$$

where, *n* is the number of samples taken since time t_0 with sample period τ_s. *F* and *G* relates the state variable at $\xi_k(n+1)$ to the state vector $\bar{\xi}(n)$ and the manipulated variable $\bar{\upsilon}(n)$ respectively. The relationship between the continuous and discrete system may be established by considering the approximate time derivative (Abarbanel, 1993):

$$\frac{d\xi_k}{dt} \approx \frac{\xi_k\left(t_0 + (n+1)\cdot\tau_s\right) - \xi_k\left(t_0 + n\cdot\tau_s\right)}{\tau_s} \qquad \textbf{(6-15)}$$

The state vector (i.e. $x_{i,j}$ and $y_{i,j}$) for a binary system at an arbitrary column location thus has 4 state variables. These four state variables pass through a four-dimension hyperplane (i.e., phase space) as the batch progresses. The process thus has a trajectory fixed in a coordinate system (i.e. phase space), which is dictated by the system dynamics. The path laid by such a trajectory in time is referred to as the phase portrait. Recall that a single temperature reading serves as inferential indicator of the four state variables. Sampled data is logged to a process database at a specified sample rate τ_s^{-1}. Large warehouses of historical data, comprised of process (input) and manipulated (output) variables, provide information regarding past dynamic responses. Ideally, historical data should be useful in constructing the phase portrait and consequently an accurate representation of the underlying differential equations. Using equation 6-15, these logged samples may be used to reconstruct the derivative equations that underlie the dynamics. To recreate the temperature differential equation, the derivative from a time series of the temperature process variable must be approximated using numerical differentiation. A regression model, containing input elements as in equation 6-14, may be constructed to predict these derivatives. Sensor noise may limit the application of such an approach. However, it proves unnecessary to approximate the derivates directly.

6.3.2 Extracting state variables from time series data

Takens' theorem states that a single time series holds knowledge of all the state variables that created the time series, i.e. the time series for the process variable T holds knowledge of the four state variables that determined T's value at any sample n. Knowledge of the state variables allows reconstruction of the phase portrait. A state space can be constructed from observed variables by using the method of delays (Ydstie, 1990; Narenda and Partharasarathy, 1990; Hernandez & Arkun, 1992). The use of a number of past (i.e., lagged) samples proves sufficient to predict the trajectory in the four dimensional space (i.e. the phase portrait), so that:

$$\bar{x}(n+1) = f\left[T(n), T(n-1), T(n-2) \ldots T\left(n - \frac{\tau_w}{\tau_L}\right)\right] \qquad \textbf{(6-16)}$$

Therefore, with sufficient knowledge of the system's past locations in the four dimensional space, the next location in the coordinate space may be predicted. Singular spectrum analysis (SSA) is a data-analysis tool introduced into non-linear dynamics by Broomhead and King (1986) for creating predictive models from time series information. As such, SSA is a rapid model development (RMD) tool. SSA uses a lagged embedding matrix (LEM) of the time series, which serves as the foundation from which the state variables are extracted. A number of time scale issues need to be resolved in constructing the LEM. A short sample time, τ_s, allows for accurate description of rapid dynamic variations, but creates highly correlated samples. High correlation between successive samples means that a large number of historical samples must be used to resolve slow dynamics. The LEM is determined by the lag time, τ_L, and the window length, τ_w. The lag time, $\tau_L = k\tau_s$, should ensure a degree of statistical independence between the samples. In this work a lag time $\tau_L = \tau_s$ is selected for the embedding. The window length, τ_w, is defined as the number of past samples needed to accurately reconstruct the phase portrait. Clearly, as τ_w is decreased, the measurable dynamic variation in the window becomes less. Increasing τ_w thus increases the amount of information contained in the window. The window length should be selected based on an estimate of the process time constant. Consider the time series for a temperature sensor on a batch distillation column *T*, with *N* data samples. The time series is consequently embedded based on the number of window length samples *M* as in Table 6-2 (Broomhead & King, 1986).

Table 6-2 - Lagged embedding matrix (Broomhead & King, 1986).

X_1	X_2	X_3			X_M
T(1)	T(2)	T(3)	.	.	T(M)
T(2)	T(3)	T(4)	.	.	T(M+1)
T(3)	T(4)	T(5)	.	.	T(M+2)
.	.	.	.	.	.
.	.	.	.	.	.
T(N-M)	T(N-M+1)	T(N-M+2)	.	.	T(N)

The rows of the LEM represent a time fragment (i.e., window) in which the dynamics of the process may emerge. The LEM columns represent an initial estimate of the state variables as determined by τ_L and τ_w. Thereby, each LEM row must map consistently to a unique location in phase space. State variables are by definition orthogonal to one another; as the state variables are represented by the axes in a phase portrait coordinate system. However, the column vectors in $\overline{X}$ are highly correlated, owing to a small shift by the lag time τ_L. In other words, choosing a restricted τ_w means that X_1 and X_M are too highly correlated and hold similar information. Choosing τ_w too large makes X_1 and X_M completely uncorrelated (i.e., independent) and the projection of an orbit on the attractor is onto two unrelated directions. A criterion for an intermediate choice is based on linear autocorrelation:

$$C_L = \frac{\frac{1}{M}\sum_{n=1}^{M}\left[T(n+\tau_L)-\overline{T}\right]\cdot\left[T(n)-\overline{T}\right]}{\frac{1}{M}\sum_{n=1}^{M}\left[T(n)-\overline{T}\right]^2} \quad \textbf{(6-17)}$$

$$\overline{T} = \frac{1}{N}\sum_{n=1}^{N}T(n) \quad \textbf{(6-18)}$$

where, C_L is the correlation matrix based on the embedding matrix (Table 6-2). The first minimum in the first row of the correlation matrix is typically a good guideline for τ_w. Should no minimum exist, τ_w is typically selected at a correlation of 0.2. A non-linear notion of independence, average mutual information (AMI), may also be used to determine τ_w, noting that AMI is sensitive to noise in the data. These guidelines should be cross-checked with the expected process time constant, since most of the dynamic information will be contained within the time frame of the process time constant. More often than not, τ_w is determined by trial-and-error (Kugiumtzis, 1996).

SSA relies on principal component analysis (PCA) of the LEM to extract the state variables from the time window in each row of the LEM. Classical PCA gives the principal axes of such a *M* x (*N-M)* matrix, $\overline{X}$, by rotating the coordinate axes to minimise the variance between the variables in an orthonormal fashion. First, the column vectors of $\overline{X}$ are normalised (i.e., centred) by subtracting each column's mean and dividing each column element by the standard deviation. Thereafter, a large set of correlated variables may be transformed into a small set of uncorrelated (i.e. orthogonal) variables by projecting the variables in $\overline{X}$ so that:

$$\overline{E} = \overline{X}\cdot\overline{PC} \quad \textbf{(6-19)}$$

where $\overline{PC}$ is the projection coefficients matrix or principal components and the matrix $\overline{E}$ is called the empirical orthogonal functions. Though *M* components are needed to reproduce the complete system variability, sufficient variability may be explained by using only *C* principal components. The dimensionality of the variable space is thus reduced. As the variables of the matrix $\overline{E}$ are orthogonal to one another, the variables are also state variable representations for the time series. Should *C* principal components explain more than typically 95 [%] of the variance, the first *C* column vectors in $\overline{E}$ are the state variables of the system. PCA thus establishes the true dimension of the phase space.

The remaining *(M-C)* column vectors in $\overline{E}$ contain minor system dynamics and noise. Discarding the coordinates containing insignificant variance, noise is implicitly filtered from the time series. This noise filtering property offers an opportunity to use closed loop data for model identification. Closed loop plant data, i.e. data generated during feedback control, is more readily available than open loop data. The fundamental problem with model identification from closed loop data is the correlation introduced by the noise in the process variable signal that is fed back into the control action signal to the plant (Forssell and Ljung, 1999). Projecting the noise out of the time series and changing the plant measurement to a state space representation decouples the noise element in the manipulated variable time series from the process variable representation. SSA is thus a robust means of identifying the open loop plant from closed loop data as in Figure 6-7.

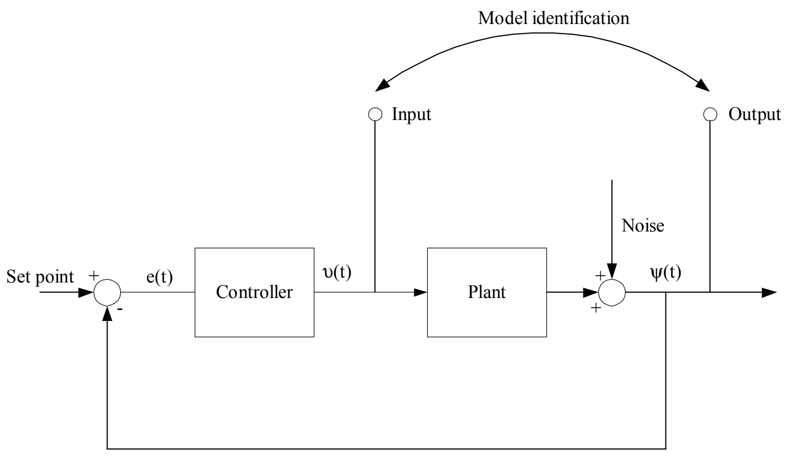


Figure 6-7 - Closed loop process model identification.

6.3.3 Trends and non-stationarities

During start-up, a continuous process may pass through several attractors as seen in Figure 6-8. Figure 6-8 shows the hypothetical open loop behaviour of a dynamic system with 3 state variables, which moves from the start-up (i.e., initial) condition and settles in the region of an open loop unstable steady state (e.g., the region of maximum economic return). Minimal plant data may be available for complete modelling of attractors only encountered during start-up. Nevertheless, a model predictive controller must stabilise the unstable attractors by robust feedback along the full start-up trajectory. The practical applications in this thesis deal primarily with batch processes, which are also inherently non-stationary. Modelling such large regions of the state space complicates SSA and the non-stationary operation tests the boundaries of the modelling technique.

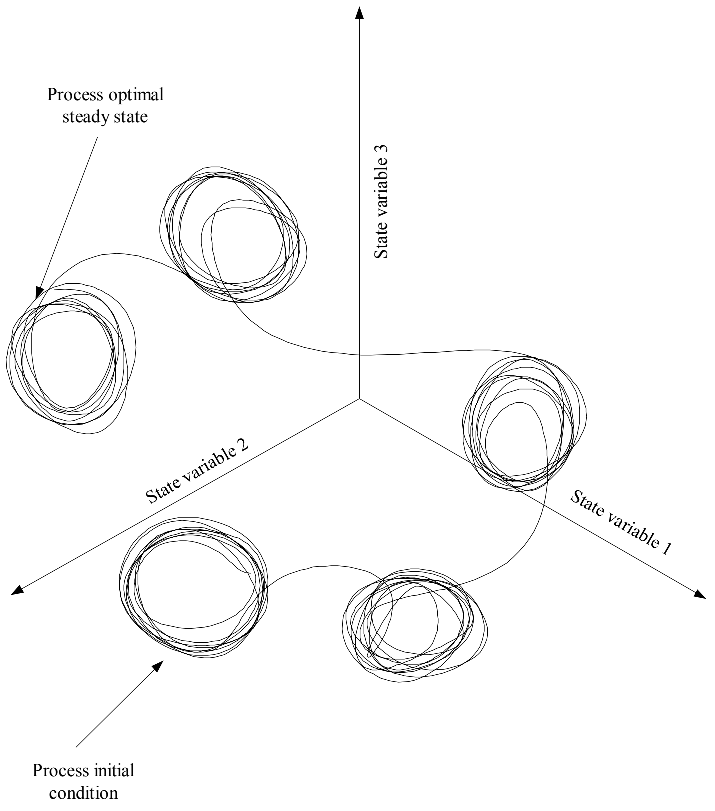


Figure 6-8 - Hypothetical chaotic attractors during state transitions or start-up.

When applying PCA, normalised data in multi-channel embeddings prevents biasing towards a time series that has a large scale in the selected engineering units. Long start-up periods for continuous processes and transitions from one steady state to another introduce a distinct trend into the time series data. A trend introduces significant variability in a particular direction in n-dimensional state space. PCA reduces this variability in the embedding matrix, by including the trend in the first few principal components. Note that any set or combination of *n* variables that completely determine the system dynamics may describe the state space. For the data in Figure 6-9, any rotation of the 3 dimensional Cartesian axes is a valid state space representation. For the non-stationary process in Figure 6-9, SSA selects a state representation with the non-stationary element as major state variable, i.e. the remaining variability of the system is reduced after aligning the n-dimensional Cartesian axes along the trend.

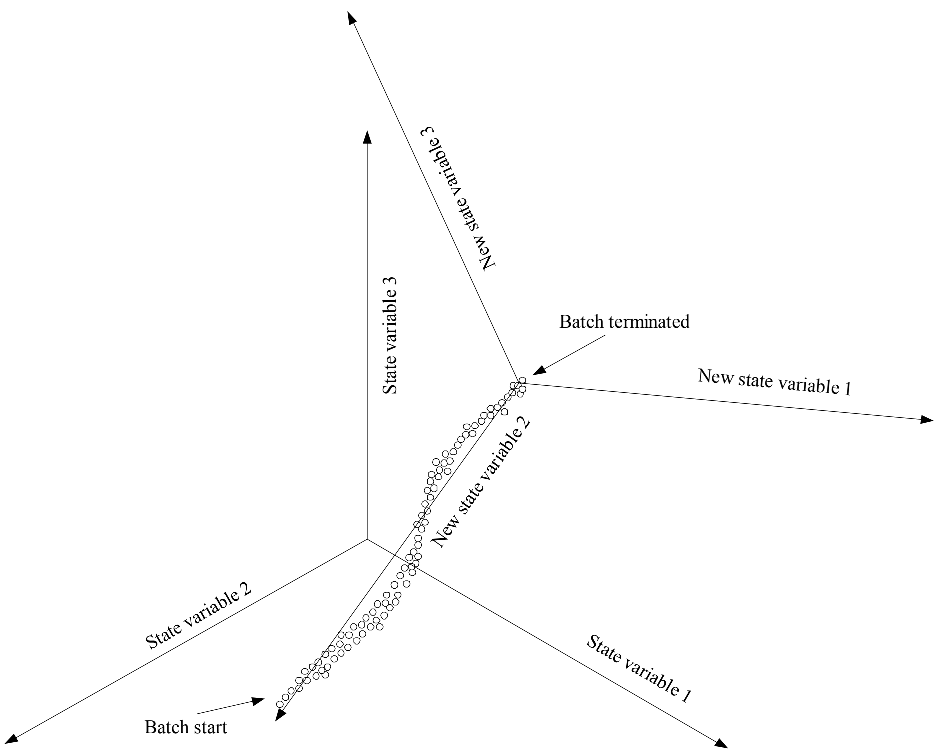


Figure 6-9 – Data for a hypothetical batch process within a true three dimensional state space. Phenomenological modelling may use the Cartesian axis determined by 'state variable 1', 'state variable 2' and 'state variable 3'. The trend in the 3-dimensional data biases SSA to a state space representation with 'New state variable 1', 'New state variable 2' and 'New state variable 3'.

For example, consider a batch distillation column with variable reflux and no product removal from the reflux drum. Assume that a batch terminates when a desired reboiler temperature is reached. The time series for the column's temperature has fast dynamics determined by the reflux rate, whilst the reboiler temperature is determined by the rate of accumulation in the reflux drum. As the batch progresses, the reboiler temperature increases. By embedding the column temperature and the reboiler temperature time series, the reboiler temperature introduces a trend in the embedding matrix. SSA selects a state space representation that includes the reboiler temperature as a major component of one of the state variables (compare Figure 6-9). Such a state space representation may obscure the importance of variability in the data that has fast dynamics. In Figure 6-9, 'State variable 1' and 'State variable 2' are combined into 'New state variable 1' as a result of the non-stationary trend of the batch. In practice, despite non-stationary dynamics, SSA still works reasonably well (Vautard et al., 1992).

6.3.4 Dynamic neural network modelling & model validity

Given the state variables that determine the time series and the manipulated variables that are highly correlated with the time series, a predictive model may be constructed. Primarily, in accordance with equation 6-20, an accurate predictor for the process variable T (i.e. ψ in equation 6-6) is sought, by finding the function:

$$T(t+1) = h\left[f(\overline{\xi}) + g(\overline{\upsilon})\right] \qquad \textbf{(6-20)}$$

Neural networks may approximate any non-linear continuous function to an arbitrary degree of accuracy. The calculated state variables at time t and the reflux rate sampled over a past window length τ_w form the inputs to the neural network. The reflux rates at [(t), (t-1), (t-2),...,(t - (τ_w / τ_s))] are incorporated as inputs to allow for time delay in the process response to a change in the manipulated variable. The Levenberg-Marquart algorithm optimises the weights of the one-step ahead neural network predictor for *T(t+1)*.. The final result is a neural network structure that is equivalent to equation 6-20 for the process variable *T*.

Neural network models are not grounded in physical theory and are most frequently employed to capture non-linearities in process data. The validity of the model remains closely related to the quality and number of data samples in the time series. The time series must contain ample dynamic information and therefore cover an adequate region of the state space, particularly the region where the process is controlled. Although neural networks interpolate successfully between data samples, extrapolation may prove inaccurate. Model reliability is thus not only determined by the accuracy of the regression fit (i.e., interpolation ability), but also by whether or not the model is extrapolating. The simplest extrapolation indicator involves bounding the model's prediction range to the upper and lower limits of each of the independent variables. However, with multiple correlated inputs, this method will frequently overestimate the region of model validity. Leonard et al. (1992) constructed radial basis function neural networks that compute their own reliability based on a density measure. This density measure determines whether sufficient training data was available in a particular region of the state space to make an accurate prediction. Furthermore, Leonard et al. (1992) calculated a confidence limit for each model prediction based on the individual confidence limit of each radial basis function in the hidden, weighted by the contribution of each node in making that particular prediction. These two validity measures thus raised caution flags when extrapolating or when interpolating to regions of the state space where training data was sparse. The inherent localised modelling of radial basis functions was exploited effectively..

Sigmoidal feedforward neural networks, due to the more global impact of each hidden neuron, are less prone to the curse of dimensionality than radial basis networks. Though extrapolation of sigmoidal networks is seldom reliable, generally extrapolation is better than for radial basis networks (Sjöberg et al., 1995). Generally, sigmoid networks thus have fewer hidden nodes than radial basis function networks, though this generality is highly dependent on the nature of the process data. Although, the validity measures of Leonard et al. (1992) hold significant merit, the SSA dynamic models in this work have been limited to sigmoidal activation functions. Note that SSA is not limited to the structure of the model, since SSA only optimises the input space to the model. SSA could be implemented with equal effect to neural networks with radial basis activation functions or any other regression model structure (e.g. multiple linear regression).

It remains critical to have a measure of model validity for sigmoidal neural network models. Since SMNE learns directly for interactions with these neural networks, SMNE needs to have an indicator of whether the predictive capability of the dynamic model has been exceeded. To ensure that SMNE confines the neurocontroller's learned behaviour within the model's validity, the control actions must confine the predictions (i.e., model outputs) to the 95 [%] confidence limit. The confidence limit for the dynamic model is dictated by how efficiently the time series has sampled the underlying state space. In this thesis, the t-statistic for the first two principal components for the model inputs determines the model validity, provided the first two principal components explain sufficient variability (e.g., 75 [%]) in the model input space. The 95 [%] confidence limit of the first two principal components borders an elliptic region in the two-dimensional input space. A transformed model input that lies outside the elliptic region has significant statistical uncertainty.

The dynamic non-linear modelling technique described in section 6.3 has been applied to data from lysine fed-batch bioreactors.

6.4 LYSINE BIOREACTOR DYNAMIC MODELLING

6.4.1 Prior bioreactor modelling approaches

Fermentation processes utilise micro-organisms to synthesise a variety of products that include amino acids, antibiotics, fuels and various foods from suitable substrate nutrients. Advances in genetic engineering have underpinned the importance of bioprocess routes as an alternative to chemical synthesis. In fed-batch bioreactors, the batch starts from an initial volume, initial substrate concentrations (e.g., carbon and nitrogen sources) and micro-organism inoculum. The batch progresses as substrates are continuously fed into the bioreactor until a final total volume is reached. The control objective is to maintain an optimal metabolic state trajectory for the micro-organisms, thereby ensuring maximum overproduction of the desired product. Deviating from the optimal operating trajectory could produce by-products that may be deleterious to the fermentation, affect the final fermentation purity and complicate downstream processing. Likewise, small improvements in performance could result in substantial economic benefits. Conventional industrial control (e.g., PID control) is generally unsuited to the non-linear dynamics and the widely varying operating conditions that are encountered during a fermentation fed-batch.

Controller development techniques that rely on the availability of non-linear dynamic models are suited to controlling such fed-batch bioprocesses. However, non-linear dynamic model development remains an obstacle to widespread application of model-based control techniques (Narenda, 1996). Most frequently, open loop control strategies are employed for fed-batch fermentations, which allow for no corrective action during process disturbances. Although first principles modelling has been demonstrated for fermentation (Potgieter et al., 2001), the complexity of industrial substrates reduces the utility of such models in industrial practice. For example, fundamental models are frequently derived using pure substrates of known composition, resulting in poor predictions when applied to complex and unquantified industrial substrates. Narenda (1996) concluded that for accuracy and robustness, neural networks offer unique opportunities to bio-processing production routes.

Fed-batch fermentations require a multi-input multi-output control strategy. The process variables are typically dissolved oxygen, oxygen uptake rate, carbon dioxide emission rate, pH, cell density and temperature. The manipulated variables in fed-batch fermentations are typically aeration, agitation speed, chilling rate, carbon source feed rate and nitrogen source feed rate. The process variables reflect the metabolic state of the micro-organisms, which over time determines the final yield, productivity, product titre and by-product titre of the batch. A large amount of operational data is generated during a batch, with which the observed (process) variables may be related to particular metabolic state (Raju & Cooney, 1998). Raju & Cooney (1998) showed that pattern recognition techniques could be employed to infer metabolic states, such as DO limitation (i.e. anaerobic growth), nutrient limitation, production phases and growth phases. Such pattern recognition routines serve as simplified models that label the dynamic state of the fermentation, but do not include cause-effect information for moving from one metabolic state to another (i.e., state transition information).

Similarly, Willis et al. (1991 & 1992) incorporated neural networks in inferential estimation for an industrial penicillin bioreactor. In 1992, existing sensor technology did not allow direct on-line analysis of primary controlled variables such as biomass concentration. Typically, biomass concentrations remain at-line analyses, sampled at infrequent intervals of 4 hours. Such infrequent sampling presented operational difficulties in penicillin production. The production of penicillin G by *Penicillum chrysogenum* was run fed-batch in two distinct operating phases. Initially, the fermentation produces large quantities of biomass, predominantly using the substrate in initial charge media. As the substrate in the initial charge becomes the limiting nutrient, substrate is fed to the bioreactor keeping the reactor substrate concentration low. Owing to the low reactor substrate concentration, the growth rate decreases and this switches the metabolic state of the organism to penicillin production. The yield of penicillin is maximised at a true optimum biomass growth rate. Higher biomass growth rates reduce the yield of penicillin drastically. Lower growth rates induce lysis conditions. The optimum growth rate is located in close proximity to the lysis operating constraint. Owing to the infrequent biomass analyses, a conservative feed strategy is employed that rather accepted lower productivity than lysis conditions. Closer operating to the optimal growth rate is thus desired. Three on-line process variables provided pertinent information regarding the progress of the fermentation, viz. carbon dioxide emission rate (CER), the batch age and the feed rates of the two components that served as substrate feed. Willis et al. (1992) demonstrated that a neural network could infer the biomass concentration, providing accurate estimates using the on-line process variables. This inferential model for biomass was used subsequently to control biomass concentration along a predefined biomass concentration trajectory. A neural network model thus allowed tight operation at the economic optimum of the non-linear bioreactor process.

Zhu et al. (1996) used neural networks for predicting the total lysine production during a fed-batch *Brevibacterium flavum* fermentation. Figure 6-10 shows the neural network architecture comprised of a neural estimator and a neural predictor. The neural estimator inferred the total consumed sugar since the start of the batch, based on total CO_2 emission and the current respiratory quotient (RQ). The inferred total consumed sugar formed the basis for predicting the total produced lysine. Essentially, Zhu et al. (1996) established an accurate carbon balance, i.e. the sugar represented "carbon in", the total CO_2 "carbon out" and the lysine production "carbon accumulated". The RQ(t) and the lagged consumed total sugar provided sufficient dynamic information to predict future lysine production at (t+1) and (t+2). Zhu et al. (1996) concluded that neural networks present encouraging opportunities as soft-sensors in bioprocessing.

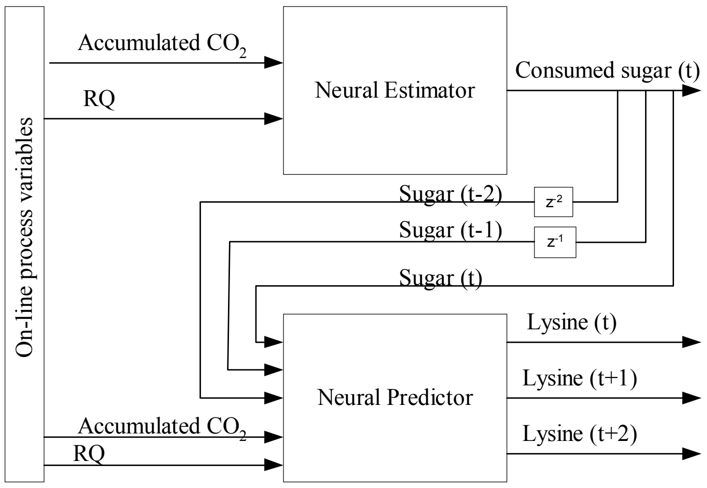


Figure 6-10 - A hierarchical neural network predictor for lysine fermentations proposed by Zhu et al. (1996).

6.4.2 Lysine bioreactor dynamic model development using SSA

The approach by Zhu et al. (1996) involved trial-and-error testing of the inputs to the neural networks. Singular spectrum analysis may offer a more systematic approach to model identification, grounded in non-linear system theory. For industrial confidentiality, a thorough description of these lysine fermentations is omitted. Historical plant data was obtained from 28 lysine fed-batch fermentations run in large scale bioreactors. Each fermenter is initially charged with complex media and initial sugars. During the first phase of fermentation, large quantities of biomass are produced, until key nutrients in the initial charge are exhausted. Thereafter the micro-organism switches the metabolic pathways to lysine overproduction. The fermentation is primarily controlled by feeding high-test molasses (HTM), ammonia (NH_3) and ammonium sulphate (AS) into the bioreactor. The HTM serves as carbon source, while the NH_3 feed serves as pH control and nitrogen source. The $(NH_4)_2SO_4$ feed provides for residual NH_4^+ as nitrogen source and sulphates that facilitates lysine efflux from the cells. The fed-batch is extended by removing partial volumes during fermentation that increases the total volume of the fermenter, but also removes biomass. The bioreactions for lysine production may be lumped into equation 6-21 to equation 6-23.

$$a \cdot C_6H_{12}O_6 + b \cdot O_2 + c \cdot NH_4^+ \rightarrow d \cdot C_6H_{15}N_2O_2^+ + e \cdot C_1H_{1.8}O_{0.5}N_{0.2} + f \cdot H^+ + g \cdot CO_2 + h \cdot H_2O + i \cdot \Delta E \quad \textbf{(6-21)}$$

$$NH_3 + H_2O \rightarrow NH_4^+ + OH^- \quad \textbf{(6-22)}$$

$$2 \cdot Lys^+ + (NH_4)_2SO_4 \rightarrow (Lys)_2SO_4 + 2 \cdot NH_4^+ \quad \textbf{(6-23)}$$

The objective of the modelling exercise was the development of a one-step ahead predictor for lysine concentration. Essentially, the underlying dynamics of the carbon balance in the bioreactor needed to be extracted from on-line time series data. Several time series were identified as possible candidates within a multi-channel SSA framework. The CER time series was included into the embedding matrix (LEM), since the CER contained both carbon balance and pertinent state information of the micro-organisms. The respiratory quotient (RQ) was included indirectly into the embedding, by adding the OUR time series into the LEM. Since partial volume drops impacted on the total biomass, the volume time series was added to the multi-channel embedding.

The CER time series was considered the most important time series in reconstructing the state space from a carbon balance perspective, since the CER provides a measure of carbon source uptake rate and consequently lysine production (equation 6-21). Therefore, the window length, τ_w, in the embedding was determined from the CER time series. As seen in Figure 6-11, a linear autocorrelation analysis of the CER time series revealed that τ_w should be approximately 43 sample periods. Taking a non-linear perspective on statistical independence, the average mutual information was calculated as in Figure 6-12. The first minimum in Figure 6-12 suggested a lag or window length of 9 sample periods. A lag of 9 sample periods corresponded to a time constant typically used by process operators to control the batch trajectory of the fermentation. The greater accuracy of AMI in predicting the embedding lag may be attributed to the non-linear nature of the fermentation and minimal sensor noise in the CER time series.

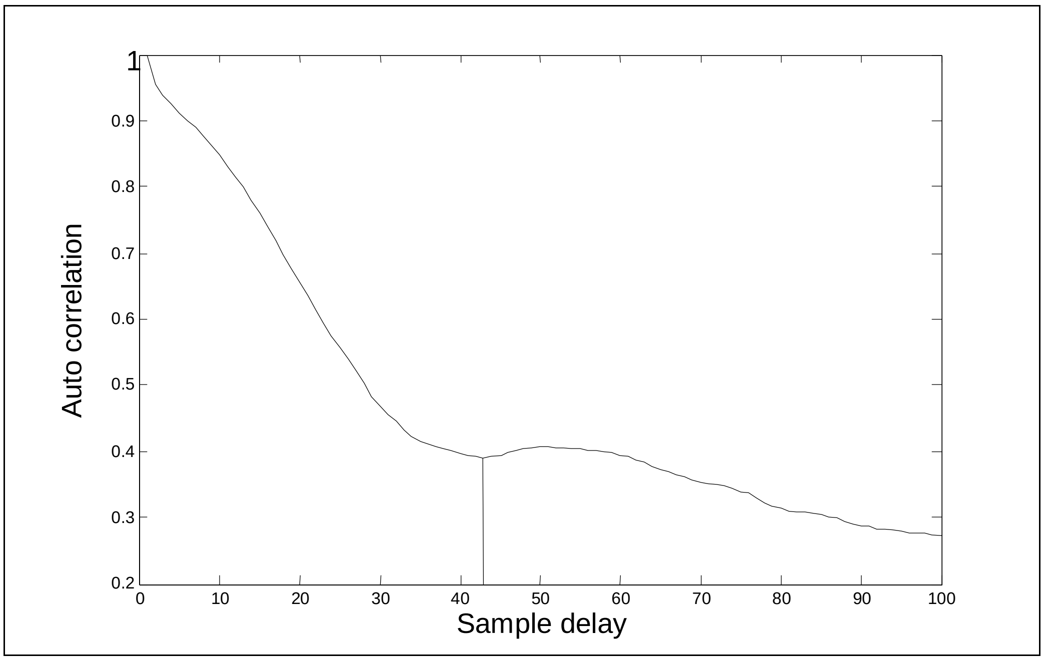


Figure 6-11 - Linear autocorrelation for lysine bioreactor CER data.

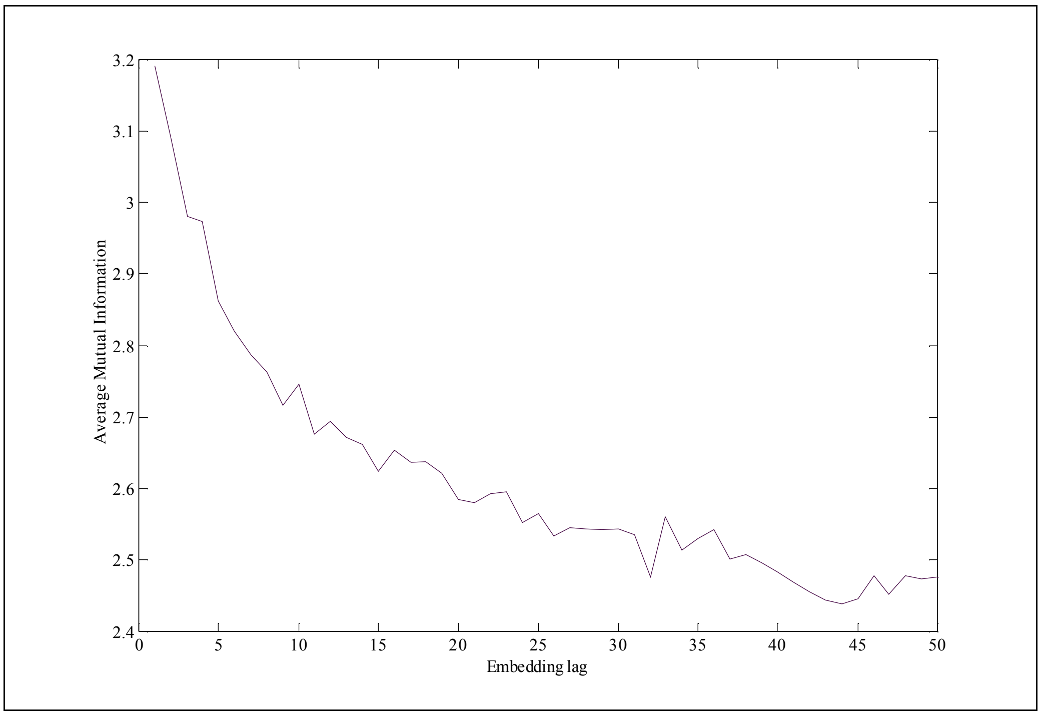


Figure 6-12 - Non-linear determination of window length using average mutual information on the CER time series.

The principal component analysis of multi-channel embedding showed that 97.5 [%] of the variance in the embedding may be explained by the first four principal components. The first four principal components were selected as the dominant state variables for a lysine fermentation. A non-linear dynamic model was constructed using these four state variables and time-delayed inputs of two manipulated variables, viz. the HTM and the AS feed rates, as inputs to the neural network (Figure 6-13). The cumulative CO_2 emission input was included as a reference to the progress of the batch, since the fermentation has a wide operating range in state space. A neural network with 8 sigmoidal hidden nodes was trained using the Levenberg-Marquart algorithm to predict CER(t+1), OUR(t+1), Volume(t+1) and Lysine(t+1). The training set comprised 50 [%] of the available data (randomly selected), while the balance comprised the validation set.

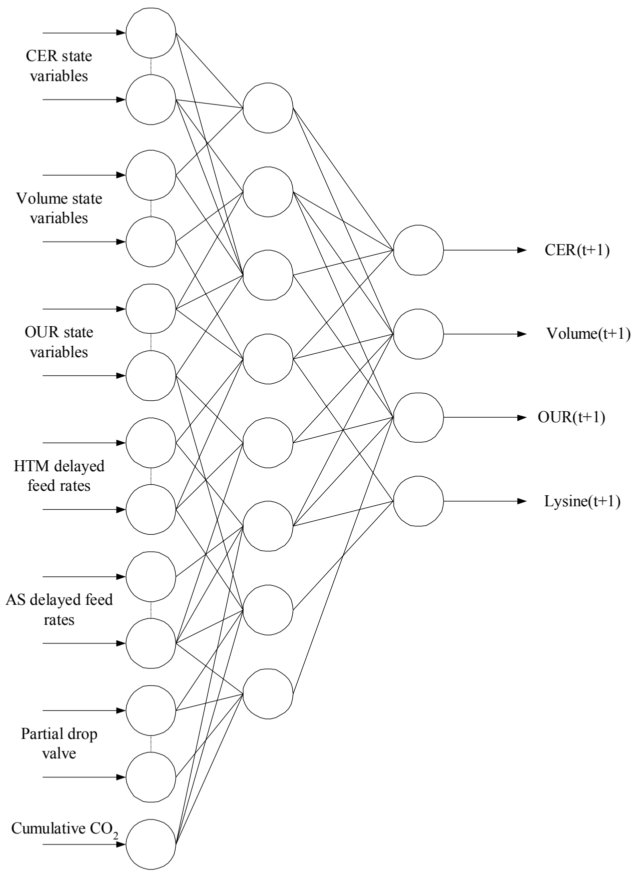


Figure 6-13 - Neural network model structure for lysine concentration. State variables for the carbon-balance empirical differential equation were extracted primarily from CER data.

Table 6-3 summarises the R^2-statistics for the one-step ahead predictor on validation data. The high R^2-statistics for all the model outputs indicates the neural network model generalises to inputs not included in the training set and accurately predicts the lysine titre. Figure 6-14 and Figure 6-15 illustrate the accuracy of the model prediction as compared to the actual plant data. Note that such a neural network is only a partial model for the fermentation, assuming that set points for aeration, pH and agitation speed are uniformly applied from batch to batch. Partial pseudo-empirical models are also limited by the sampled range in each time series, as shown in Figure 6-16. Figure 6-16 shows the batch initial condition in the bottom left-hand quadrant. The elliptical 95 [%] confidence limit for the first two principal components encompass the majority of the batch data, but the start-up period remains outside the area of model validity. Importantly, model validity is purely a guide for neurocontroller development with the SMNE algorithm. The validity of the model input space limits the search of the SMNE algorithm for an optimal neurocontroller as described in chapter 7.

Table 6-3 - R^2-statistics for the one-step ahead lysine bioreactor model predictor.

Process variable	*R^2-statistic*
CER	0.93
OUR	0.91
Level	0.97
Lysine titre	0.95

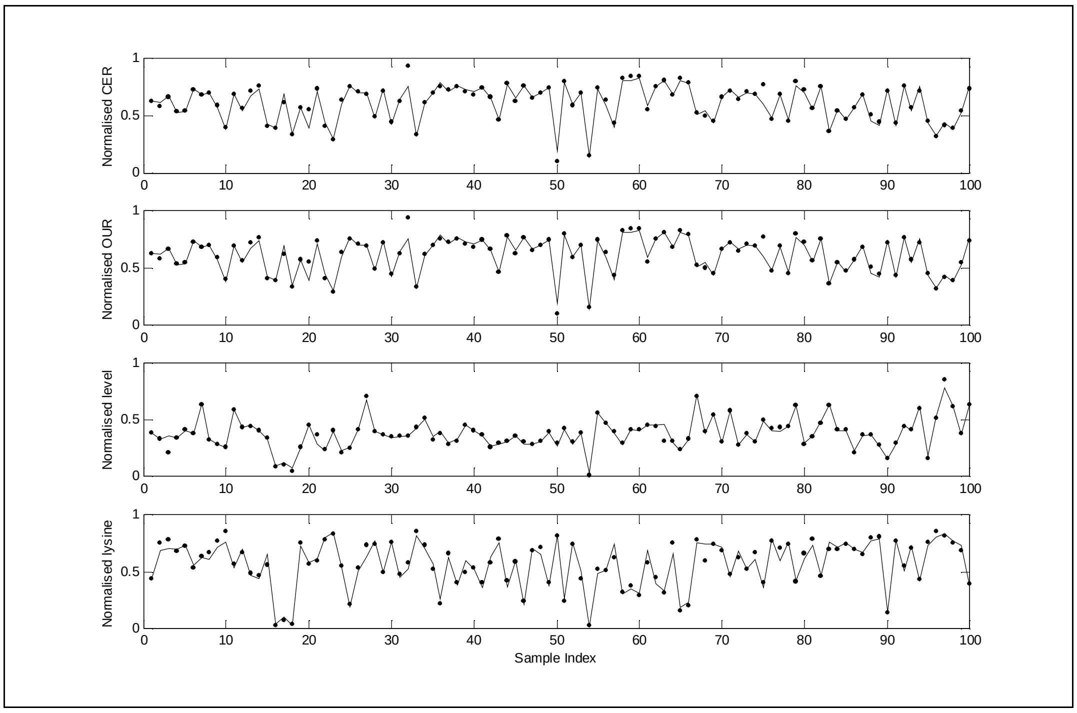


Figure 6-14 - Randomised actual (markers) and one-step ahead predicted (line) data for carbon emission rate (CER), level (L) and lysine concentration (LYS). Randomised and normalised data protects the proprietary nature of the fermentation process.

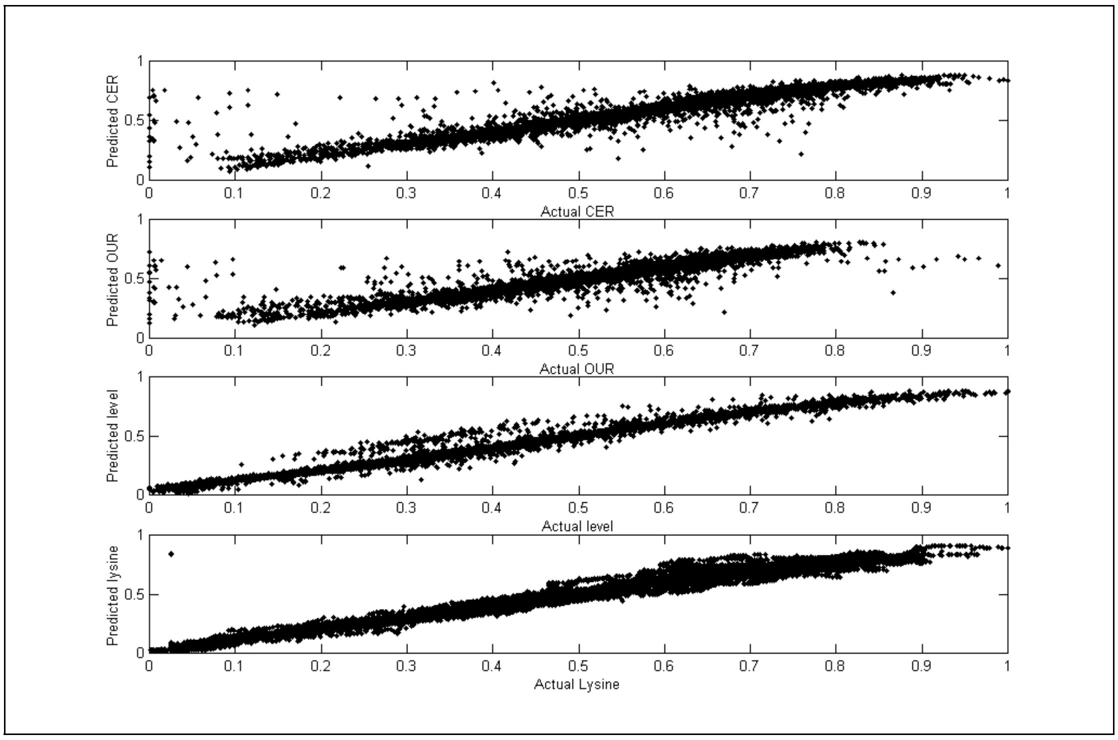


Figure 6-15 - Regression fit for carbon emission rate (CER), level and lysine concentration, plotting actual versus predicted, over the normalised data range. Deviations from a 45° line show inaccurate predictions.

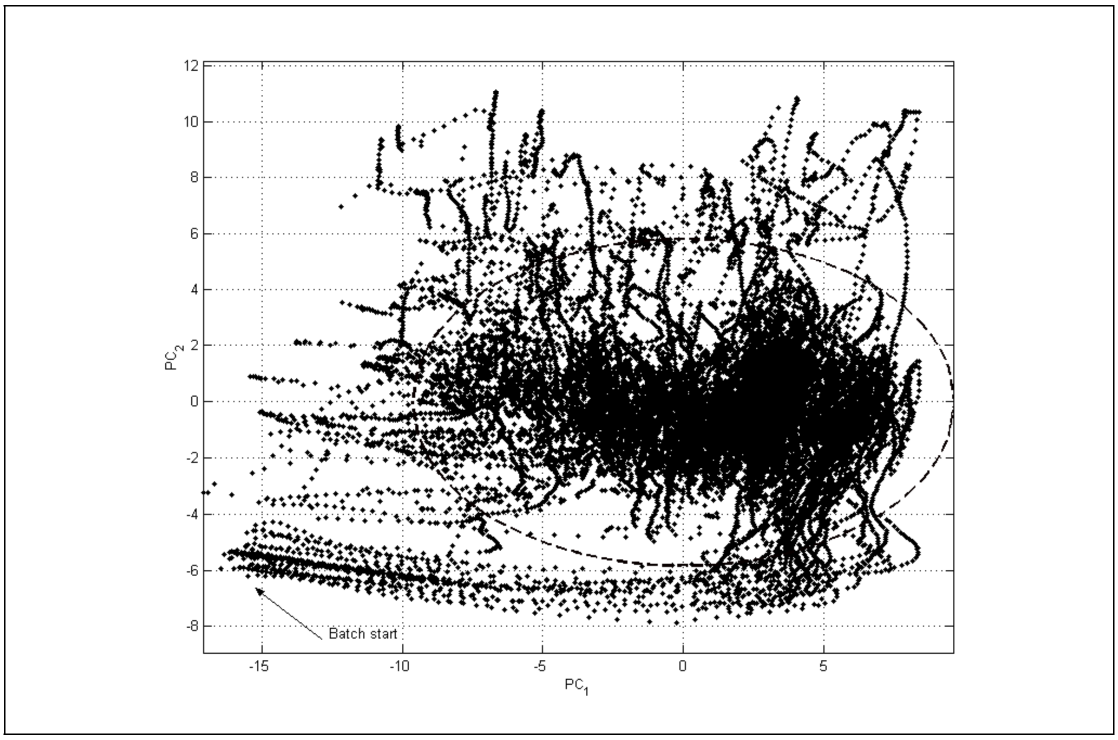


Figure 6-16 - Model validity of a lysine fed-batch bioreactor. Dashed lines indicate the 95 [%] confidence limit with the first two principal components describing 62 [%] of the variability in the model input space.

6.5 CONCLUDING REMARKS FOR CHAPTER 6

Developing phenomenological models is both time-consuming and costly and may be infeasible for poorly understood processes. Modern process management systems log vast quantities of historical process information to data warehouses. The SSA methodology outlined in chapter 6 provides a power tool for rapid pseudo-empirical model development from historical input-output data. The general applicability of SSA to dynamic process modelling is demonstrated for a non-linear batch process, viz. a lysine fed-batch fermentation. SSA allows the construction of partial models from dominant state variables. Such dynamic models are ideal for use in the SMNE neurocontroller development framework.

7 NEUROCONTROL OF A MULTI-EFFECT BATCH DISTILLATION PILOT PLANT

OBJECTIVES OF CHAPTER 7
• Demonstrate neurocontroller development from pseudo-empirical neural network dynamic models using the SMNE algorithm.

New control algorithms are most frequently evaluated within the confines of simulation studies. Even though these simulated processes may have demanding process control requirements, new control algorithms often show significant promise. Yet, few reports of such new control algorithms describe a real industrial application. Simulation studies are predominant, since simulated processes offer the advantages listed in Table 7-1 (Kershenbaum, 2000).

Table 7-1 - Advantages of simulated studies (Kershenbaum, 2000).

Advantages of simulated control testing
The utility of testing a new algorithm under clearly defined conditions.
The impact of disturbances and measurement error may be tested in isolation.
Robustness to plant-model mismatch may be evaluated under controlled conditions.
The sensitivity of the algorithm's tuning parameters may be related directly to well-defined process conditions.
Rapid method of evaluation.

Despite these advantages, simulation studies offer limited guidance on how a new control algorithm will perform faced with unknown process conditions that may exist in industrial application. Experimental studies offer all the benefits listed in Table 7-1, also validating the new algorithm under poorly defined, more realistic conditions that lack the artificial niceties of simulated environments. Kershenbaum (2000) found that experimental evaluation leads to striking differences in performance with regard to measurement noise, variable time delays, process lags and unforeseen non-linearities. On-line control experiments generally reveal additional complications that were not envisaged or duplicated in a simulation. Even the most meticulous modelling effort cannot include all constraints and process dynamics uncertainties. The law of diminishing returns is soon prevalent in lengthy modelling exercises that also coincide with rapidly escalating costs (Kershenbaum, 2000).

Control studies of batch processes suffer most acutely from discrepancies between simulated and experimental evaluation.

7.1 BATCH PROCESSING CASE STUDY

Batch processing is a challenging control problem suited to non-linear control, since a typical batch operates over a wide operating range and the process dynamics change significantly due to non-linearities. The control objective is fundamentally different from continuous processing, rather posed as an optimal tracking of non-stationary states. Obtaining an accurate batch model is difficult, making a conventional model-based control system susceptible to tracking errors. As a result, few applications of MPC exist for batch processing. An MPC approach may need to be tailored to the specific requirements and characteristics of batch processing, making generic tools difficult to implement (Morari and Lee, 1998).

Batch distillation is frequently employed to purify high added value products in the fine, pharmaceutical and biochemical industries. Batch distillation columns are broadly classified as rectifying, stripping and emerging columns. Emerging columns are comprised of a rectifying column and several stripping columns, depending on the number of components, connected via vessels for product hold-up. Batch distillation is flexible and a single batch distillation unit can separate a variety of feed mixtures of uncertain initial composition. Thereby, batch distillation units are able to respond swiftly to market demands. Emerging column designs separate a mixture of *B* components with *(B-1)* columns into pure components without off-cut products. Emerging columns also have promise in extractive and reactive batch distillation. Although batch rectifiers require far less capital investment than a continuous column, batch rectifiers are less energy efficient than continuous distillation columns (Kim and Diwekar, 2001).

7.2 MULTI-EFFECT BATCH DISTILLATION COLUMNS

An emerging configuration, viz. multi-effect batch distillation (MEBAD) or multi-vessel column, has bridged the divide between the energy efficiency of rectifiers and continuous columns. The MEBAD column's heat integration allows optimal use of the reboiler's heat duty in the train of stripper columns. Simulation studies have confirmed that such emerging columns typically consume 50 [%] less energy than a conventional rectifier for the same initial charge (Furlonge et al., 1999). The energy efficiency of MEBAD columns approach that of continuous columns (Kuroka et al., 2001).

The greater flexibility of MEBAD columns complicates optimal operation. Possible control policies may include: (1) optimal distribution of the initial charge to the product vessels, (2) optimal hold-up in each intermediate vessel during operation, (3) optimal reflux in each column section, (4) variable heat duty, (5) product withdrawal to accumulator vessels during operation and (6) adaptive tuning of controller parameters. The more complex process dynamics, as a result of significant process interactions between column sections, compounds the selection of an operating policy. Considering the large number of decision variables, a control strategy should be multi-input multi-output (MIMO) and determined from a plant-wide control perspective (Furlonge et al., 1999).

An appropriate performance index for optimal MEBAD control minimises the mean rate of energy consumption until steady state is reached in the intermediate vessels:

$$\varepsilon = \min_{t_f, Q_R, t_s} \frac{\int_0^{t_f} Q_R(t)dt}{t_f + t_s} \qquad \textbf{(7-1)}$$

where ε is the minimum mean energy consumption rate, $Q_R(t)$ is the instantaneous rate of energy consumption in the reboiler, t_f is the processing time and t_s is the set-up time for each batch. Minimising ε is related directly to maximising the profit from the batch (Furlonge et al., 1999).

7.2.1 Existing control policies

Noda et al. (2000) investigated the operational policy of an on-line optimisation system for the MEBAD column. In the proposed variable policy (V-policy), the initial and operational hold-ups in each intermediate vessel are optimised as a function of time. The optimisation is undertaken using a dynamic model of the MEBAD configuration and is implemented in open loop operation. However, the plant and the mathematical model are invariably mismatched. Even though the vessels follow the optimal hold-up profiles, the vessel compositions failed to track the optimal composition profiles. This discrepancy requires an on-line optimisation system in which the optimal operation profile is re-evaluated based on the error between the current process condition and the optimal condition. Re-evaluation of the optimal hold-up profile requires solving the non-linear programming problem periodically. This provides a degree of feedback control, which also requires a near infra-red (NIR) analyser to measure the compositions in the intermediate vessels. The need for a NIR analyser makes the control scheme expensive. Furthermore, the NIR analyser needs to be calibrated using many known samples, collected over a wide range of temperatures and compositions. The re-evaluated hold-up profiles serve as set points for the controllers that control the reflux rates from each vessel. The tuning parameters of these SISO controllers also need optimisation and decoupling. The control scheme was verified in a pilot plant MEBAD column. The complete initial charge was fed to the reboiler and a constant heat duty was maintained throughout the batch. The experimental results confirmed that the vessel compositions track the optimal trajectory.

Wittgens and Skogestad (2000) proposed a closed loop total reflux control policy. A constant temperature in the middle of each column is maintained by controlling the reflux rate to each column. As the batch progresses, the separation in each column section becomes pseudo-binary and the temperature measurement serves as inferential estimate for composition control. With a fixed temperature in each column section, the column and vessel temperature profiles are fixed thermodynamically. The degrees of freedom for a MEBAD column, with constant heat duty and no product withdrawal, is equal to the number of columns. With the entire initial charge in the reboiler and constant heat duty, the hold-up in each intermediate vessels increases gradually to the final hold-up. Linear proportional-integral (PI) controllers maintain the desired column temperatures. Wittgens and Skogestad (2000) proposed that the temperature set point in each column is the boiling point mean of the two components separated in that column section. The control strategy was verified in a laboratory scale multi-vessel column. This robust control policy is simple and the indirect control of vessel hold-ups reaches the final product compositions regardless of the initial charge composition. Though robust to process uncertainties and process disturbances, this control policy is not optimal. Furlonge et al. (1999) found that the temperature control strategy (i.e., total reflux) does not relate directly to the economic objective of minimising the batch time or energy consumption. In simulations, the mean energy consumption for the total reflux control strategy was greater than for the open-loop optimisation of the composition and hold-up profiles. Also, the mean energy consumption increased as the set point tracking performance of the temperature controllers was improved. Furthermore, the use of two SISO controllers without decoupling neglected the process interaction between the columns.

Furlonge et al. (1999) considered optimal operation of the MEBAD column in a simulation study that considered all the possible control degrees of freedom. In this open loop control policy, distributing the initial charge, product withdrawal during operation and variable heat duty was considered. Charging the entire feed to the reboiler resulted in better operation than distributing the feed by composition to the intermediate vessels. Also, product withdrawal offered no significant benefit over total reflux operation. However, the simulation results confirmed that the optimal hold-up scheme suggested by Noda et al. (2000) may be improved by optimising both the heat duty and the vessel hold-ups over time.
The optimal hold-up control strategies (Noda et al. (2000) & Furlonge (1999)) have entailed both a complex phenomenological batch modelling and the use of expensive analytical equipment. In contrast, the control strategy by Wittgens & Skogestad

(2000) is sub-optimal but industrially robust. The gap between the total reflux and the optimal hold-up strategies may be bridged with SSA and SMNE. Such a neurocontrol strategy allows for rapid pseudo-empirical model development and a model-based controller that relied on cost-effective temperature measurement. A model-based control strategy provides for optimal temperature profiles throughout the batch.

7.3 EXPERIMENTAL SET-UP

A MEBAD column was constructed on pilot plant scale to demonstrate the use of SMNE with a SSA model (Figure 7-1). The MEBAD column was operated at atmospheric pressure to separate a ternary system of n-pentane, n-hexane and n-heptane. The two insulated stainless steel columns have an inner diameter of 0.068 [m] with Montz BSH 400 structured packing. Structured packing has low liquid hold-up and pressure drop, with large surface area and excellent wetting characteristics. Column A has a length of 2 [m] with 1.9 [m] of structured packing, while column B has a length of 2.6 [m] with 2.5 [m] of structured packing. Each column has a high performance orifice distributor to ensure maximum packing wetting. However, the available structured packing is insufficient to separate the components with high purity. A 50 [dm^3] spherical glass reboiler, a 30 [dm^3] glass intermediate vessel and a 20 [dm^3] glass reflux drum comprise the MEBAD unit's vessels. The vapour from column A is fed directly into the liquid hold-up of the intermediate vessel, heating the liquid hold-up through this heat integration. The reboiler has a 1.5 [kW] stab-in heating element and a 3 [kW] heating mantle that is controlled using a solid state relay. A glass coiled vertical condenser provides a cooling duty of 3 [kW]. Two positive displacement pumps, each with a maximum flow rate of 60 [$dm^3 \cdot h^{-1}$], provide the reflux to the columns. The reflux rates are controlled using variable speed drives.

The MEBAD unit has 11 PT100 temperature sensors, 3 hydraulic pressure level sensors and 2 column differential pressure sensors (Figure 7-1). Analogue-to-digital conversion cards convert the sensor signals with a sampling frequency of 0.25 [s]. A median filter (i.e., non-linear filter) reduces measurement noise, by taking the median of the past four and the current signal reading. A Delphi MMI (man-machine interface) interfaces with the plant's sensors and final control elements, executes the control laws and logs the sensor readings to a database.

The entire initial charge was fed to the reboiler. A PI controller ensured a constant heat duty throughout the batch, by maintaining a temperature difference of 100 [°C] between the heating mantle and the reboiler's liquid hold-up. The PI controller set the time interval during which the heating mantle was powered within a 10 [s] window. The stab-in heating coil was always powered. This resulted in a mean heat duty of 2.5 [kW] in all the control experiments.

Start-up involved establishing total reflux in each column with minimal hold-up in the intermediate vessel (i.e., 5 [%]) and the reflux drum (i.e., 9 [%]). The desired product temperatures were specified as 93 [°C] in the reboiler and 66 [°C] in the intermediate vessel, thereby also fixing the product composition in the reflux drum. Thereafter the control laws were activated and the batch run until steady state was reached.

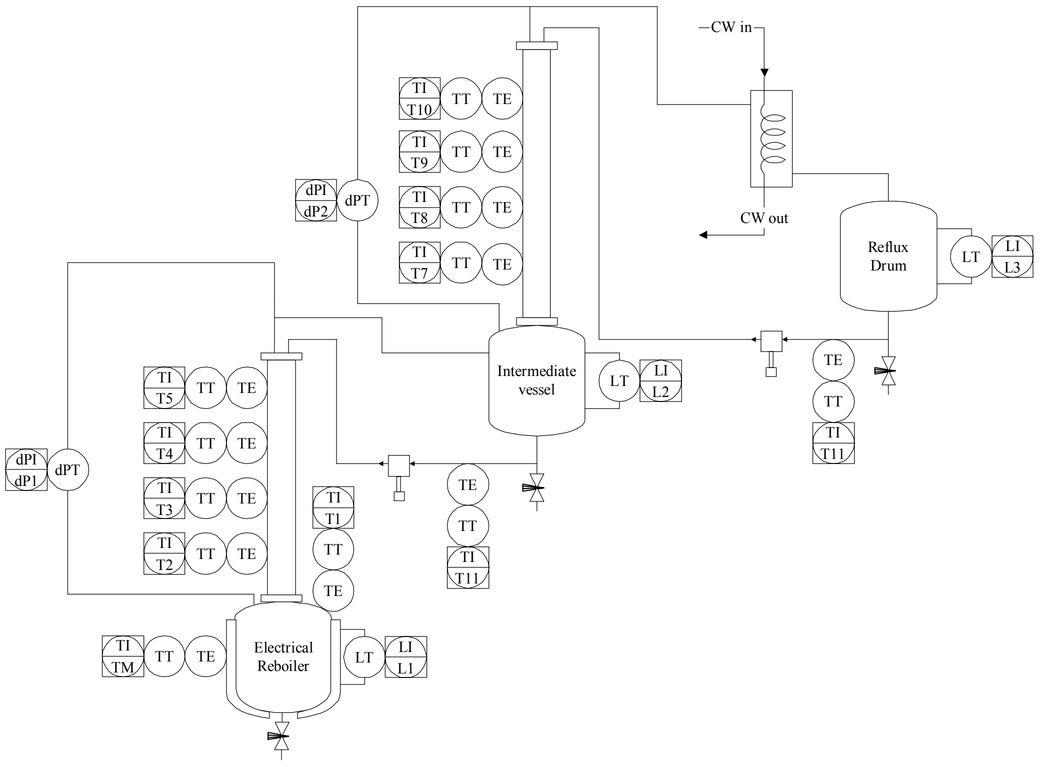


Figure 7-1. Pilot plant MEBAD set-up.

7.4 COLUMN DYNAMIC BEHAVIOUR

For a MEBAD unit with two columns and constant heat duty, the optimal control strategy involves two degrees of freedom. This results in a 2x2 control system. However, the control strategies proposed by Wittgens and Skogestad (2000) and Noda et al. (2000) are essentially multi-loop SISO control systems. The process-manipulated variable pairings are simple, but the two distillation columns display process interaction. This complicates SISO control, where no consideration is given to decoupling the controllers. A MIMO control design is thus preferable.

The process interaction is illustrated in Figure 7-2 to Figure 7-4. After total reflux was established in the MEBAD columns, a step change was made at 0.09 [h] in the reflux rate to column A (i.e., *F1*), while maintaining the constant reflux rate in column

B (i.e., *F2*). In column A, *T4* began to decrease as a result of the increased reflux from the intermediate vessel (Figure 7-3). The composition in the intermediate vessel became more volatile as the n-hexane was returned to the reboiler (i.e., *T1* and *T6* decreased), reflected by the increase in *L1* and the decrease in *L2* after 0.15 [h] (Figure 7-4). At 0.15 [h], *T9* in column B also started to decrease due to the change in *T6*, until *T9* had decreased significantly by 0.3 [h] (Figure 7-3). The level in the reflux drum stayed constant due to the total reflux operation in column B. Clearly, a change in *F1* not only affects the temperature profile in column A, but also the temperature profile in column B. A similar argument follows for a step change in column B. The column temperature responses are coupled, which implies that two SISO controllers, without decoupling, interact.

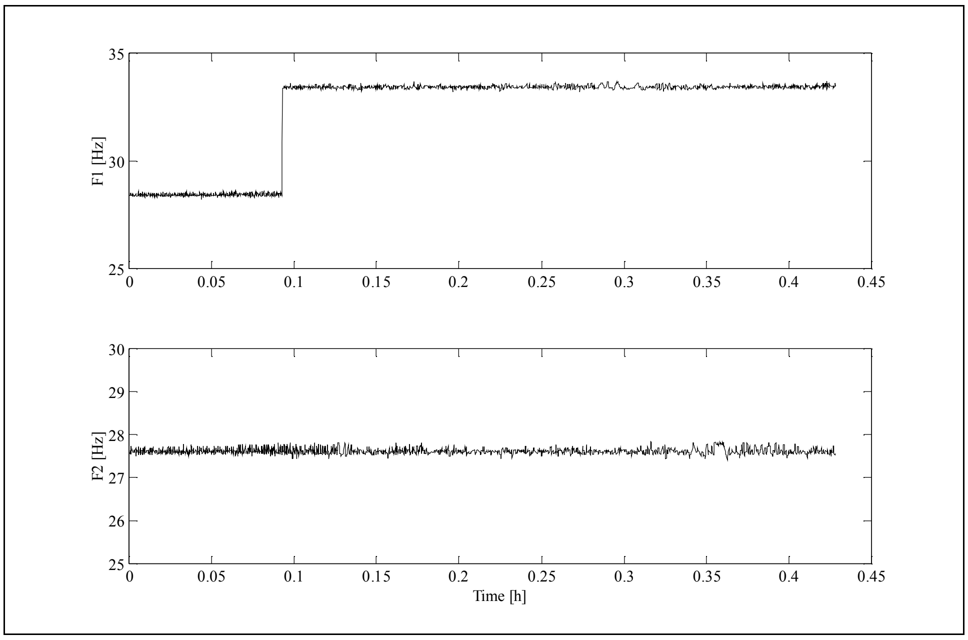


Figure 7-2. Step change in the reflux rate to column A, while maintaining constant reflux to column B.

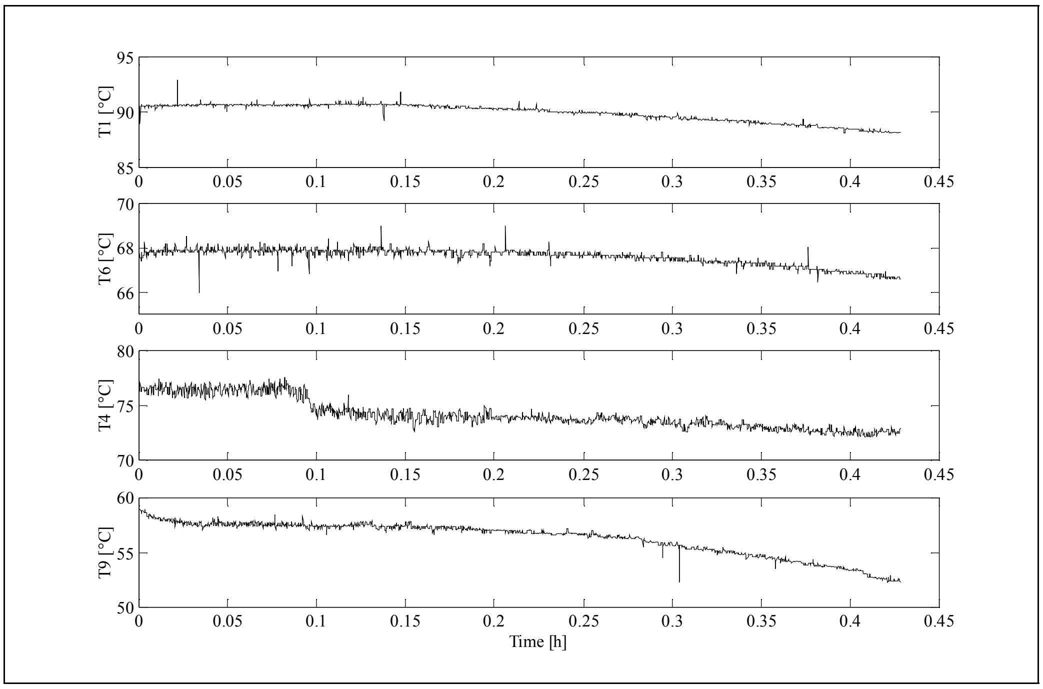


Figure 7-3. Process variable responses to the step change in reflux rate to column A. T1 and T6 are the temperatures in the reboiler and intermediate vessel respectively, whilst T4 is a temperature in column A and T9 a temperature in column B. The process interaction between the process variables is evident.

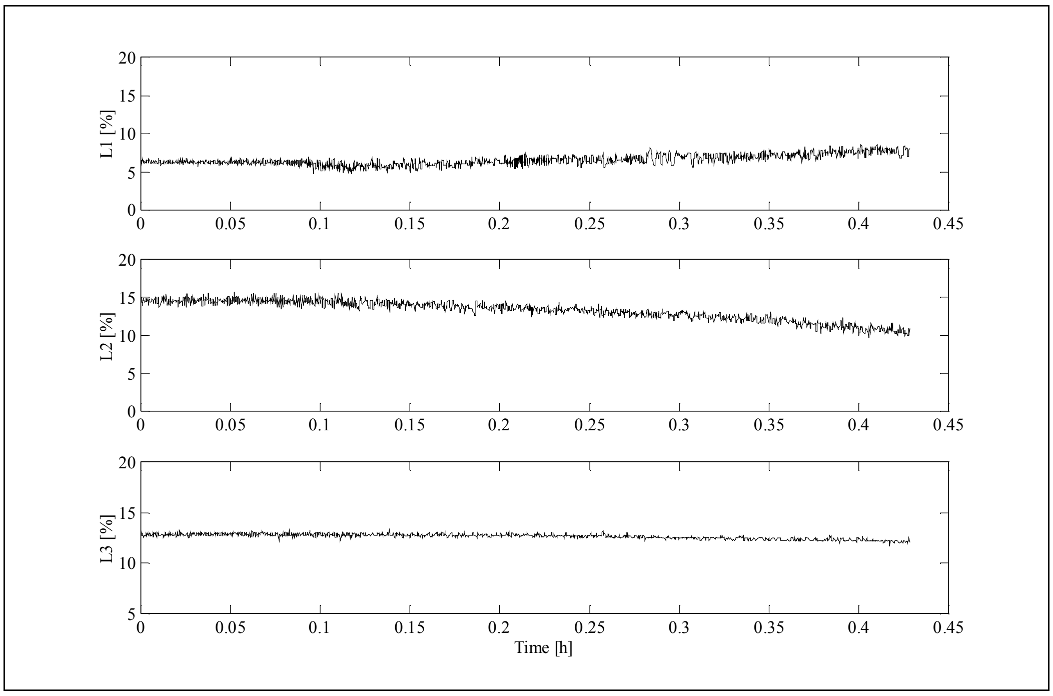


Figure 7-4. Vessel level responses to a step change in reflux rate to column A.

7.4.1 Closed loop behaviour for total reflux multi-loop PI control strategy

For variable impurities in the initial charge, it may be necessary to adjust the temperature set points for the MEBAD columns during a batch. This scenario illustrates controller interaction between two columns. Consider a decrease in the temperature set point in column A from a steady state. During closed loop operation, *F1* increases to decrease *T4*. A lower *T4* is attained swiftly. However, *T9* begins to decrease as *T6* eventually decreases, whereby *F2* decreases to keep the composition in the reflux drum constant. *F1* also decreases as *T6* decreases. As *T6* subsequently increases, both *F1* and *F2* increase to maintain the set points in column A and B. This slight oscillation in *T6* continues until *T6* settles to a new steady state temperature (*F1* and *F2* return to total reflux). This controller interaction delays the approach to steady state.

Nevertheless, the PI control strategy (Figure 7-5) is robust, inexpensive and the implementation time short. More complex control strategies would have to improve performance considerably to justify their implementation. The PI control strategy by Wittgens and Skogestad (2000) was implemented to serve a base line or reduced model for comparison with the more complex neurocontrol strategy.

Selecting the temperature set points for PI control requires a single experiment to establish which column temperature set points match the temperatures specified in the reboiler and intermediate vessel. The steady state product temperatures were set as 93 [°C] and 66 [°C] in the reboiler and intermediate vessel respectively. At steady state, the column temperature set points were changed appropriately until the desired product temperatures (i.e., inferred compositions) were obtained.

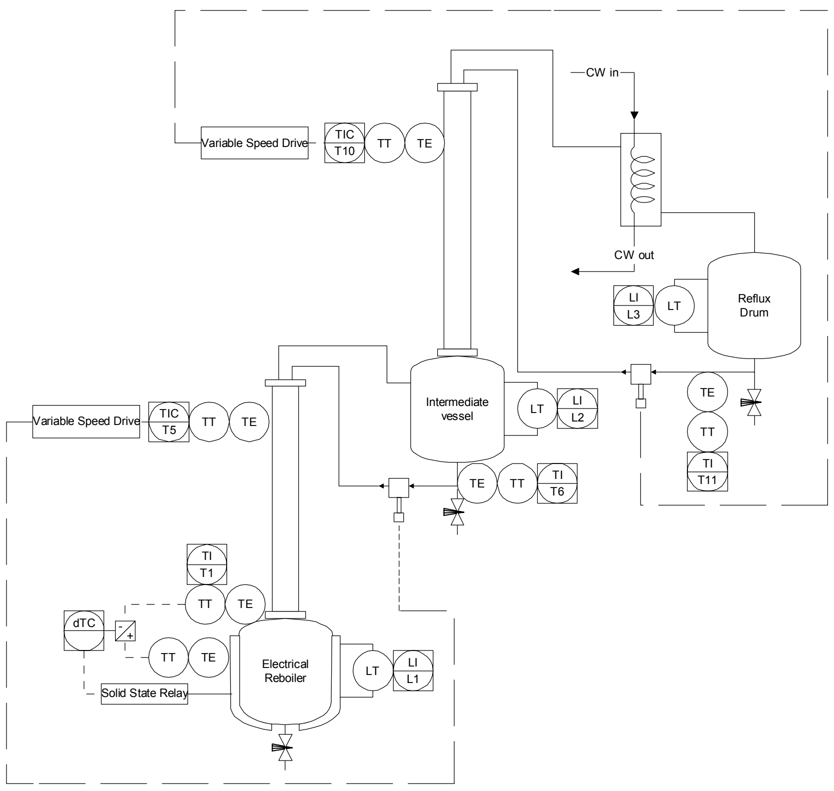


Figure 7-5. PI temperature control scheme as proposed by Wittgens and Skogestad (2000).

Temperatures *T5* and *T10* were the most appropriate column locations for PI control, since these temperature sensors responded soonest to changes in the reflux rates. No significant lag time in these temperature responses was observed, with sufficient structured packing above the sensors for temperature profile development. The PI controller parameters were selected to ensure high performance servo and regulatory responses during the batch.

Figure 7-6 and Figure 7-7 show the temperature and level responses for the PI control batch over a 7 [h] period, after activating the PI control. Figure 7-8 shows the reflux rates set by the PI controllers during the batch. The low initial reflux rates allowed the column temperatures *T5* and *T10* to increase rapidly towards their set points (Figure 7-6c & Figure 7-6d), increasing the liquid hold-ups *L2* and *L3* significantly. The overdamped response in *T5* and the slight overshoot in *T10* were deemed appropriate servo responses. The PI tuning parameters also ensured high performance set point tracking.

However, Figure 7-6a shows the slow approach of *T1* to the desired reboiler temperature. At 7 [h], *T1* was ± 2 [°C] below the desired temperature, while *T6* was slightly above the desired temperature. The vessel temperatures (i.e., inferentially the compositions) are the process variables that reflect the true progress of the batch. The batch only attained steady state at a batch time of 11 [h], due to slow purification of the reboiler volume. *T1*'s slow settling time suggested sub-optimal temperature and hold-up profiles in the column.

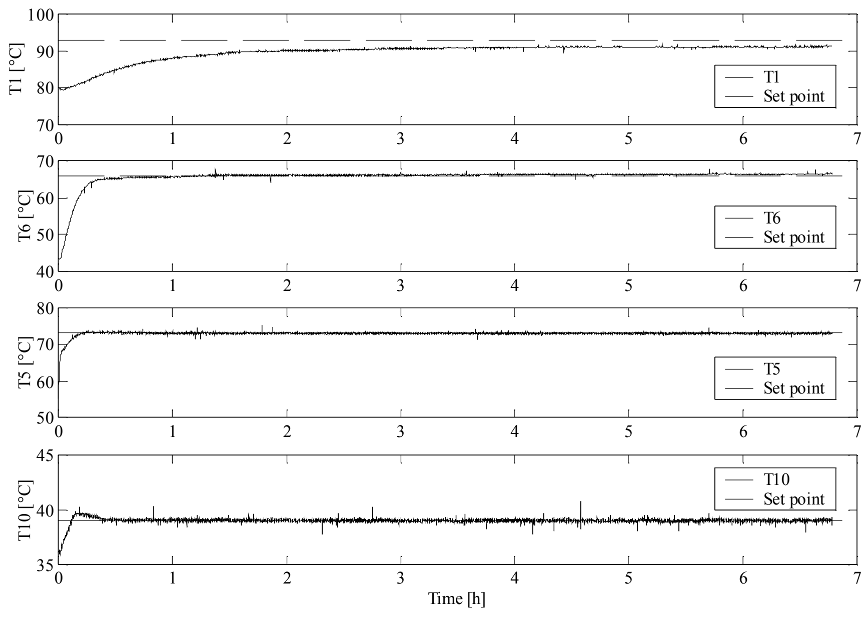


Figure 7-6. Process variable control responses for PI temperature control over a 7 hour period.

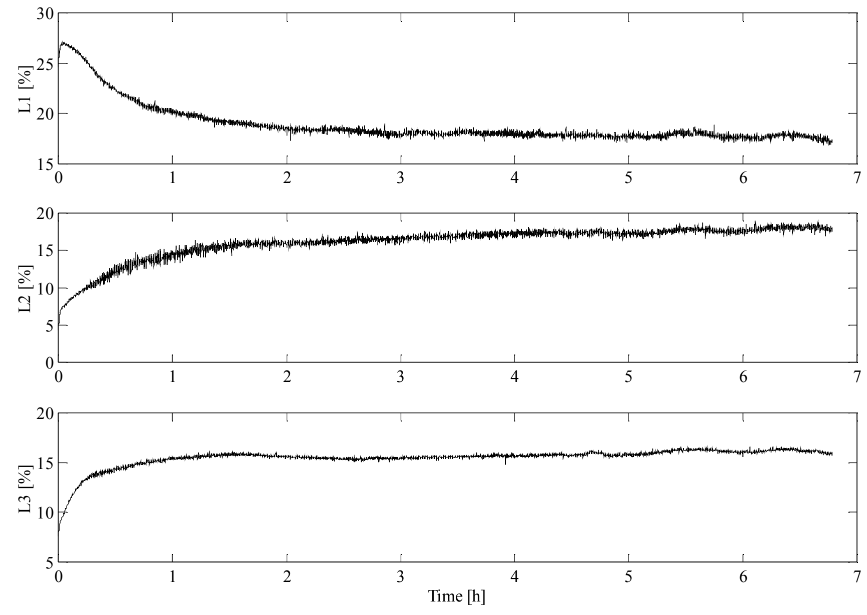


Figure 7-7. Vessel level responses for PI temperature control over a 7 hour period.

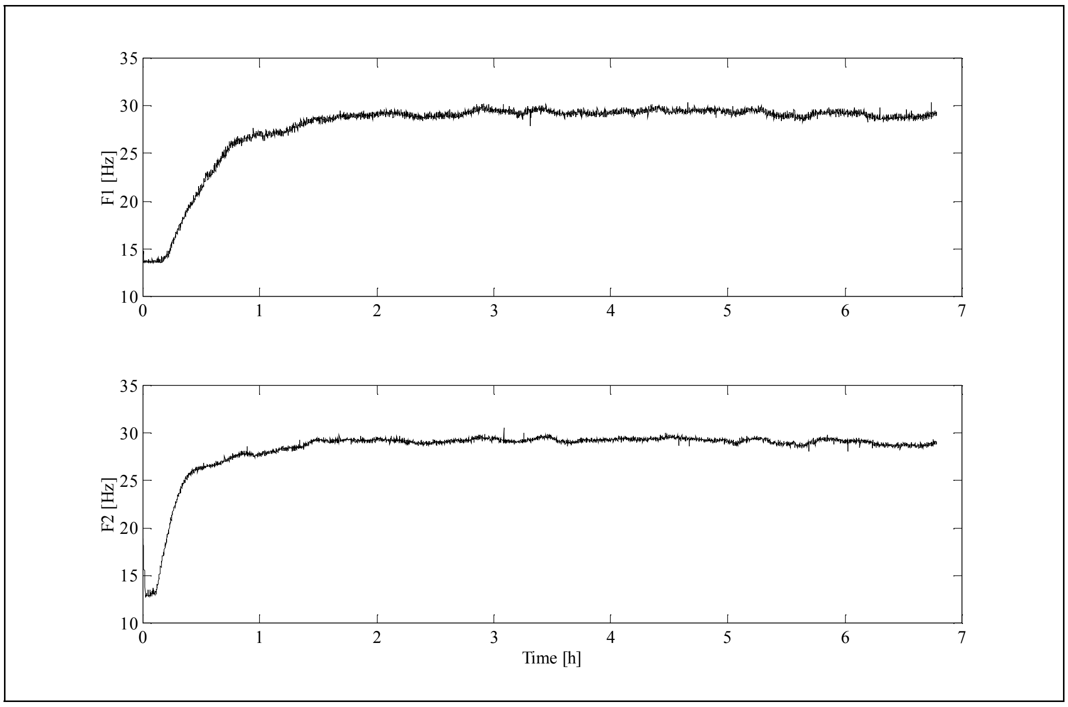


Figure 7-8. Manipulated variable control actions dictated by the PI temperature controllers.

As the operating cost of a batch distillation unit is determined largely by the energy consumption and therefore the batch time, control improvements could substantially reduce the unit cost per volume of product. A non-linear process model was constructed using SSA for neurocontroller developed with SMNE.

7.5 CLOSED LOOP IDENTIFICATION OF MEBAD COLUMN

Batch distillation is an inherently unsteady state process. The process' state variables cross a large region of the state space during operation. Obtaining sufficient dynamic information is thus more complex than for a steady state process, where operation is limited to a region around a set point. To ensure adequate dynamic information in the time series, the columns must be perturbed during the course of the batch, thereby exploring a wider region of the state space.

The PI control strategy by Wittgens and Skogestad (2000) was used to gather data for process identification. Each PI controller's integral action was changed to allow for a sample period of 5 [s], which allowed for pseudo open-loop step changes for the sample period. The process data was also logged at 5 [s] intervals. Sufficient dynamic information was obtained by perturbing both the temperature set points independently every 3 [min] during the course of the batch. The perturbation had a gaussian distribution around the nominal set point in each column, with a maximum deviation of ± 3 [°C]. Each new perturbation set points could generally be attained within 3 minutes. The stochastic nature of the set point perturbations introduced no determinism into the logged data. All the set point changes were thus independent of prior perturbations. The perturbations allowed data logging over a wide region of column temperatures and reflux rates, from which the column dynamics and the causal effect of reflux rates could be extracted. Figure 7-9 and Figure 7-10 show the variation in the temperature responses and reflux rates, introduced by the stochastic set point perturbations. Approximately 9000 data points were collected over a 12 hour period of operation, which was deemed sufficient for finding the determinism inherent in the system.

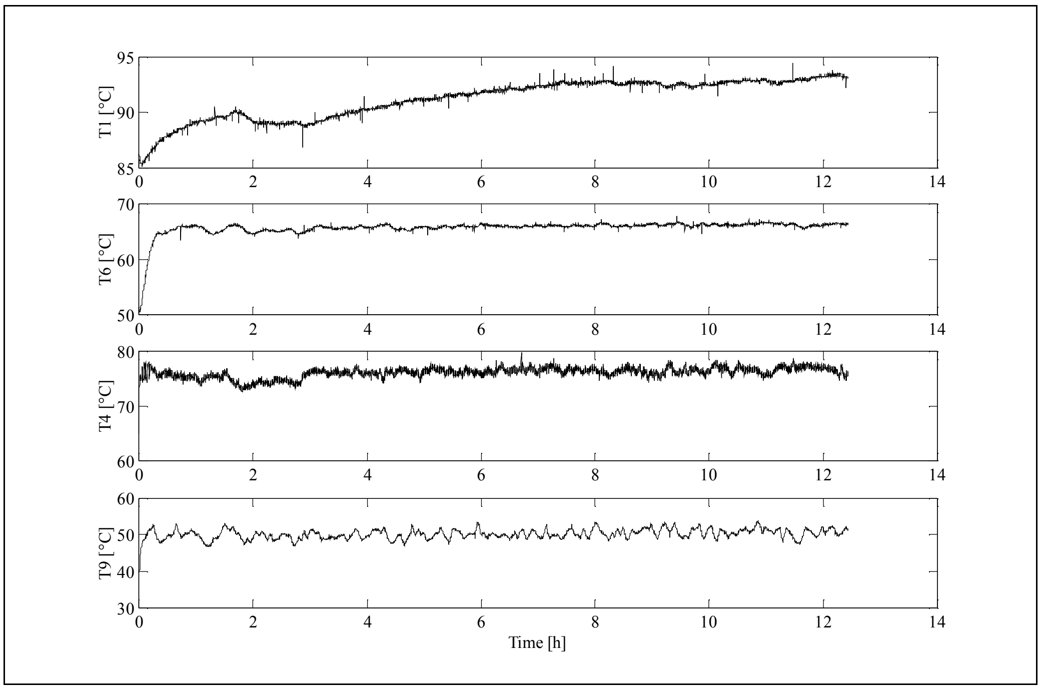


Figure 7-9. Process variable response owing to perturbations to the set points of the temperature controllers.

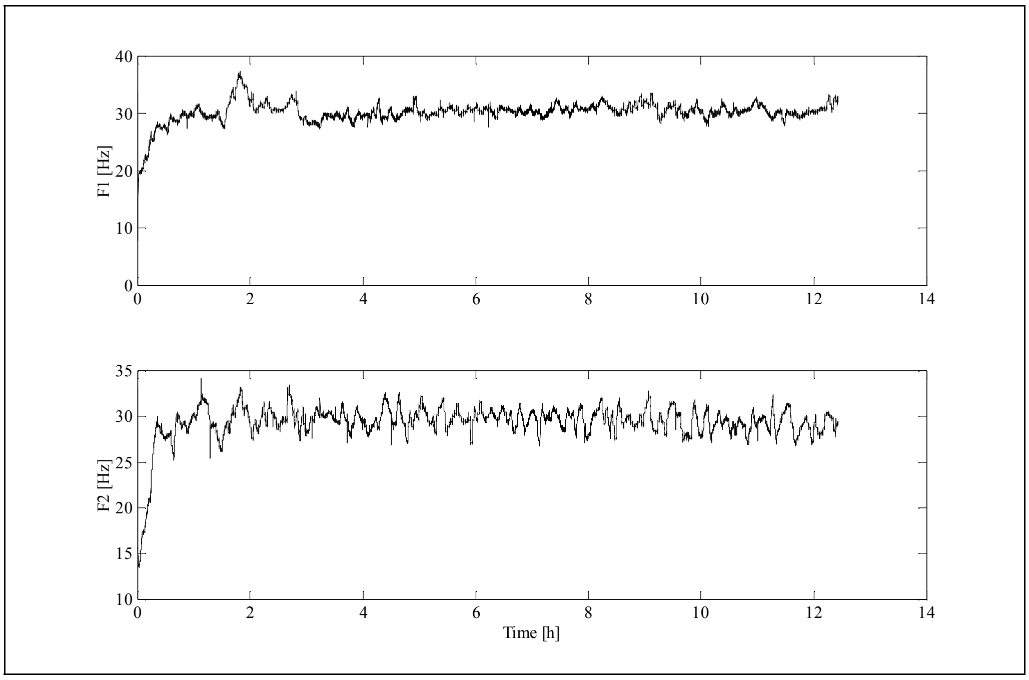


Figure 7-10. Manipulated control actions to accommodate the perturbations to the set points of the temperature controllers.

T1 and *T6* contained unique information regarding the slower temperature dynamics in the reboiler and the intermediate vessel. *T4* and *T9* contained information regarding the faster column dynamics. *F1* and *F2* formed the manipulated variable time series. The SSA implementation was thus multi-channel, whereby the state variables were extracted from the four temperature time series.

A delay time of 1 (i.e., $\tau_L = \tau_S$) captured the fastest dynamics in the system, since changes in the column temperatures generally required more than 5 [s]. The window length was selected using an estimated process time constant for temperature changes in the two columns (section 6.3). A window length $\tau_w = 10$ was deemed sufficient to capture the most significant column dynamics, since large temperature changes could generally be induced in 50 [s]. Therefore, the phase portrait is initially assumed to fit within a 10 dimensional state space. PCA of the lagged embedding matrix extracted the true state variable representation, with 4 state variables explaining more than 95 [%] of the variance in the multiple time series (section 6.3). The dominant column dynamics are thus described by 4 state variables, barring the state space reconstruction enforced by the non-stationary trend.

The 4 state variables and time lagged reflux rates (i.e., lagged over 10 sample periods) for *F1* and *F2* formed the 22 inputs to a feedforward neural network predictor (Figure 7-11a). *F1* and *F2* were lagged over 10 sample periods to allow for transport delays. A hidden layer with 10 sigmoidal neurons and 4 linear output neurons completed the one-step ahead predictor for *T1(t+1)*, *T4(t+1)*, *T6(t+1)* and *T9(t+1)*. Training comprised 80 [%] (i.e., randomised) of the available data and the remainder was used for validation. The neural network weights were trained using the Levenberg-Marquart algorithm. The R^2 statistics for the validation data are summarised in Table 7-2. The neural network model predicted accurately for a separate perturbation batch generated with different PI tuning parameters. The closed loop control thus had minimal impact on the identification of the open-loop plant model (section 6.3). A non-linear process model for the MEBAD column was thus created.

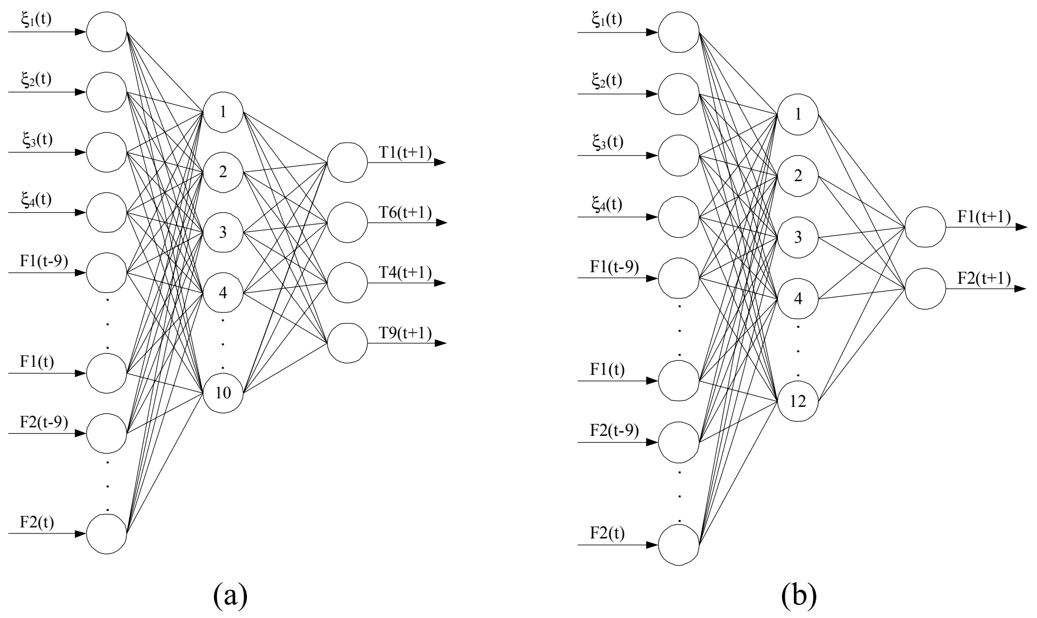


Figure 7-11. (a) Structure of neural network model and (b) structure of neurocontroller.

Table 7-2 - R^2 statistic for neural network model.

Predicted variable	*R^2 statistic*
T1(t+1)	0.99
T6(t+1)	0.99
T4(t+1)	0.84
T9(t+1)	0.99

Although the non-linear model predicts accurately and neural networks interpolate effectively, the extrapolation ability of the neural network may be poor. The model's validity is thus limited to the regions of the state space in which the data was gathered. PCA revealed that 74 [%] of the variance in the input vector space of the neural network model was explained by the first two principal components. The t-statistic for the first two principal components borders an elliptic region of the 2-dimensional input space as seen in Figure 7-12. This elliptic region is the 95 [%] confidence limit for the first two principal components of the model's input space. The model's validity, with some additional uncertainty in the unexplained variance, lies within this elliptic region (section 6.3). The model input vectors at the start of a batch fall outside this bounded region, since few data vectors are collected in regions of the state space that were crossed rapidly during the batch. At best, the model is thus a partial model of the MEBAD column, largely limited to regions of the state space where a large number of data vectors were collected. This knowledge of the model's validity is essential to developing the neurocontroller from the model. Figure 7-13 shows the batch trajectory for the total reflux control strategy by Wittgens & Skogestad (2000). A comparison of Figure 7-12 and Figure 7-13 demonstrates the exploration of the state space by perturbing the PI controller set points. This validity boundary ensures that SMNE searches regions of the input space for which minimal data have been collected.

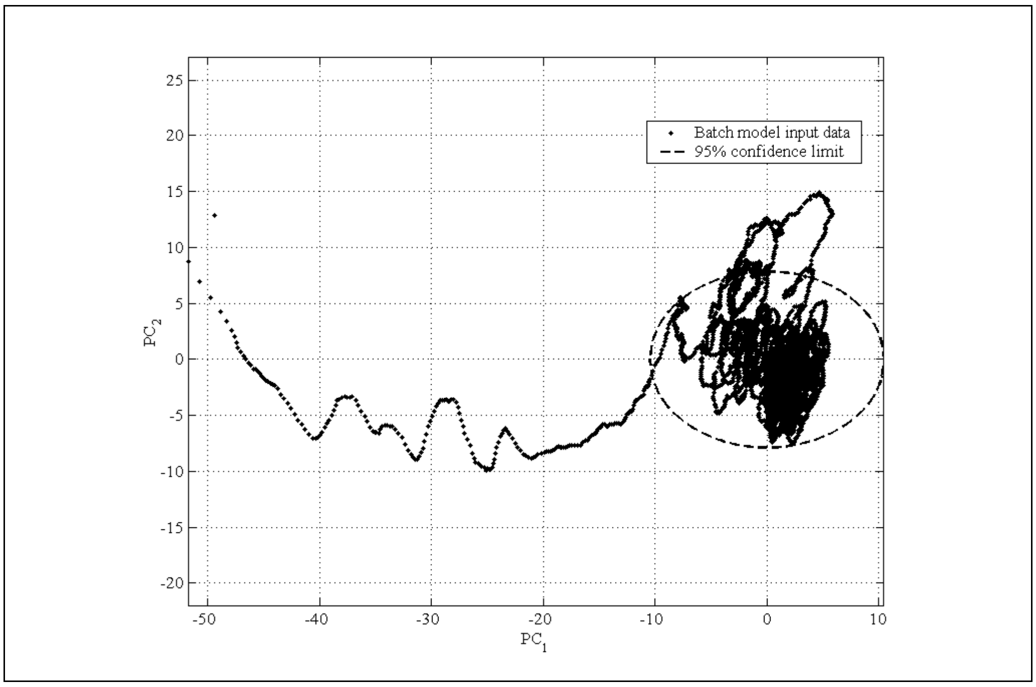


Figure 7-12 - Plot of first two principal components for the model input vectors, which shows the exploration of the state space by the perturbation batch. The initial condition for the model identification batch was (PC_1, PC_2) = (-50,10).

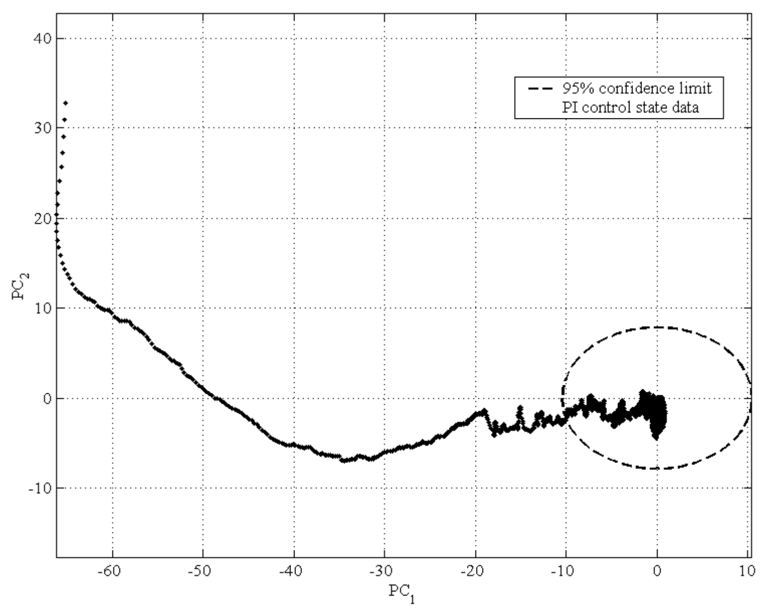


Figure 7-13 - The process trajectory for the first 7 [h] of the PI temperature control strategy. The initial condition for the total reflux batch was (PC$_1$, PC$_2$) = (-65, 35).

7.6 NEUROCONTROLLER DEVELOPMENT

Two experimental runs were required for neurocontroller development using SMNE. Like for the total reflux strategy, the column temperatures that corresponded with the desired product temperatures (i.e., compositions) needed to be determined. For a steady state reboiler temperature of 93 [°C] the temperature at *T4* was 76 [°C]. The steady state intermediate vessel temperature of 66 [°C] corresponded to *T9* at 50.5 [°C]. An additional PI control temperature perturbation run was necessary to determine the MEBAD column dynamics for model development (section 7.5). Given the neural network model developed with SSA and the steady state column temperatures, the SMNE algorithm was utilised to develop a neurocontroller for the MEBAD pilot plant.

Each neurocontroller in the SMNE genetic population was comprised of the same 22 inputs as in the SSA model, 10 sigmoidal hidden neurons and two sigmoidal output neurons that determined the reflux rates *F1(t+1)* and *F2(t+1)* (Figure 7-11b). For each reinforcement evaluation, the model was initialised randomly from the first 150 training data vectors obtained during the perturbation batch. Each neurocontroller evaluation was required to interact with the SSA model and learn to reach steady state from a typical initial condition within 3000 time steps. A reward was assigned to each new state s_{t+1} (i.e, each time step) based on the absolute error from the desired set points in the columns and the vessels. A smaller error thus corresponds to a larger reward. For linear systems, an integral-time-absolute-error (ITAE) response results in minimal overshoot. The absolute error from each set point, multiplied by the time *t* at each sample, was thus integrated over the specified 3000 samples. These integrals are minimised within the overall fitness function. Recall that the SSA model is strictly a partial model (section 7.5), therefore the reward at each time step was penalised in the overall fitness function when the model confidence limit was exceeded. The penalty per sample was large enough so that effective neurocontrollers would learn to remain within the confidence limit. A penalty was assigned to a sample when the first two principal components (i.e., the empirical orthogonal functions, $\overline{E}$) of a neurocontroller's input vector lay outside the elliptical region (i.e., *Ellipse* > 1):

$$Ellipse = \frac{E_1^{\ 2}}{\alpha^2} + \frac{E_2^{\ 2}}{\beta^2} \tag{7-2}$$

where α and β are the major and minor axis for the confidence limit ellipse respectively. The overall fitness function assigned a reward to each neurocontroller based on a simulation evaluation:

$$Fitness = \frac{1}{\left(\int_0^{3000} t\cdot\left|93 - T_1(t)\right|dt + \int_0^{3000} t\cdot\left|66 - T_6(t)\right|dt + \int_0^{3000} t\cdot\left|76 - T_4(t)\right|dt + \int_0^{3000} t\cdot\left|50.5 - T_9(t)\right|dt\right)} - Penalty \tag{7-3}$$

The fitness is maximised when the errors in the vessel and column set points are minimised in the shortest possible time. The implied driving force in the evolutionary search is thus to reduce the processing time to steady state. Since the MEBAD pilot plant was operated with constant heat duty, the fitness function thus related indirectly to minimising the mean energy consumption as in equation 7-1.

Figure 7-14 and Figure 7-15 plot the simulated temperature responses and the learned reflux rates for the best neurocontroller that SMNE developed from the SSA model. The process reached steady state in approximately 1.5 [h]. The errors at each sample after 1.5 [h] were thus near-zero. From Figure 7-14, the temperature response in the reboiler (i.e., *T1*) was overdamped as would be expected for the larger hold-up of the reboiler. The remaining temperature responses were slightly underdamped, giving an approximate ITAE (integral-time-absolute error) response. SMNE thus attempted to reproduce an ITAE response in each process variable, within the confines of maximising the overall fitness for each neurocontroller. Analogous to modern linear control pole placement, SMNE is thus an approximate means of non-linear pole placement for non-linear process models.

From Figure 7-12, it is evident that the initialisation for each neurocontroller evaluation lay outside the confidence limits for the model. A neurocontroller thus needed to establish reflux rates that drove the SSA model into the elliptic area as soon as possible (Figure 7-15). For the best neurocontroller, Figure 7-15c shows that the model's confidence limit was exceeded only at the beginning of an evaluation. Learning penalties were thus incurred only during the initial stage of the batch. Figure 7-16 shows the neurocontroller's input space, transformed by the same projection coefficient matrix obtained for model validity, during the batch. A similar plot to Figure 7-13 thus results, showing the batch trajectory proposed by the neurocontroller. A neurocontrol strategy was developed largely within the validity confines of the available process model.

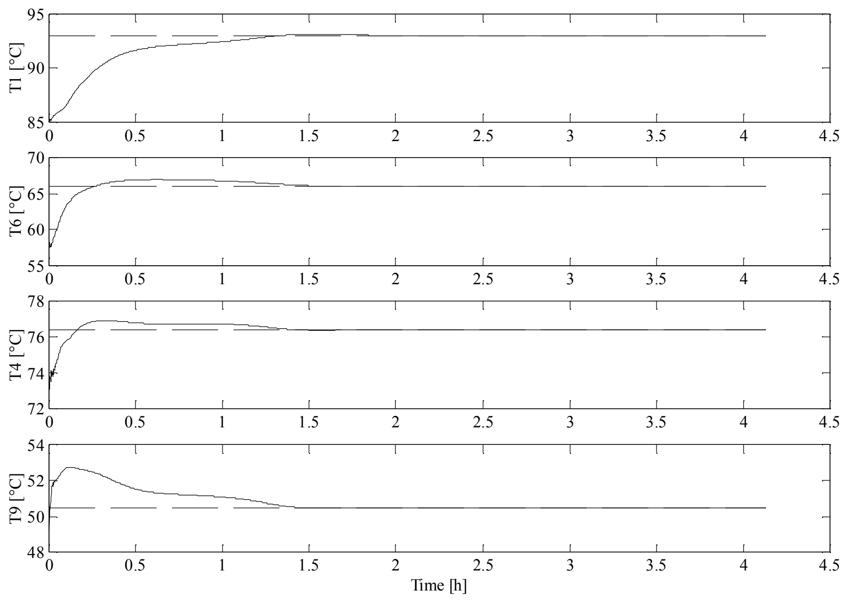


Figure 7-14. Determined process variable responses for the neurocontroller as learned from the neural network model

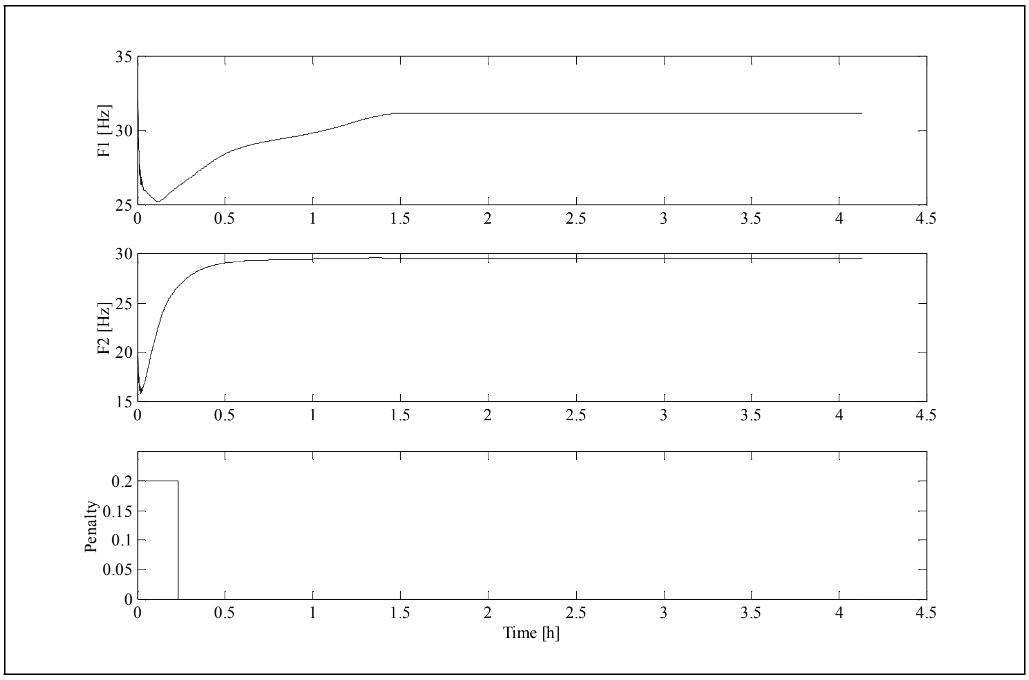


Figure 7-15. Manipulated variable control actions based on the response of the neural network model. The penalty assigned for exceeding the model validity boundaries is indicated in (c).

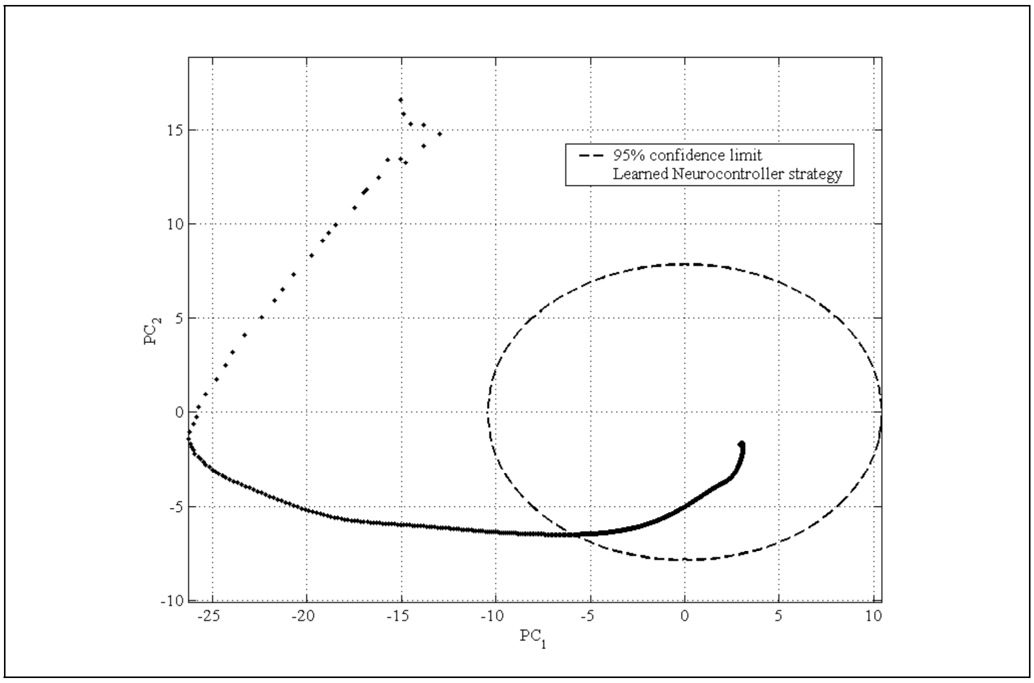


Figure 7-16. First two principal components of the neurocontroller input vectors during an evaluation of the best neurocontroller.

7.7 ON-LINE NEUROCONTROL

The developed neurocontroller was implemented on-line to control the MEBAD pilot plant as in Figure 7-17. The neurocontroller's sample rate conformed to the sample rate of 5 [s] in the SSA model. The state variables were calculated on-line using the projection coefficients matrix $\overline{PC}$ determined during model identification. This ensured that the neurocontroller inputs from the MEBAD sensors were transformed to state variables in the same manner as during reinforcement learning.

Given the partial nature of the SSA model, the neurocontrol strategy relied on a similar heat duty as in the model identification batch. Therefore, a constant heat duty was maintained at the same set point as during the model identification batch. The neurocontroller was thus only required to reject minor disturbances in heat duty. Clearly, the SSA model is specific to a ternary system of n-pentane, n-hexane and n-heptane. However, the composition and volume of the initial charge was substantially different from the initial charge composition and volume used in the model identification batch. The neurocontroller would thus be confronted with an initial condition substantially different from the initial condition during the reinforcement learning process. Sensor noise, absent in the SSA model, would also test the robust performance of the neurocontroller. Experimental validation thus established whether the neurocontroller generalised to process uncertainty; only relying on the calculated state variable representations and past reflux rates to control the pilot plant.

The settling time to steady state for the on-line neurocontroller response (Figure 7-18) is considerably longer than for the simulated neurocontroller (Figure 7-14). This was due to the larger volume charged to the reboiler. The on-line temperature responses in Figure 7-18 and the simulated temperature responses in Figure 7-14 are also dissimilar. *T1*'s on-line response is slightly underdamped, which is contrary to the overdamped response in the simulated neurocontrol response. This may be expected due to the substantial difference in composition and volume of the reboiler charge. The other on-line temperature responses are similar to the simulated controller responses. The reflux rate policies in column A and column B (Figure 7-20) are also similar to the learned responses (Figure 7-8), except that the on-line control actions overshot the final reflux rates whereas the simulated reflux control actions increased gradually to the final reflux rates. These discrepancies between the simulated and on-line neurocontrol actions are due to plant/model mismatch. Markedly, the on-line hold-up profiles (Figure 7-19) were different than for total reflux control (Figure 7-4), particularly in *L3*. The robust performance attests to the generalisation of the neurocontroller. Neurocontrol established the steady state between 6 [h] and 7 [h], which resulted in a batch time considerably shorter than for total reflux control

(Figure 7-6). The neurocontrol strategy thus established more optimal temperature and hold-up profiles than PI control.

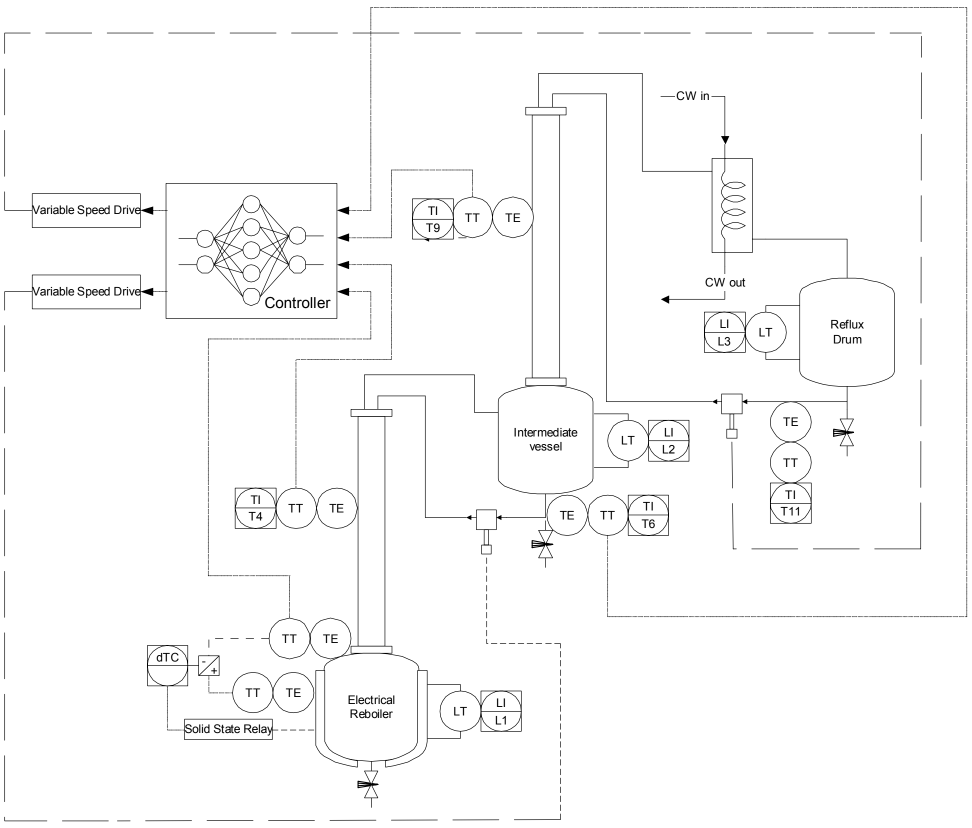


Figure 7-17. Neurocontrol scheme for MEBAD pilot plant.

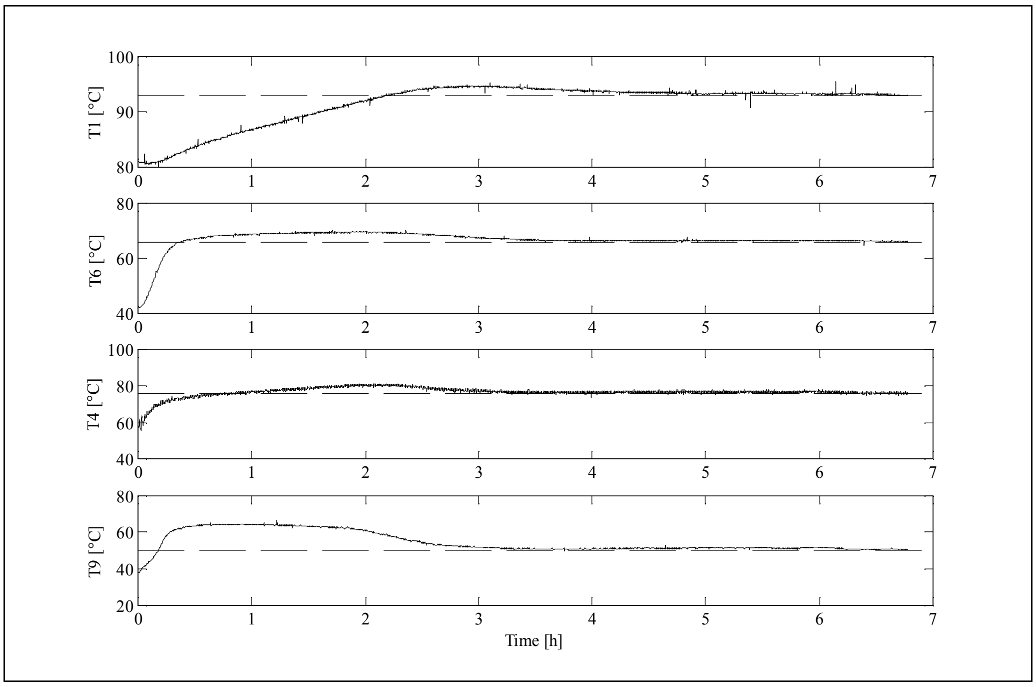


Figure 7-18. Process variable responses for SMNE neurocontrol strategy.

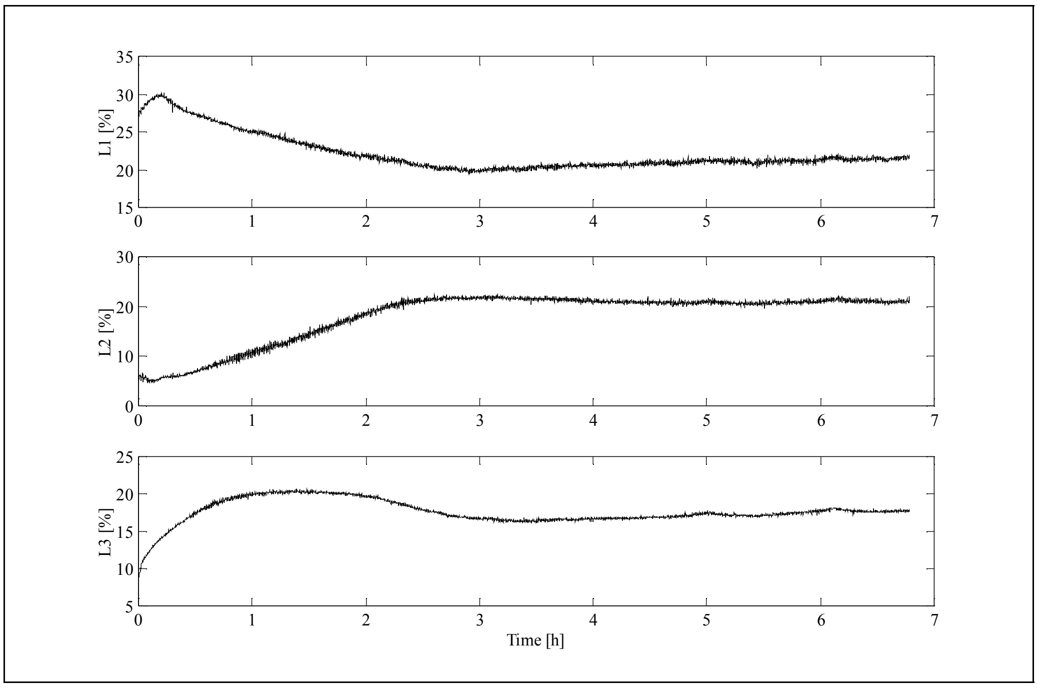


Figure 7-19. Vessel level responses for SMNE neurocontrol strategy.

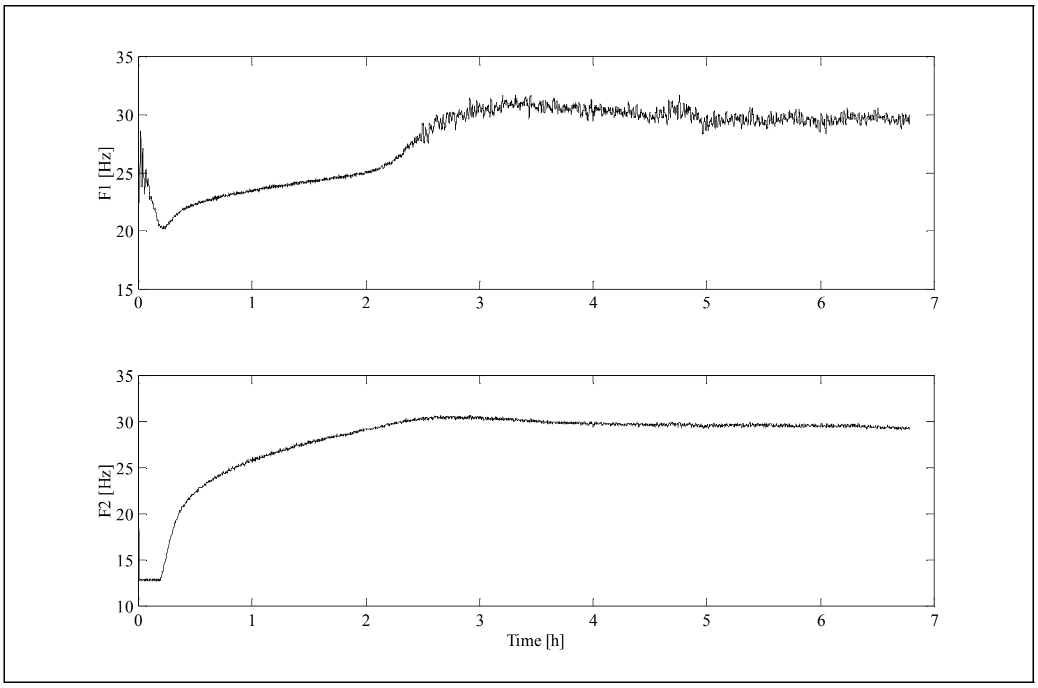


Figure 7-20. Manipulated variable control actions for SMNE neurocontrol strategy.

Figure 7-21 shows the input vectors to the neurocontroller during the on-line batch, transformed with the same projection coefficients matrix determined for model validity. The ellipse in Figure 7-21 reflects the region in which the model is considered valid. The neurocontroller operated the MEBAD pilot plant outside the region of model validity for a significant period of time. The initial condition for the neurocontrol batch (Figure 7-21) is similar to the total reflux batch (Figure 7-13), but vastly different to the initial condition for the model identification batch (Figure

7-12). The number of sample periods spent outside the region of model validity reflected the greater batch time required to separate a larger initial volume. The neurocontroller's robust performance, despite extensive operation outside the model validity, also reflects the generalisation of the neurocontroller to uncertain process conditions.

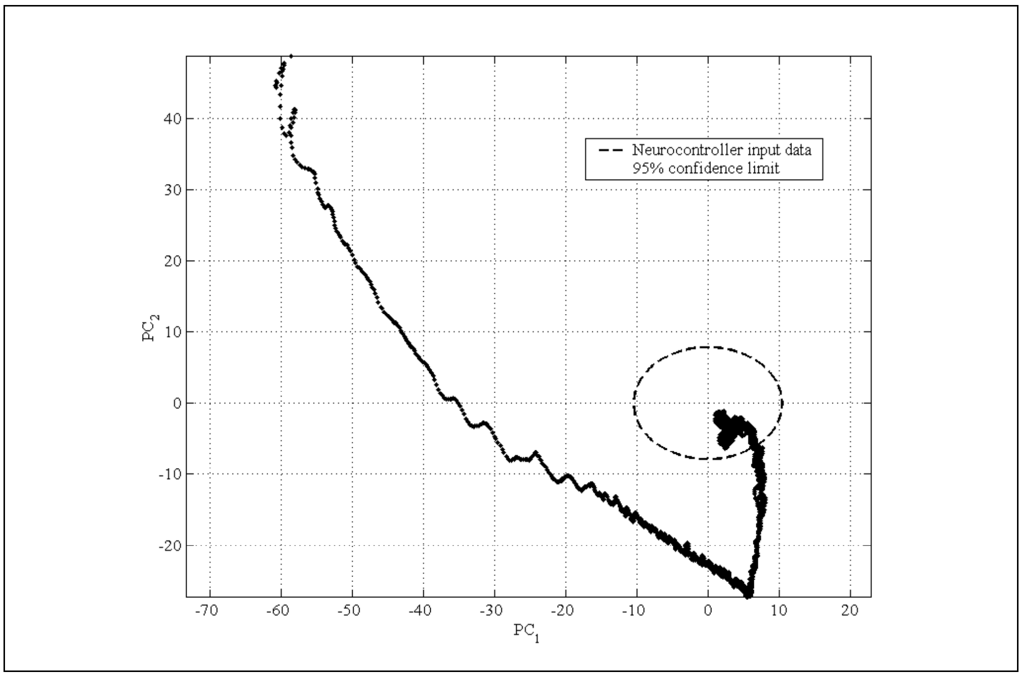


Figure 7-21- First two principal components of the neurocontroller input vectors during on-line control of the MEBAD pilot plant. The initial condition for the on-line neurocontrol was (PC_1, PC_2) = (-60,50).

7.8 DISCUSSION

Sections 7.8.1 and 7.8.2 compare the total reflux control strategy in section 7.4.1 (Wittgens & Skogestad, 2000) and the on-line neurocontrol strategy in section 7.7.

7.8.1 Total reflux PI control

The PI control strategy solves the control task by maintaining constant temperature set points at a single location in each column. The batch terminates when the steady state temperature in each vessel is reached at total reflux. A batch was started at total reflux. With the entire charge in the reboiler, total reflux prevented the accumulation of product in the intermediate vessel and the reflux drum. After activating PI control, the PI controllers set $(L/V) << 1$ in both columns, allowing the temperatures in the columns to rise to set point (Figure 7-6) and for liquid to accumulate in the vessels (Figure 7-7). As soon as the set points were reached, constant temperature control was established near total reflux, which provided for maximum fractionation in each column (Figure 7-8). With the intermediate vessel and the reflux drum not at their final hold-ups, continuous total reflux operation would prevent further component separation. However, the initial hold-ups in the intermediate and reflux drum vessels did not have the desired product composition. As the hold-up in each vessel is turned over, while maintaining constant column temperatures, each vessel's hold-up is progressively purified. The temperature at the top of each column is highly dependent on the composition in the vessel from which that column's reflux is pumped. At near total reflux, these vessel compositions became inconsistent thermodynamically with the temperature set points in the columns. The PI controllers thus momentarily suspend near total reflux, so that the column temperatures don't drift from the set points. A $(L/V) < 1$ reduced the reboiler hold-up and distributed this vapour as liquid hold-up between the intermediate vessel and the reflux drum.

As the vessel compositions became inconsistent with the column temperature set points, near total reflux is re-established. This slowed an increase in the vessel hold-ups, while the volume turn-over purified each vessel towards steady state. This process is repeated on a micro-scale throughout the batch. Cyclic operation between $(L/V) \approx 1$ (i.e., purification) and $(L/V) < 1$ (i.e., rectification) thus ensued, which gradually reduced the reboiler hold-up. The larger the hold-up in the intermediate and reflux drum vessels, the greater the turn-over (i.e., purification) time to optimally distribute the components in the vessels.

Figure 7-6 to Figure 7-8 reveal this PI control strategy throughout the batch. At 3 [h], near total reflux was established in both columns (Figure 7-8). Even though *L2* and *L3* are below their final hold-up volumes (Figure 7-7), the PI controllers maintain *T5* and *T10* at constant temperature (Figure 7-6) thereby approximating total reflux operation. The constant purification of the vessel compositions at pseudo-constant hold-up, required the PI controllers to abandon purification for rectification, which ensured that the hold-ups *L2* and *L3* could increase.

The slow settling time in *T1* attested to significant amounts of batch time spent purifying large vessel hold-ups at near total reflux, resulting in a slow increase to the intermediate vessel hold-up. The process interaction between the columns also delays effective redistribution of the liquid hold-up at near total reflux (section 7.4). Ideally, near total reflux should be attained only once the final vessel hold-ups have been reached. The temperature profiles and vessel hold-ups should thus be controlled so that total reflux coincides closely with attaining the final product compositions and vessel hold-ups. More importantly, PI temperature control only uses a single column to separate two components, whereas a more optimal batch trajectory may utilise more than one column (i.e., more theoretical stags) to separate two components. The slow purification in the reboiler, is indicative of slow rectification owing to limited theoretical stages. Constant temperature set point and the pure feedback nature of PI temperature control was thus detrimental to optimising the batch time. For PI temperature control, the vessel hold-ups can only change once the compositions changes in the intermediate vessels and an error from set point is perceived by the temperature controllers. A combined feedforward and feedback strategy should be used to optimise the (L/V) ratio in each column as in the V-policy used by Noda et al. (2000) in section 7.2.1.

7.8.2 SMNE Neurocontrol

The SMNE neurocontrol strategy is such a combined feedforward and feedback strategy. Neurocontroller development with a partial process model included knowledge of process dynamics as feedforward elements into the network structure, whilst the state inputs provided corrective feedback to the control strategy. This dual strategy suggested a mild cyclic operation of the MEBAD unit, as seen in the temperature responses in Figure 7-18. For the time interval between 0 - 1 [h] in column B the neurocontroller set *F2* to a (L/V) << 1, which caused *T9* to overshoot its set point significantly. This implies that excess n-hexane was allowed to accumulate in the reflux drum, which is reflected by the rapid increase in *L3*. Between 0 - 2 [h] in column A, *F1* also had a (L/V) < 1 with *T4* rising sharply with slight overshoot of the set point in column A. Excess heavy components began to accumulate in the intermediate vessel, as *T6* overshot 66 [°C] and *L2* increased sharply. The low reflux rates in both columns thus removed rapidly the light components from the reboiler, which is evident in the decrease in *L1* (Figure 7-19) and the increase in *T1* (Figure 7-18). Between 1-2 [h], the neurocontroller set *F2* to a (L/V) > 1 which dumped the excess n-hexane in *L3* into *L2*, corresponding to a decrease in *T9* and the decrease in *L3*. The accumulation of light components in the intermediate vessel, caused *T6* to decrease between 2-3 [h] and approach its set point. Total reflux rate was reached in column B at 3.5 [h] with *T9* and *L3* at steady state. The slow increase in column A's (L/V) ratio from 0 - 2 [h] allowed *T1* to overshoot its set point, with *T4* remaining above its set points. The increase in lights in the intermediate vessel and the sharp increase in *F1* between 2 - 3 [h] caused *T4* to decease towards its set point. At approximately 3 [h], the neurocontroller set (L/V) > 1 in column A and dumped the contents in *L2* to *L1*, thus decreasing *T1* and increasing *L1*.

Initially, n-pentane and n-hexane was separated in both columns, thus utilising the maximum number of stages for this initial separation. This aided in shortening the processing time. Both columns are thus used initially in rectification. The overshoot in the steady state hold-ups in the reflux and intermediate vessel indicates that both columns were utilised as stripper columns later in the batch. SMNE determined this two-step cyclic operation neurocontrol policy from the SSA model implicitly, as the strategy is evident in the simulated neurocontrol overshoot responses for *T4*, *T6* and *T9* (Figure 7-14). However, though the neurocontrol policy proved more effective than PI control, it remained sub-optimal as seen in a comparison of the batch trajectories of Figure 7-16 and Figure 7-21.

When the neurocontrol batch trajectory reached (PC_1, PC_2) = (-20,-10), the trajectory appeared off course. At 2 [h], the batch trajectory reached (PC_1, PC_2) = (5,-30) in Figure 7-21 and changed direction dramatically. This coincided with a greater sensitivity in *F1*'s response to the neurocontroller inputs in Figure 7-20, where *F1* started to increase sharply and *T1* overshot its set point. This indicates that a large number of the 10 hidden neurons mapped to regions of the state space encountered after this time in the batch. From (-60,50) to (5,-30) in Figure 7-21, the neurocontroller's performance is robust (i.e., the control actions are appropriate), but sub-optimal due to minimal mapping of these regions of the input space. Inadequate process model information was thus available in these regions of the state space for model identification. At (5,-30) the control actions became near-optimal until steady state was reached. The neurocontroller's trajectory in the input space lay well beyond the 95 [%] confidence limit, attesting to significant generalisation.

7.9 CONCLUSIONS

Wittgens and Skogestad (2000) suggested a multi-loop SISO linear control policy for the non-linear MEBAD control. A MEBAD pilot plant confirmed that the pure feedback PI control strategy established sub-optimal temperature and hold-up profiles in the columns and vessels, concurring with Furlonge et al. (1999). A combined feedforward and feedback control policy is preferable. In addition, a non-linear MIMO control policy that eliminates process interaction between controller pairings is desirable. Incorporating SSA for model identification and SMNE for neurocontrol development, efficient and practical non-linear control is achieved. SSA recreated a state variable representation for the MEBAD pilot plant from limited input-output plant data and created a one-step ahead neural network model. SMNE used this partial model of the MEBAD process to learn near-optimal temperature profiles in the columns and vessels without needing expensive analysers. SMNE included process knowledge from the process model into the neurocontroller, thereby including feedforward elements in the neurocontroller structure. The state variable inputs to the neurocontroller incorporated feedback elements into the non-linear control strategy. Experiments confirmed that the neurocontrol strategy is robust and established the product compositions from any initial composition or volume in the reboiler. Significant generalisation was evident, as the neurocontroller dealt effectively with process uncertainty during operation in regions not mapped effectively within the SSA model. Near-optimal control resulted once the process trajectory moved into regions mapped by the SSA model. As a result of near optimal performance, the batch time for neurocontrol was significantly less than for PI temperature control.

Batch process control is far more complex than continuous process control and more difficult to implement within an advanced control framework (section 7.1). Batch dynamic modelling remains a bottleneck in model-based control schemes. SSA bridges the gap to batch modelling, though large process data sets are clearly desirable. SSA is better suited to continuous processes where control is limited to a smaller region of the state space than encountered in batch processing. Nevertheless, SMNE complemented with SSA models in a general neurocontrol paradigm, finds particular utility where non-linear processes are poorly understood and reliable fundamental models have been difficult to attain.

Future work will include variable heat duty as a manipulated variable for the MEBAD pilot plant, thus using additional degrees of freedom to improve control.

7.10 SYMBOLS FOR CHAPTER 7

Symbol	*Description*	*Unit*
a	interfacial area per unit volume packing	[dm^{-1}]
B	number of components for batch separation	[-]
C	number principal components > 95% variance	[-]
d	process disturbance	[-]
e	error	[-]
E	empirical orthogonal functions	[-]
f	function	[-]
g	function	[-]
G	liquid volume hold-up	[dm^3]
h	function	[-]
H	liquid mass hold-up	[kg]
I	liquid enthalpy	[$kJ\cdot kg^{-1}$]
J	vapour enthalpy	[$kJ\cdot kg^{-1}$]
k	overall mass transfer coefficient	[$m^{-1}\cdot min^{-1}$]
K	equilibrium constant	[-]
L	liquid flow rate	[$kg\cdot min^{-1}$]
M	number of samples in window length	[-]
n	sample number	[-]
N	number of data samples	[-]
P	pressure	[kPa]
PC	principal components	[-]

r	reinforcement learning reward	[-]
S	cross-section area	[dm^2]
t	time	[min]
t_f	batch processing time	[min]
t_s	batch set-up time	[min]
T	temperature	[°C]
u	manipulated variable	[-]
V	vapour flow rate	[$kg \cdot min^{-1}$]
x	liquid phase mass fraction	[-]
X	lagged-covariance matrix	[-]
y	vapour phase mass fraction	[-]

Greek symbols

Symbol	*Description*	*Unit*
α	ellipse major axis	[-]
β	ellipse minor axis	[-]
ε	mean energy consumption	[kW]
ρ	liquid density	[$kg \cdot dm^{-3}$]
τ_s	sample period	[min]
τ_L	lag time	[min]
τ_w	window length	[min]
υ	manipulated variable	[-]
ξ	state variable	[-]
ψ	process variable	[-]

Subscripts

Symbol	*Description*
i	component
j	unit volume of structured packing
k	state variable number

Superscript

Symbol	*Description*
*	equilibrium
m	model
p	process
SP	set point

8 PLANT-WIDE CONTROL OF THE TENNESSEE EASTMAN CONTROL CHALLENGE

OBJECTIVES OF CHAPTER 8
• Demonstrate plant-wide control with high dimensionality using a benchmark simulation model.   • Contrast the SMNE approach to existing plant-wide control approaches.

8.1 INTRODUCTION

Plant-wide control seeks a systematic approach to identifying superior control architectures for processes with high dimensionality (chapter 2). Nevertheless, high dimensionality is frequently debilitating to plant-wide control methodologies. For example, a plant with 10 manipulated variables and 10 process variables has 184755 possible square control systems. Incorporating the number of possible feedback connections, a total of $9.73 \cdot 10^{13}$ candidate configurations exist, assuming SISO interconnections (Banerjee & Arkun, 1995).

A number of benchmark simulation models exist for testing plant-wide control methodologies. The most prominent are the Tennessee Eastman (TE) control challenge, a vinyl acetate monomer process, a HDA plant and the Luyben & Luyben plant (Robinson et al., 2001). Of these available benchmarks, the TE model is a truly significant plant-wide control problem, incorporating a large number of interacting process and manipulated variables. The process model was developed on an actual industrial process so that the simulation closely approximates what may be expected in reality. The Tennessee Eastman control challenge offers numerous opportunities for control study purposes, of which the exploration of multivariate control, optimisation and non-linear control are most pertinent to this case study.

8.2 TENNESSEE EASTMAN SIMULATION MODEL

The Tennessee Eastman control challenge involves the control of five unit operations: (1) an exothermic 2-phase reactor, (2) a water-cooled condenser, (3) a centrifugal compressor, (4) a flash drum and (5) a reboiler stripper. The simulated plant has 41 process variables and 12 manipulated variables as illustrated in Figure 8-1, which are modelled with 50 state variables. In this case study the control objective involved keeping the product compositions within a desired range, while maintaining maximum production at minimal production cost (i.e., maximised contribution). This objective needed to be accomplished, while keeping the reactor pressure, reactor temperature and vessel levels within specified process constraints to avoid shutdown of production.

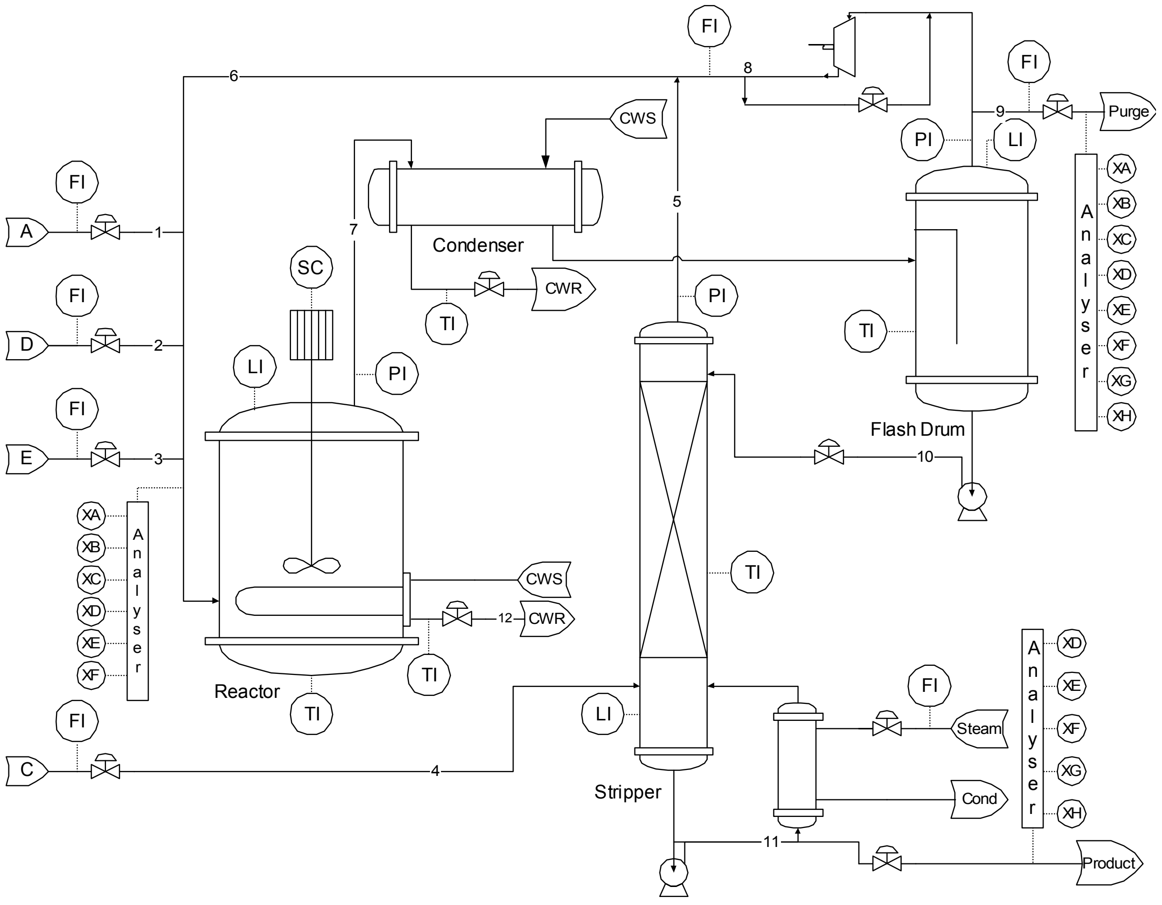


Figure 8-1 - Tennessee Eastman Control Challenge Process Flow Diagram.

The twelve manipulated variables (Figure 8-1) are the four feed rates, the purge rate, the agitation rate, steam rate, condenser coolant rate, reactor coolant rate, compressor recycle, flash drum discharge rate and the stripper production rate. The 41 process variables include level, pressure, temperature, flow and composition indicators as illustrated in Figure 8-1 (Downs and Vogel, 1993).

The chemical reactions are irreversible and occur in the vapour space of the reactor. The chemical reactions are: $A(g) + C(g) + D(g) \rightarrow G(l)$; $A(g) + C(g) + E(g) \rightarrow H(l)$; $A(g) + E(g) \rightarrow F(l)$ and $3D(g) \rightarrow 2F(l)$. The formation of an inert byproduct, *F*, is determined by kinetic selectivity. The products *G* and *H* accumulate in the reactor and the reactor level must be controlled by equating the production rate to the vapour removal rate. Product may thus only be removed via the vapour stream to the condenser. The rate of exothermic heat removal is controlled by the agitation speed and the cooling water flow rate to the cooling coils in the reactor. Should the liquid level in the reactor fall below 50 [%], the loss of heat-transfer surface area for cooling becomes pronounced. For the reactor pressure, Ricker (1995) has indicated that the optimal steady states for the various operating modes are in near proximity to the upper shutdown limit of 3000 [kPa]. The ability to operate in close proximity to the upper shutdown limit is determined by the ability of the controller to maintain high performance in the presence of disturbances.

The vapour discharge from the reactor is fed to a partial condenser (Figure 8-1). The flash drum serves to separate the liquid and vapour phase fed from the condenser. The liquid fraction is fed to the reboiled stripper and the vapour fraction is returned to the reactor via a centrifugal compressor as a recycle stream. The liquid feed to the stripper is distilled to remove impurities in the bottoms product and the vapour stream is recycled to the reactor. A purge is necessary to prevent the accumulation of the inert *B* that is present in *A*+*C* feed. The compressor recycle valve adjusts of the net feed rate to the reactor.

In this case study, the three desired set points are 50 [%] *G* and 50 [%] *H*, 90 [%] *G* and 10 [%] *H*, 10 [%] *G* and 90 [%] *H* on a mass basis. The maximum production rate must be maintained at minimal variable cost of production (VCOP). Byproduct *F* may be present in the product, provided 97.5 [%] of the product is composed of *G* and *H*. The product purity of either *G* or *H* must remain within 5 [%] of the desired set point.

Highly interactive relationships exist between reactor temperature, reactor pressure and reaction rate. The chemical reaction reduces the number of moles, which tends to decrease the pressure as the reaction rate increases. The reaction rate is determined by Arrhenius temperature dependence and is approximately third-order in the reaction pressure. The liquid-vapour equilibrium also contributes to strong interactions. These interactions could destabilise the plant should incorrect control actions be taken (Price et al., 1994). The model contains both integrating (i.e., vessel levels) and self-regulatory (i.e., plant pressures) subsystems. Also, the presence of a recycle stream compounds the control problem. The recycle flow makes up 64 [%] (m/m) of the feed to the reactor at the design conditions. Owing to the large recycle, slight variations in the reactor operating conditions may be amplified and returned to the reactor (Price et al., 1994). The model simulates a wide range of disturbances, from sticking valves to random process upsets to the loss of key feed steams. Plant dynamic behaviour is also extended to control valves, in that control valves also have a transient response. The process variables are comprised of continuous variables (i.e., temperatures, levels and pressures) and discrete variables (i.e., analysers' outputs) with different sample periods (Downs and Vogel, 1993).

Of greater interest than the shear scope of the TE control problem, is the opportunity that it offers to compare various plant-wide control methodologies. Various approaches have been considered in past work, such as multi-loop SISO control strategies, dynamic matrix control (DMC) and linear/non-linear MPC. Section 8.3 describes and analyses both decentralised and centralised control methodologies that have been considered by other researchers.

8.3 PRIOR TENNESSEE EASTMAN CONTROL SOLUTIONS

8.3.1 Decentralised control

This chapter focuses on the development of a centralised control system design, which is in sharp contrast to the decentralised approach taken in multi-loop SISO control designs (see section 2.1.1). In a multi-loop SISO design the problem needs to be decomposed into a number of design stages to make the design process manageable. For example, the optimal steady state for each operating mode needs to be located independently of the controller design. This requires solving a non-linear programming problem. Ricker (1995) used an augmented Lagrangian strategy for locating the optimal steady states. The design approach also requires a degree of engineering judgement that results from experience with pairing process and manipulated variables. Intimate knowledge of the plant dynamics and relative proportions of expected flow rates and rates of change in process variables is required before any such design may be undertaken. For example, general consensus on the best throughput manipulator for the TE control challenge has not been forthcoming. A large number of different controllability analysis techniques need to be utilised. Moreover, loop tuning and the selection of appropriate process-manipulated variable pairings need to be considered in the presence and absence of noise and disturbances (McAvoy & Ye, 1994).

For a multi-loop SISO design, several iterations on the design procedure may thus be required. Control loops may also need to be tailored to deal with expected or known disturbances. This gives no guarantee of the control system's performance in the presence of unknown or unexpected disturbances. Obtaining robust control by detuning generally results in a significant loss of loop performance. Further, only PI controllers are used throughout the design process. Although PI control is appropriate for a large proportion of control problems, severe non-linearities around set points may significantly degrade the performance of linear controllers. The tuning parameters may thus only be appropriate over a limited range of the desired operating range.

Recall from section 2.6 that McAvoy & Ye (1994) proposed a plant-wide control methodology based on the time-scales of certain process variables. First, fast inner loops (e.g., flow rates and temperatures) were closed based on process judgement, which allowed local disturbances to be rejected effectively. Second, McAvoy and Ye (1994) sought to reduce the maximum 12x12 controller design using a wide range of control analysis techniques, viz. Bristol's relative gain array, the Niederlinski Index and linear saturation analysis, non-linear disturbance and saturation analysis and dynamic simulation. Third, McAvoy and Ye (1994) configured the analyser loops for product composition control and throughput requirements. Fourth, PI controller tuning was accomplished by tuning the inner loops first followed by the outer loops on a trial-and-error basis. All the requirements set by Downs and Vogel (1992) were met, although the loss of feed A needed special provision to maintain pressure control. McAvoy (1999) proposed a different approach to plant-wide control synthesis that uses steady-state models and optimisation. Optimisation involved a mixed-integer linear programming (MILP) problem that minimises the absolute value of valve movements when a disturbance occurs. A system that requires large valve changes is deemed inferior to a system that requires smaller valve changes. The first such optimisation identifies candidate control architectures for controlled variables that must be held constant for safety and other reasons. These candidate control structures are screened using controllability tools. Similarly, candidate control architectures are identified for throughput and purity control using the MILP approach. Notable differences exist between the plant-wide control strategy proposed in McAvoy & Ye (1994) and that proposed by McAvoy (1999). Most notably, in McAvoy & Ye (1994) the reactor pressure is controlled by the flow rate of reagent *A*, whereas in McAvoy (1999) the reactor pressure is controlled by nested loops that finally control the reaction temperature. The optimisation analysis of McAvoy (1999) determined that the feed rate of reactant *C* controls the throughput of the TE plant.

Price et al. (1994) provided plant-wide guidelines for throughput and inventory control in selecting candidate control architectures. Price et al. (1994) emphasised that decisions during the development of the throughput and inventory control structures impact the overall performance of the plant-wide control system. Internal throughput manipulators typically provide superior performance, particularly when the throughput manipulator is located near the centre of the process flow path. Consequently, production rate changes propagate in both directions speedily (i.e., self-consistent control strategies). The best throughput/inventory control structure resulted when the production rate was controlled by manipulating the duty of the reactor condenser, thereby adjusting the separation rate of product from the recycle (Price et al., 1994). Lyman and Georgakis (1995) proposed the same control strategy as Price et al. (1994).

Banerjee & Arkun (1995) decomposed the design problem into two tiers. The first tier controlled the critical variables that influenced the reactor stability. The second tier prioritised variables that affected the compositions of inlet and outlet streams to the reactor. The throughput was manipulated based on the participation of reactants *D* and *E* in the formation of *G* and *H* respectively. Since reactant *D* is required to form *G*, the feed rate of *D* was manipulated to control the mass flow rate of *G* in the product. Similarly, reactant *E* is required to form *H* and the mass flow rate of *H* was controlled by manipulating the feed rate of reactant *E*. This strategy required that the mole fraction of both *C* and *A* to the reactor feed remained constant using the respective feed, so as not to affect the reaction kinetics and consequently the composition and throughput of the product. The accumulation of by-product, *F*, was controlled using the purge valve. Both Price et al. (1994) and McAvoy (1999) used the purge valve to regulate the accumulation of by-product, *B*, in the purge.

Ricker (1996) commented on the work by Banerjee & Arkun (1995) and McAvoy & Ye (1994), noting that heuristics needed to be employed, despite a quantitative approach. Also, direct numerical linearisation of the Downs & Vogel FORTRAN code, created unrealistically good models for quantitative analysis. Ricker (1996) followed a more industrial approach akin to that of Price et al. (1994). The TE process has 12 degrees of freedom. Six measured variables are paramount to the control objectives and need to be controlled at an optimum, viz. production rate, mole % in underflow, reactor pressure, reactor liquid level, separator liquid level and stripper liquid level. Setting the agitator at the maximum agitation speed, five degrees of freedom remain for economic optimisation. The design was structured around the throughput manipulator. Ricker assigned the variable most likely to impact on steady-state operation for production rate control. Noting that plants are typically operated at full capacity, the feed rates of reagents *D* and *E* most likely constrain the production rate. Ricker (1996) recommended that the ratio of *D*/*E* control the throughput. Inventory control was developed around this ratio control strategy. A simple algorithm identified the active constraint on production rate, whereupon the composition was controlled at the active constraint. Simple static feedforward control provided excellent product composition and throughput control. Since the purge rate has a significant effect on the operating cost, Ricker (1996) proposed that the reactor pressure be controlled using the purge. Typically, the purge has been used to control the inert mole % of by-product *B*. A poor set point choice for the mole % of *B* in the purge could severely hamper the operating cost. By using the purge for pressure control, Ricker (1996) thus minimised the operating costs implicitly.

Luyben et al. (1997) applied a process-orientated approach. Depending on the operating objective, the throughput was set either using the stripper bottoms product flow rate or a reactant feed rate, leading to completely different control strategies. As for most decentralised control strategies, overrides were incorporated to deal with special operating circumstances. Luyben et al. (1997) presented no dynamic simulation verification of the proposed control system.

Larsson et al. (2001) considered decentralised control at the base case. Contrary to most other methodologies, the design was driven by steady-state economic considerations. Using the concept of self-optimisation (see section 2.4.5), process insight and heuristics, the possible candidate control variables (ignoring variable combinations such as ratios) were reduced from $2.67 \cdot 10^{11}$ to 165 possible combinations of three controlled variables. Self-optimisation dictated that controlling reactor temperature, mole % *C* in the purge and either recycle flow or compressor work resulted in the lowest economic self-optimising loss. Larsson et al. (2001) proposed using the feed rate of reactant *C* as the throughput manipulator. After stabilising the TE plant, a 7x7 control system was paired using the RGA. The loops were tuned using the Ziegler-Nichols method, thereby only addressing decoupling through the RGA. However, retuning was required using trial-and-error methods. Larsson et al. (2001) found that this self-optimised scheme remained highly interactive and reverted to a decoupled pairing structure similar to Ricker (1996). Although self-optimisation addressed the selection of controlled variables in a systematic though partially heuristic manner, appropriate pairing and tuning of decentralised control structures remained daunting. As shown by other authors (Price et al., 1994) the throughput manipulator is a critical decision in the design process. Larsson et al. (2001), using self-optimisation, initially chose the feed rate of reactant *C* to set the production rate. In an improved control structure, the throughput manipulator was chosen as the total feed flow with a notably different design (Table 8-1). Larsson et al. (2001) made no mention of how the improved control structure was designed, nor was the reason for the differences between the initial design and the improved design explained. The selection of the controlled variables remains largely consistent (Table 8-1), but the pairing with manipulated variables shows the shortcomings of the RGA for non-linear systems.

Table 8-1 - Two control structures proposed by Larsson et al. (2001), using self-optimisation and insight from Ricker (1996).

Controlled variables	*Initial manipulated variable*	*Improved manipulated variable*
separator level	separator liquid flow	separator liquid flow
separator temperature	**Not used**	condenser cooling water flow
stripper level	stripper liquid product flow	stripper liquid product flow
production rate	C feed flow	total feed flow
product ratio G/H	D feed flow	D/E feed flow ratio
reactor level	E feed flow	set point separator temperature
reactor pressure	purge flow	purge flow
reactor temperature	set point cooling water outlet T	C feed flow
cooling water outlet temperature	reactor cooling water flow	**Not used**
mole % C in purge	A feed flow	C feed flow
recycle flow	condenser cooling water flow	A feed flow

8.3.2 Multivariate control

As suggested by McAvoy & Ye (1994), their multi-loop SISO control system may serve as a platform for an advanced predictive control system. It has been demonstrated that multi-loop SISO and linear MPC-type algorithms are unable to deal with the full range of possible process conditions, unless overrides and logic operators are added to the design process. Creating such overrides may comprise a large portion of the design effort. In essence, multi-loop SISO strategies are not appropriate for dealing with multiple, interacting constraints as posed by the TE problem. A non-linear model predictive control (NMPC) approach may be considered to overcome these shortcomings (Ricker and Lee, 1995).

The TE process is open loop unstable. Although NMPC is possible for unstable processes, the complexity of the design procedure escalates considerably, making stabilisation of the plant using SISO controllers necessary. Before NMPC may thus be considered, the multi-loop SISO control problem first needs to be solved. As described, the multi-loop SISO design that stabilises the open loop plant is non-trivial. Advanced control thus adds an additional layer of complication to a control system. In the NMPC scheme by Ricker & Lee (1995), the manipulated variables are set points of lower level control loops. Poorly tuned PI loops may impact negatively on the performance of the NMPC controller, as the SISO loops change the dynamics of the system on which the NMPC controller is designed. Should the SISO controllers slow the dynamics of the process, the settling times for set point changes by the NMPC controller may be unnecessarily sluggish (Ricker and Lee, 1995).

A NMPC design entails significant design effort. Particular care needs to be taken in the modelling and NMPC formulation. The main stumbling block to NMPC is model development, as the formulation of a useful (i.e., simplified) non-linear model for on-line implementation in the control loop is relatively difficult and time consuming. Though analytical tools are available to support multi-loop SISO loop pairings, these tools have far less value to NMPC designs. The critical process-manipulated variable pairings require substantial engineering judgement and experimentation. This introduces some uncertainty as to whether an optimal choice of control pairings has been made, which impacts on the rest of the design (Ricker and Lee, 1995). For the NMPC development by Ricker & Lee (1995), determining the optimal steady state played an important part in selecting the manipulated variables. Based on the steady state optimisation, only 8 of the possible 12 manipulated variables were used (Ricker, 1995).

Ricker (1996) claimed that the TE control challenge hampers the transparency of a NLMPC design by requiring overrides for special operating conditions. Ricker (1996) compared the decentralised PI control approach and centralised NMPC based on three years of experience with both methodologies. Both NMPC and decentralised control strategies required critical decisions without quantitative justification. The selection of controlled variables from a large set of process variables was the most important qualitative decision process. Ricker (1996) indicated that in a MPC design, controlling only the process variables with defined set points, leaving the remainder to on-line optimisation, fails in the TE process. Existing quantitative methods for control structure selection are mostly inadequate for the TE control challenge. NLMPC control provided excellent set tracking and the prediction horizons and penalty weights were easy to specify. Individual tuning of PI controllers was more time-consuming, but with equal performance. Although MPC should provide better constraint handling than decentralised control, Ricker (1996) found that the TE problem had too many multi-objective goals and special cases. Conventional MPC formulations could not cope with this complexity and decentralised control had better constraint handling. Furthermore, in simple MPC applications the set point tracking weights prioritise the importance of variables based on high weight values. For the TE problem, the importance of a controlled variable depends on the process conditions. Overrides in the form of SISO control loops were required in the most thorough MPC design, although a similar number of overrides were also necessary in decentralised designs. Finally, coordination of the numerous system elements was non-trivial. Experience and engineering judgement alone proved insufficient and dynamic simulation essential. Time-consuming trial-and-error tuning was a feature of both MPC and decentralised control strategy designs (Ricker, 1996).

Sriniwas & Arkun (1997) used the PID structure of Banerjee & MacAvoy (1995) to stabilise the TE plant. A MPC controller was developed using input-output models to provide supervisory control by manipulating the set points of the PID controllers. Mode changes around the base case were facilitated through this supervisory layer. The MPC control structure was kept small, using only the set points of the reactor pressure, reactor level, product flow rate and the mass ratio of G/H in the MPC design. The indirect manipulated variables were the feed rate of reactants D and E, the compressor recycle valve and the reactor cooling water flow rate. The first step in creating the supervisory layer was generating the identification data. The lower PID controllers were uniformly excited with input sequences to all four manipulated variables and the controlled variable responses were recorded. Certain PID loops were closed during the identification process, which masked the non-linearities inherent in the open-loop dynamics. Linear input-output models were deemed sufficient for MPC development. The reactor level and reactor pressure were modelled as SISO models, where the time-lagged set points of the reactor level and pressure were the model inputs. The product flow rate and product mass ratio were modelled as MISO models, where the set points for reactor level, reactor pressure and the flow rates of reactants D and E were the model inputs. Using these linear models, Dynamic Matrix Control (i.e., linear MPC) was implemented. Excellent servo responses with short rise times were obtained. However, the MPC design does not explicitly include economic criteria (Sriniwas & Arkun, 1997).

Many of the difficulties associated with conventional decentralised and centralised control system design are circumvented in an evolutionary reinforcement learning (ERL) framework as outlined in section 8.4.

8.4 NEUROCONTROLLER DEVELOPMENT AND PERFORMANCE

8.4.1 Neurocontroller development

Three neurocontrollers were developed for the TE problem using the SANE algorithm (section 4.3). Each neurocontroller was required to learn to reach one of the three desired set points at the maximum possible production rate and minimum production cost. Although set point changes can be accommodated by a single neurocontroller (section 5.4.2), it was found that the genetic search progressed faster (i.e., more focused) when the goal was limited to a single steady state of high economic return. As each neurocontroller was able to control the plant from a wide range of initial conditions, simple boolean logic allowed for the three neurocontrollers to perform the necessary set point changes in unison.

Each neurocontroller consisted of 63 input nodes, with 12 hidden nodes and 12 output nodes. Each neuron in the hidden layer (sigmoidal activation functions) was allowed 60 connections, which connected the input and output nodes. Each neuron was thus unable to connect to all the input and output nodes, which reduced the dimensionality of each partial solution (i.e., each single neuron) and required neuron cooperation to solve the control task. The 63 input nodes were comprised of the 41 process variables at time, t, and the 22 continuous process variables at time, (t-1). Temporal information may be exploited via such time delay lines. The TE process model has 50 state variables. To ensure a Markov control problem (i.e., all states available for learning) as described in section 3.2.1, the neurocontroller needs to represent the 50 state variables in its internal structure. Although ERL deals effectively with weakly non-Markov control problems, the discrepancy between the number of states and the number of process variables necessitated historical inputs that may have contained state information. Also, the time delay inputs of the continuous process variables allowed for the development of integrating structures in the neural network. The sample period for each neurocontroller was selected as 0.055 [Hz]. This sample frequency allowed for valve transients to pass within half the sample period and for time delay lines to contain pertinent state information.

Table 8-2 - Ranges of process and manipulated variables, normalised between 0 and 1 as inputs and outputs of neurocontrollers.

Neurocontroller inputs & outputs	*Process& manipulated variables*	*Unit*	*Minimum normalisation*	*Maximum normalisation*
y_1	A feed	[kscmh]	0	1
y_2	D feed	[kg/h]	0	6000
y_3	E feed	[kg/h]	0	8500
y_4	A & C feed	[kscmh]	0	15
y_5	Recycle flow	[kscmh]	0	100
y_6	Reactor feed rate	[kscmh]	0	100
y_7	Reactor pressure	[kPa] (g)	2000	3000
y_8	Reactor level	[%]	0	150
y_9	Reactor temperature	[°C]	100	145
y_{10}	Purge rate	[kscmh]	0	1
y_{11}	Product Sep temperature	[°C]	40	160
y_{12}	Product Sep level	[%]	0	150
y_{13}	Product Sep pressure	[kPa] (g)	2000	3500
y_{14}	Product Sep underflow	[m^3/h]	0	48
y_{15}	Stripper level	[%]	0	15
y_{16}	Stripper pressure	[kPa] (g)	2000	3500
y_{17}	Stripper underflow	[m^3/h]	0	40
y_{18}	Stripper temperature	[°C]	40	120
y_{19}	Stripper steam flow	[kg/h]	0	450
y_{20}	Compressor work	[kW]	100	400
y_{21}	Reactor coolant temperature	[°C]	30	160
y_{22}	Separator coolant temperature	[°C]	0	100
y_{23}	Component A to reactor	[mole %]	0	100
y_{24}	Component B to reactor	[mole %]	0	100
y_{25}	Component C to reactor	[mole %]	0	100
y_{26}	Component D to reactor	[mole %]	0	100
y_{27}	Component E to reactor	[mole %]	0	100
y_{28}	Component F to reactor	[mole %]	0	100
y_{29}	Component A in purge	[mole %]	0	100
y_{30}	Component B in purge	[mole %]	0	100
y_{31}	Component C in purge	[mole %]	0	100
y_{32}	Component D in purge	[mole %]	0	100
y_{33}	Component E in purge	[mole %]	0	100
y_{34}	Component F in purge	[mole %]	0	100
y_{35}	Component G in purge	[mole %]	0	100
y_{36}	Component H in purge	[mole %]	0	100
y_{37}	Component D in product	[mole %]	0	100
y_{38}	Component E in product	[mole %]	0	100
y_{39}	Component F in product	[mole %]	0	100
y_{40}	Component G in product	[mole %]	0	100
y_{41}	Component H in product	[mole %]	0	100
u_1	D feed flow	[%]	2.5	100
u_2	E feed flow	[%]	2.5	100
u_3	A feed flow	[%]	2.5	100

u_4	C feed flow	[%]	2.5	100
u_5	Compressor recycle flow	[%]	1	100
u_6	Purge flow	[%]	1	100
u_7	Separator liquid flow	[%]	1	100
u_8	Stripper liquid product flow	[%]	1	100
u_9	Stripper steam flow	[%]	1	100
u_{10}	Reactor cooling water flow	[%]	1	100
u_{11}	Condenser cooling water flow	[%]	1	100
u_{12}	Agitator speed	[%]	1	100

Except for the mole fractions of *G* and *H*, the initial plant state for each evaluation was calculated using a gaussian distribution around the TE base case, with a standard deviation of 10 [%] from the nominal value (Downs and Vogel, 1993). The mole fractions of product *G* and *H* were allowed to vary over their full composition range. The process variables were normalised between 0 and 1 (Table 8-2). The 12 neurocontroller outputs were scaled to the full range of the 12 manipulated variables (Table 8-2). Once an initial condition was calculated that fell within the TE process constraints, a neurocontroller was evaluated over a fixed period of 20 simulated hours. An error value was calculated at each sample interval as indicated in equation 8-1:

$$Error(t) = \left|\frac{VCOP(t)}{1000}\right| \cdot \left|1 - F_{ST}(t)\right| + \left|SP_{C_G} - C_G(t)\right| + \left|SP_{C_H} - C_H(t)\right| \qquad \textbf{(8-1)}$$

where, *VCOP*(*t*) is the normalised variable cost of production, F_{ST} is the stripper bottoms flow rate, C_G and C_H are the mole fractions of *G* and *H* respectively. The fitness function that represented the RL reward from each neurocontroller evaluation is expressed in equation 8-2:

$$Fitness = \frac{1}{\int_0^{20} t \cdot Error(t) \cdot dt} \qquad \textbf{(8-2)}$$

The integration of the error signal results in an ITAE (integral-time-absolute-error) characteristic response in the error signal for each neurocontroller. This ERL implementation thus allows for non-linear pole placement with an approximate ITAE response. As the TE process model is open loop unstable, premature failure of an evaluation typically violated the process constraints during the 20 hour evaluation. Addition process constraints were placed on the reactor pressure and temperature to terminate the evaluation of ineffective neurocontrollers. For example, a reactor pressure below 1800 [kPa] was deemed ineffective, although this condition does not result in shutdown. The search for an optimal reactor temperature was also limited between 90 [°C] and 140 [°C]. Such pressure and temperature range information is assumed available from any catalyst vendor. Additional search constraints reduced the processing time during the early evolutionary generations. The remaining sample periods after premature failure were assigned the highest possible error.

By assigning the highest possible error to the remaining sample periods of an evaluation, a greater emphasis was placed on attaining stability during the early stages of the evolution as per equation 8-2. In later generations, as more neurocontrollers attained stable control over the entire evaluation period, the highest reward became the dominating factor (equation 8-2). When the fitness emphasis shifted from stability to high performance, implicit fitness sharing (section 3.5.5) ensured the continued dynamic optimisation towards the global steady state optimum by maintaining genetic diversity.

8.4.2 Economic optimum and operating mode changes

The NMPC approach by Ricker & Lee (1995) was simplified by first stabilising the open loop unstable TE process model. The SANE algorithm did not need to discriminate between an initially open loop stable or unstable plant. From results published by Ricker and Lee (1995), a set point change required from 10 to 20 hours to settle to the final steady state. As seen in Figure 8-2, a typical neurocontrol settling time for a change in operating mode was less than 5 hours. The poor settling time for NMPC may have been the result of detuned controller parameters in either the stabilising or supervisory control layer, thereby ensuring robust performance.

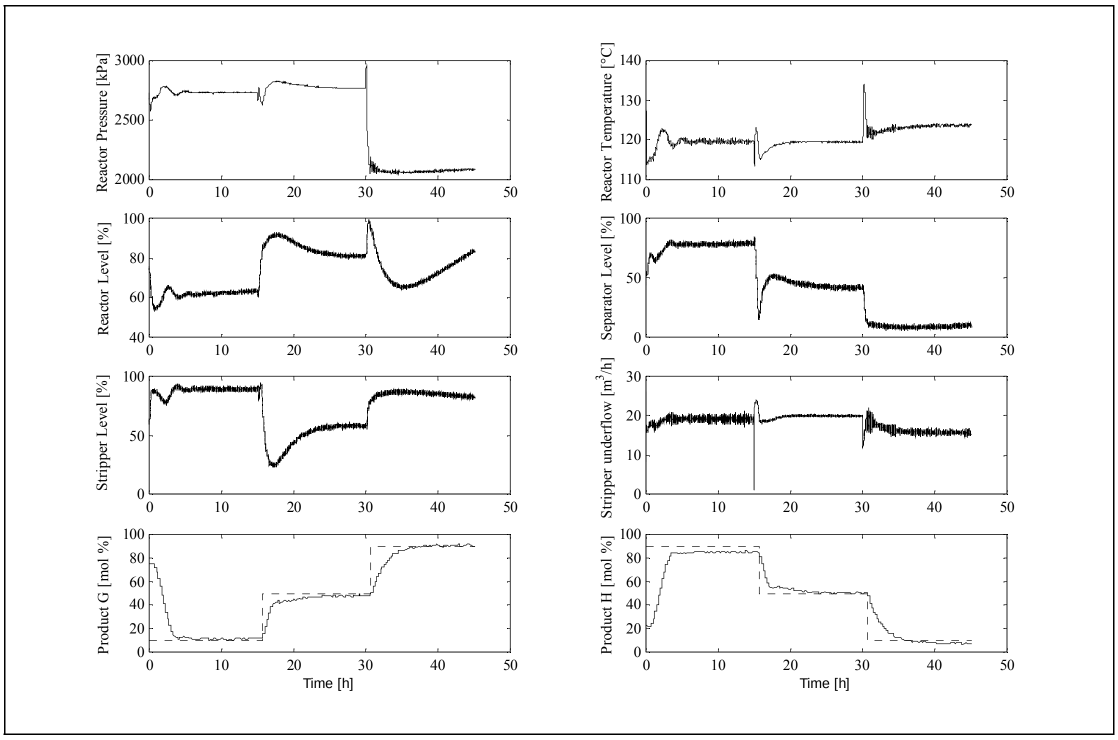


Figure 8-2 – SANE neurocontroller transient response of selected process variables for set point changes from 10/90 to 50/50 to 90/10 G/H. Only sensor noise was included as disturbance during these set point changes.

As indicated in section 8.3.2, an NMPC implementation required the explicit optimisation of the steady state for each operating mode. Ricker (1995) solved the non-linear programming using all 50 state variables, though complete state variable information is seldom available in practice. The optimisation thus served as a benchmark for more realistic optimisation strategies that do not depend on complete state information (Ricker, 1995). The SANE algorithm was not provided with any set point information. In sharp contrast to previously proposed algorithmic approaches (section 8.3), SANE was required to find the states with the highest possible reward as per equation 8-2. The final steady state was determined implicitly by simultaneous dynamic optimisation and neurocontroller development. For example, the SANE algorithm implicitly learned that the region around 120 [°C] was optimal for the reactor temperature and that a higher reactor pressure reduced the operating cost (Figure 8-2). The vessel level limits were held within the shutdown constraints. Significantly, the SANE approach learned to maintain the reactor level above 50 [%], ensuring that sufficient heat transfer area was available for cooling. The SANE algorithm thus eliminated the need for an optimisation step that first determined the optimal economic steady states (i.e., set points).

From an economic perspective the final operating cost attained by each SANE neurocontroller is of great interest. The state variable optimisation by Ricker (1995) served as a benchmark for the more realistic SANE optimisation with knowledge of the only process variables. The operating cost for each set point is also compared in Table 8-3. From Table 8-3 it is evident that the SANE optimisation was sub-optimal to the state variable optimisation. For mode 10/90 and mode 50/50 the operating cost is a fair approximation to the optimal operating cost (Table 8-3). The SANE optimisation thus had attempted to balance the operating cost with attaining the desired product purities without full state information.

Although the SANE algorithm afforded a highly robust search, it may become stalled in local optima. This is illustrated for the SANE neurocontroller that established the 90/10 operating mode change (Figure 8-2). The low reactor pressure and high total operating cost (Table 8-3) reflected a sub-optimal neurocontroller. The insensitivity of the economic objective to large changes in the states may have caused a lack of convergence (Ricker, 1995). The high dimensionality of the TE process also challenged the effectiveness of SANE.

Table 8-3 - Economic optimisation comparison of total operating cost.

Operation mode	*State variable optimisation [\$/m^3]*	*SANE optimisation [\$/m^3]*
Mode 10/90	8.26	10.36
Mode 50/50	6.77	7.66
Mode 90/10	2.44	6.05

In contrast, the SMNE algorithm developed a neurocontroller (for the 90/10 mode with far greater economic return than the SANE algorithm. Within the same number of RL evaluations, SMNE attained the transient responses in Figure 8-3. The process variables were in close approximation to the state variable optimisation by Ricker (1995), with higher reactor pressure and stripper throughput than the SANE neurocontroller. From Figure 8-4, the SMNE neurocontrol had an operating cost of 2.93 [\$/m^3], which compared favourably with the global optimum defined by Ricker (1995) and was a considerable improvement on the SANE neurocontroller.

Table 8-4 - SMNE neurocontroller for the Tennessee Eastman process at the 90/10 operating mode.

INPUT LAYER (PROCESS VARIABLES)

$y_1(t) \ldots y_{41}(t) \quad y_1(t-1) \ldots y_{22}(t-1)$

TRANSPOSED NEURON INPUT WEIGHTS

-1.1469707	0.8936319	0.2792299	1.1565688	0.4487445	0.4354613	-0.5009624	-0.4288719	-0.6717225	0.2463733	-1.3117179	0.2328806
1.9151458	1.0525870	0.2424162	1.2041216	1.2622944	1.6826495	0.4405155	1.1578927	-0.7092059	-0.0481321	-0.6544330	0.2745580
-0.0563952	-2.0891147	-1.7540151	-0.0175148	-2.7872736	-0.9363393	-1.3440263	-2.1593449	-2.0604963	-0.3564261	-1.6902211	-2.0100589
-0.7668506	0.0994404	-0.8182834	0.3272147	0.8432733	0.4123343	-0.2372072	-0.3074891	-1.6698261	-0.0157877	-0.3624145	-1.1665483
-2.1081369	-1.2255448	-0.6046816	-2.6016994	-1.2384782	0.5338687	-0.9141424	-0.8627392	0.0023698	-2.1447716	-1.2680893	-1.0336789
0.1070336	0.7802818	-1.0536016	-0.4713472	0.1836538	1.1848698	-0.6252890	-1.0282726	-0.5500071	0.0238854	-0.7786144	-0.7632372
-0.3056889	-2.0249791	-1.8690004	-0.4141274	-2.1790931	1.3660169	-1.5696166	-2.0550418	-0.1183820	0.2926623	0.2819795	0.7407159
0.8074507	-1.0149945	-1.1791269	0.2665267	-0.8612958	-1.1581910	-0.9297814	-1.2019106	-1.8622222	0.3477920	1.5469375	2.6214592
0.8089502	1.2571001	-0.9117380	0.0844570	2.3642473	0.4580119	-0.6095310	-1.5535510	-1.5377911	0.8997238	-0.0250232	0.0148697
4.0573769	1.5848380	-0.6686477	-0.6848289	0.3300259	0.5683355	0.5683826	-1.4485148	0.4888206	-1.0102483	1.6550000	-0.8431567
-0.1066021	0.1095730	0.4132556	-1.8498796	0.4743028	0.5060569	-0.5530326	-1.2280582	-0.2310223	-1.5182586	-0.5287190	-4.1938248
1.0291873	0.4716901	-1.8660774	0.2155985	-0.3951173	1.9493887	0.5444935	2.4298706	2.1633470	0.1024975	-2.7617297	1.9638227
-0.0521350	0.3147472	-0.3709573	0.7607257	0.3104823	0.8727214	0.4783720	-0.1901107	-0.7575115	0.5141452	0.5018937	-0.0551980
0.0305542	0.1609305	-1.2482759	-0.2708451	-0.5075249	-0.3763364	1.7855783	0.7991126	3.3326786	-1.5742923	-0.1428700	1.0845338
-0.4741140	2.0907354	0.8663805	0.6160809	1.2831047	1.1073121	0.7794776	1.1649431	0.6281672	1.3226919	1.1907934	1.9428296
0.8098642	1.8530803	2.4265630	1.3034537	1.5982816	4.7997909	1.2763263	1.1561654	1.2461452	1.7839934	-0.5473574	0.4333169
0.2464784	-1.4736969	2.9942923	0.8416336	0.0718762	0.6782376	-0.1495495	0.2508069	-1.1003215	1.4057872	5.7236471	-0.0870442
-1.7702888	-1.8568990	0.4159176	-2.0330672	-0.8202661	0.2376561	-3.0939534	-1.3286170	-3.3373132	-1.1566000	4.4534192	-0.9039676
0.3346125	-0.2827565	-0.0962375	0.4432815	-0.3488239	1.1271151	0.3650254	-0.1237609	0.2271687	0.1938286	-0.6609743	-0.8934613
1.4805163	-1.4669470	-0.6052416	-2.1403232	0.3353363	-2.4181409	-1.2644087	0.1198324	0.0818993	-2.5191760	-2.2747669	0.2238890
0.0990292	-0.7009366	-0.2315623	-1.1395887	-0.1086670	0.4685518	-0.5647791	-0.5977541	0.8398868	-1.2014076	-0.8694637	-1.3014239
-0.0827499	1.8605561	2.1552660	2.3166604	0.6606467	3.3997035	1.6482331	0.5567017	2.1014619	2.0318558	1.8661140	1.4690027
-0.2375802	0.3291971	0.9413246	1.1234269	0.2967386	-1.7043359	0.0140548	0.4635281	0.1056048	1.4212270	0.4716570	-1.0741889
0.4248677	-0.5442211	1.4334562	-0.7936524	-0.3924933	-0.7447119	-0.2070992	-0.7818589	0.5850059	-0.9290627	-1.5717046	0.6687658
1.3441862	0.0289985	0.9785963	-0.5072027	0.1666207	-0.4426233	0.9430949	0.9056234	1.7489828	-0.5410565	0.8441809	-2.4039984
1.4337392	-0.7581226	0.5360302	-1.0104446	-1.0218760	-2.1847181	-1.4705160	-0.3417801	1.9364519	-0.4837683	-0.0573207	-2.5730717
0.2374763	-0.6353192	1.1421063	-1.4301078	-0.0304580	0.1573821	-0.7336854	0.1885500	0.6063753	-1.8927441	-2.4007175	-0.7208523
0.7624067	-0.0641344	-0.6537413	-1.0442232	0.7594391	1.2853816	-0.6398553	0.7682175	0.9941425	-0.5039362	0.3089818	-0.1379387
-0.7724456	-0.2025781	-0.1875788	-1.1043384	-0.2859541	-1.8617148	-0.1511319	-0.0069562	-0.1212685	-0.6240074	0.3131533	2.6719458
0.0931290	0.2970105	0.7838951	0.1968521	-0.8742124	0.2048419	-2.1217711	0.4489207	-2.2352471	1.0731405	0.2788934	-0.7723503
0.4794441	-0.1377380	1.0064057	-1.0139107	-1.0193127	0.3063864	-1.2364661	-0.3591885	1.3104608	0.2345340	2.0846703	0.4951828
0.8581834	0.2083436	-2.0745800	0.0023821	-0.1207391	0.7422130	0.6674424	-1.2171062	-0.9755303	-0.2369077	0.8915071	-0.1169709
1.1165367	-0.3326353	-0.8629808	-0.5392546	0.3415376	0.6863732	-0.1309362	0.2554666	-0.7772661	-1.5148644	-1.8995280	0.7076089
-0.5697531	-0.0315023	0.8666861	0.7204946	0.3651695	-0.2533613	1.9913518	0.3903451	0.5016534	-0.7530832	0.4425889	3.6514573
2.7142882	0.1736882	-0.5774795	1.6616772	1.9555660	1.3695819	1.8844122	2.1208036	2.2079952	-0.1032593	-1.0373775	0.3077648
-0.1931563	-0.7355975	-0.2297996	-1.1530052	-0.0841774	-0.1798630	0.1648119	-0.1807967	-4.1605754	0.4730930	0.4850908	2.6517544
-1.7585622	0.8440156	0.8517995	1.8155780	0.1580033	1.6983441	0.8435748	0.4145428	1.4738197	0.2854919	0.9685484	0.5924023
1.8640413	1.3421544	0.0508025	0.2337865	-1.0439206	2.1560853	0.5345342	-1.9692990	-2.4841433	-0.9422926	0.8297541	0.0314422
0.1126824	-0.1158515	-1.0684392	-1.1843150	-0.3554763	0.8220227	-0.9828925	-0.2247446	0.3265035	-0.3185794	-0.0753499	-1.2076635
-1.8240401	-0.7486998	1.6846138	-2.3340983	-2.7727890	-2.3450868	-2.0454409	-2.2981303	-0.2211990	-1.5501341	-0.3954048	0.3419849
0.2391333	0.4091404	-1.1020968	0.8352862	-0.1582425	-0.0520764	0.8943813	-0.4047346	-1.0214883	-0.1072427	-0.6553252	1.5122607
0.0830033	2.0200067	-0.4414814	-2.2047038	2.2241497	0.2576305	-2.3879526	2.1459806	0.1932975	0.1159274	-0.2654929	-1.3443191
0.5003504	0.5836649	-0.2308523	0.4237974	0.4529687	1.5255657	0.6411780	1.2281612	1.3996296	-1.0787674	-0.1551553	-2.8608191
-0.7916158	-1.6628138	0.2139313	0.1851629	-1.6762178	-2.1529660	0.5480534	-1.0160906	0.3369913	0.1987262	-0.1056082	-0.2394455
-1.3716726	-0.5743516	-2.1634808	-1.0948068	0.7189874	-0.9297124	-1.9271612	0.2601795	1.0331975	1.5294921	0.3113570	2.2032135
1.2694919	1.6001891	0.8352109	-0.7532104	0.7826885	1.2792557	-2.3557835	-0.4946452	1.6427307	0.1826239	-0.4584718	-0.0136001
-0.2044386	-0.9721593	1.2416993	-0.6549419	1.3671778	-0.0730013	-2.2925766	1.0041248	3.7890849	0.4141978	0.6013866	0.6298883
-0.2487067	1.7820425	-1.3539354	-1.0444943	-0.7240847	1.3562570	-0.9264528	-0.5863426	-0.1666895	-0.2748767	2.0489254	0.2593775
-1.8095423	-3.2160418	-1.2713180	-0.0328304	1.7257041	-1.1264567	-2.8055665	-0.2930833	-2.5630796	-0.3746459	-0.4907357	-0.0892146
-2.0652299	2.6066930	-3.6271038	0.0005507	0.1675475	-1.4969382	-0.5108944	-1.2254457	-2.8966863	0.0083604	4.3740325	0.3419437
2.0460517	3.9628055	0.8386236	1.5304135	0.3068855	2.5794015	2.8972242	-2.1388059	1.4204363	-0.9411636	2.0097780	-1.9786490
0.0938929	-0.8464087	0.5965968	-0.1725377	-0.3345034	-2.4796336	-0.4781517	-0.6943511	1.3373047	-0.7035649	0.0029089	-2.4414263
-0.7015871	1.4394715	-2.7347436	-0.4771712	1.5173135	-1.3785748	1.1925205	0.2720848	-0.6030785	-0.7479829	-0.4668109	0.6833261
0.7588288	0.4802599	1.5127438	1.2595590	-1.8854825	-0.4208923	2.7275982	-0.9811650	1.1487670	2.6700053	-1.2080479	3.1006095
0.1075429	0.4877128	1.9386928	0.9241946	1.2041340	-0.5580000	-0.3846533	-0.1538519	-0.1982397	0.4551477	-0.4573060	0.3601083
1.9703070	-2.0591264	-1.0763966	0.5721932	0.3543220	0.8659968	0.3364830	3.1431921	0.2358962	-0.0019973	2.2070487	1.8089763
0.9645494	-0.6419954	-0.0120886	-0.3227671	0.2642187	-0.5045097	0.8472000	-0.2165650	-1.3639611	-1.9499512	-1.5348005	-0.8790695
-1.3167827	-0.7858613	0.1352562	-0.3587836	0.7489992	-0.9306968	0.7728910	0.7170967	1.4538985	-0.7361682	-1.3792071	-1.3911697
0.1499859	-0.0745634	-0.8320980	0.2426177	-0.3648543	-2.1985767	-0.3075419	0.2054778	1.9770703	2.5667472	0.6957368	-0.5034222
0.8946636	0.4079188	1.8405849	1.8176494	2.7227106	2.9497743	0.8696660	0.9606488	-0.4448153	-1.3996487	-1.0519576	-0.9306969
0.5730465	1.4347054	1.0205305	1.7536306	1.5673076	-0.8393621	-2.2235000	3.6804757	-3.6523609	-0.5888126	-2.3075233	-0.8451648
1.0823779	-0.4632647	1.5091255	0.7571737	-0.1621126	0.9888470	-0.0601288	-1.0175352	-1.1561015	2.7085109	0.6796777	3.5003724
-0.7927538	-1.6784397	-0.7595213	-0.4332724	-1.7909843	-0.4003238	-0.7530299	-1.0891731	-0.6492891	-1.0171340	-0.4107882	-2.8076756

HIDDEN LAYER (SIGMOIDAL NODES)

$a_1 \ldots a_{12}$

TRANSPOSED NEURON OUTPUT WEIGHTS

0.7298192	3.2250574	-0.0582615	-0.1156824	-0.2938692	2.1819007	-1.1680871	-0.1899580	-0.0695350	2.0975754	0.2622011	3.3453298
-0.8249802	0.5188420	-0.2723796	-0.6143954	-0.7820206	-0.1735836	-0.2045649	-0.8662594	-1.0537887	-0.8699819	0.8542017	-0.7865853
1.2489587	-0.2405953	-0.7830750	-0.2940292	0.0526399	-1.1927899	1.2951391	-0.9066057	-2.6025720	-0.4176418	0.9146375	-0.4867404
-2.0106380	-0.2484831	-0.1877432	0.4286661	2.2069509	-2.4436619	-1.4251519	3.3819413	-0.6211164	-0.1788389	-0.0155228	-0.1626053
-2.1020098	0.0349929	-1.2732805	-0.6683775	0.3994987	-0.8614248	-1.1406537	0.1131350	-0.1288095	0.8368879	1.0744857	0.7008803
-2.8707004	1.4581505	0.5115857	-2.7154009	0.4633722	-0.2988345	-1.2818862	0.1591241	0.2448191	0.0012676	-1.0774531	0.0539823
-1.6010556	0.1868966	-3.2980824	-1.1690460	-0.1366483	2.3742683	0.4131811	0.7763332	2.6350114	0.0320386	-1.9543658	-1.1887667
1.9926333	-1.4825916	-2.1738355	0.2281428	0.7027885	-3.0775533	-1.3280237	1.1592522	-0.2951953	1.0595360	-0.1017034	3.0054116
0.9468944	1.7297634	2.1347456	-0.1734065	-1.3050139	-1.1137064	-1.9524583	1.4174075	-0.7099852	-2.0110202	-0.8542789	-2.2162938
1.2929702	3.0526707	-0.6159256	-1.7514105	-1.9766028	-0.3878916	-2.9459817	-1.6993110	-0.8833701	-1.1571509	2.3483050	-0.9789274
1.5660921	-1.3811404	-0.4124617	0.3670759	-1.8047886	1.3465389	-1.8069233	-1.8175291	-1.0937409	-0.3457187	-0.8376411	-0.5792620
-0.5505116	0.6857509	0.0269788	0.3055088	1.0829645	-0.0672990	0.6633862	-1.9911121	0.3972640	1.3226862	0.7892121	-0.8677865

OUTPUT LAYER (MANIPULATED VARIABLES)

$u_1(t) \ldots u_{12}(t)$

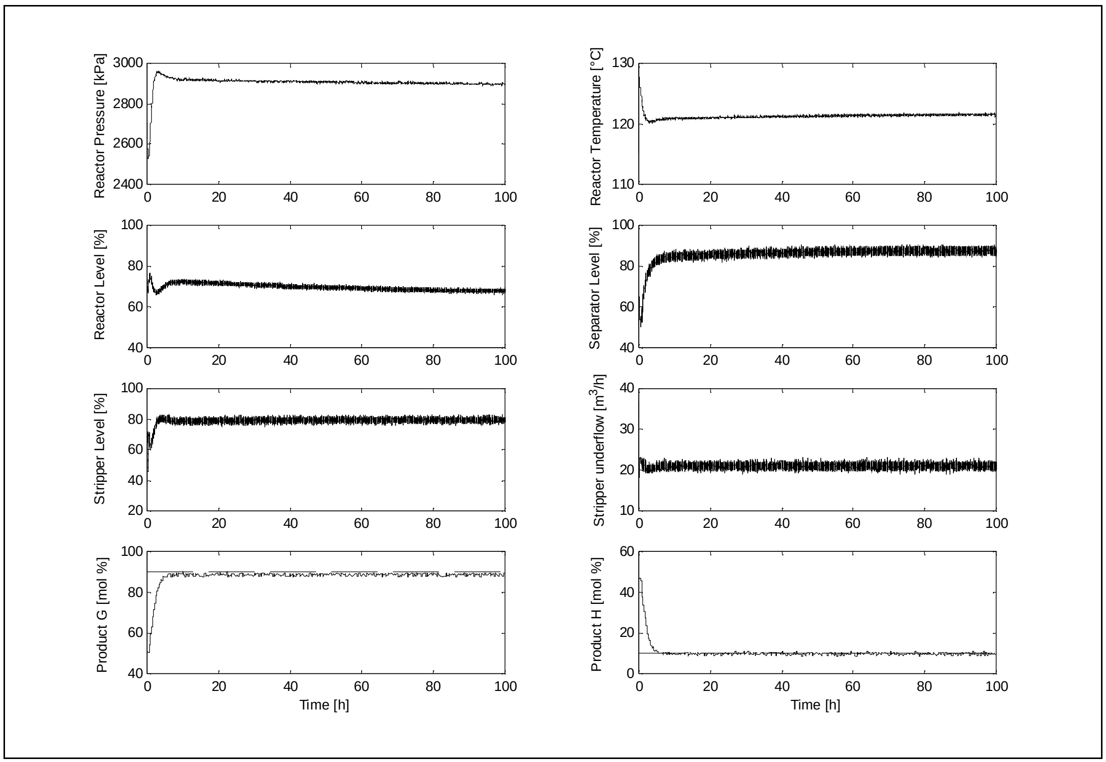


Figure 8-3 – SMNE neurocontrol transient response of selected process variables settling to the set point 90/10 G/H. Only sensor noise was included as disturbance during the set point change.

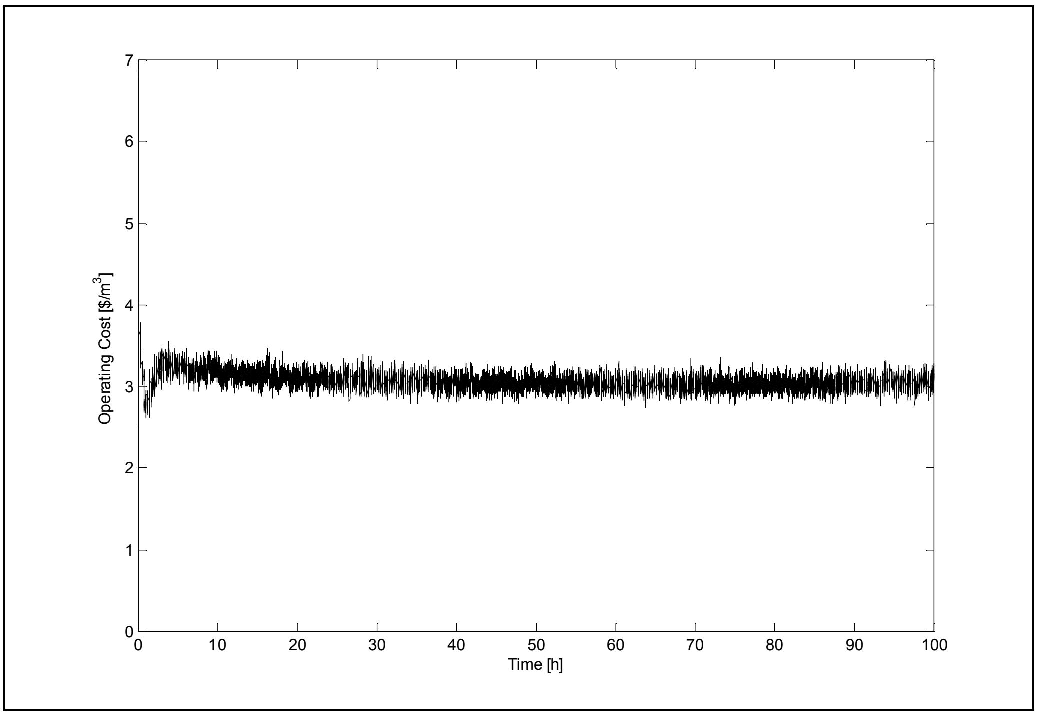


Figure 8-4 – Operating cost of SMNE neurocontroller settling to set point 90/10 G/H. Only sensor noise was included as disturbance during the set point change.

8.4.3 Robust performance and disturbance rejection

High dimensional processes are typically subject to numerous disturbances. The ability of a neurocontroller to reject sensor noise and significant load disturbances reflects the ability of biological models to maintain performance despite uncertain process conditions. Efficient disturbances rejection is solely attributed to the generalisation of the neural network. In conventional plant-wide control, special consideration is given to known disturbances that may occur in the process. The algorithmic control strategy thus incorporates knowledge of known disturbances. Though the inclusion of known disturbances to the reinforcement learning process would have been simple, no explicit knowledge of disturbances was assumed. No disturbances were introduced during RL evaluation, thus no disturbance identification was undertaken (section 2.4.3). The ability of a controller to reject unknown disturbances is a true indication of performance, since unknown (i.e., unmodelled) disturbances are considered a more common real scenario.

Figure 8-5 illustrates the SANE neurocontroller performance in attaining the 50/50 mode from a difficult initial condition with sensor noise and numerous disturbances. The disturbances in Table 8-5 were simultaneously introduced from the start of the simulation, thus requiring the neurocontroller to reject these disturbances during a large process state change. From Figure 8-5, the disturbance rejection was effective and robust performance of the neurocontroller is maintained. The number of simultaneous disturbances rejected by the neurocontroller was far more than required by Down and Vogel (Table 8-5).

Table 8-5 - Simultaneous disturbances rejected by neurocontroller.

Disturbance code	*Disturbance description*	*Type*
IDV 8	A, B, C feed composition	Random variation
IDV 9	D feed temperature	Random variation
IDV 10	C feed temperature	Random variation
IDV 11	Reactor cooling water inlet temperature	Random variation
IDV 12	Condenser cooling water inlet temperature	Random variation
IDV 13	Reaction kinetics	Slow drift
IDV 14	Reactor cooling water valve	Sticking
IDV 15	Condenser cooling water valve	Sticking
IDV 16	Unknown	Unknown
IDV 17	Unknown	Unknown
IDV 18	Unknown	Unknown
IDV 19	Unknown	Unknown
IDV 20	Unknown	Unknown

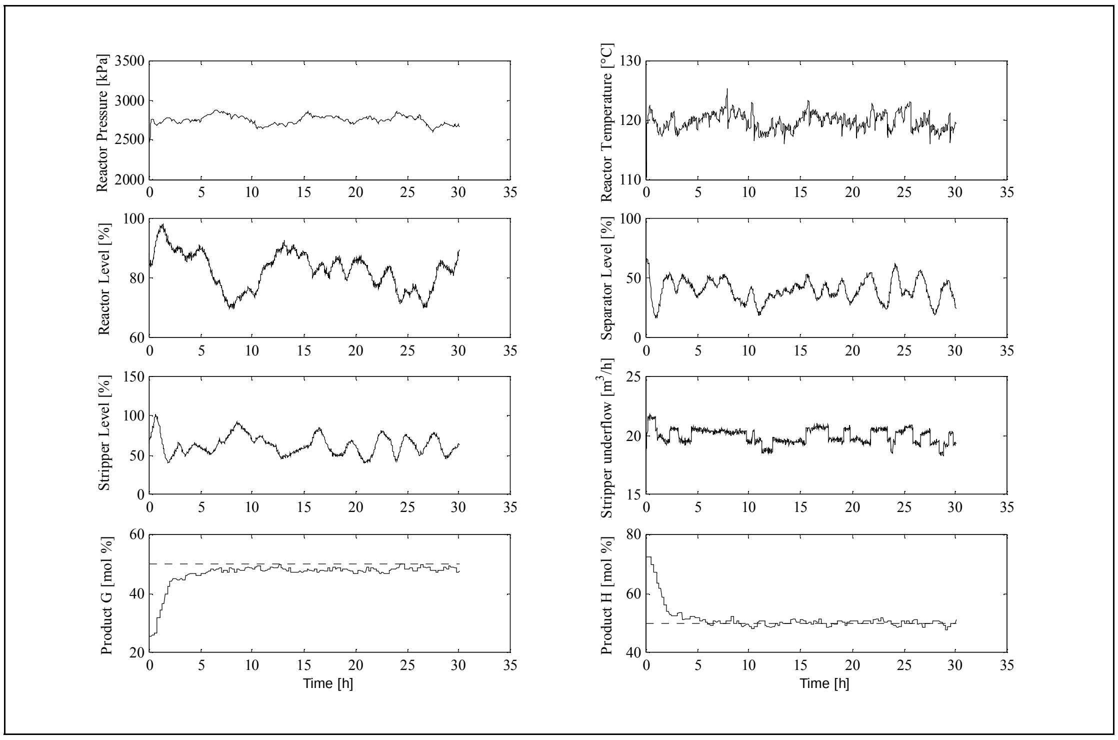


Figure 8-5 – SANE neurocontroller transient response of process variables to the numerous simultaneous disturbances (Table 8-5).

Effective disturbance rejection is directly related to the appropriate pairing of process and manipulated variables. Implicit to both the SANE and SMNE was the decoupling and appropriate pairing of interacting process and manipulated variables. The neurocontroller was provided with all the process variables as inputs and all the manipulated variables as outputs. No information was provided to the either ERL algorithm regarding appropriate process-manipulated variable pairings. As the selection of appropriate pairings was a significant design consideration in previous algorithmic approaches (e.g., NMPC and multi-loop SISO), effective pairing within a high dimensional state space is a major advantage of an ERL design methodology. Only neurons that made connections between appropriate process (inputs) and manipulated (outputs) variables were able to cooperate effectively. Appropriate connections were propagated as effective genetic material from one generation to the next. Almost no engineering judgement or explicit process knowledge was required by the control engineer. The disturbance rejection in Figure 8-5 attests to successful pairing and robust generalisation for a high dimensional process.

8.4.4 Loss of critical feed *A* to the TE process

The most difficult disturbance to reject in the TE process is a loss in component *A*'s feed. No special provision in the form of high reactor pressure overrides was provided and the disturbance was not presented during reinforcement learning. This disturbance proved highly detrimental to the neural network's control strategy. A loss in *A*'s feed was initiated at time 15 [h]. As seen in Figure 8-6, the reactor pressure ramped until the shutdown process was activated. The neurocontroller was thus reliant on *A*'s feed rate for reactor pressure control. Of note, the pairing of reactor pressure and *A*'s feed rate was used by McAvoy & Ye (1994) in their multi-loop SISO design, which was implicitly discovered by the SANE algorithm.

From Figure 8-6, the TE plant was controlled within product specification at slightly reduced production rate for 10 hours. During this 10 [h] period, it is assumed that the feed problem could be resolved on an actual plant. Should this not be the case and the disturbance is a likely occurrence, this disturbance may be incorporated during reinforcement learning. By including a loss in *A*'s feed within each evaluation, the neurocontroller would learn to deal with this known disturbance. Consequently, the ERL algorithm may pair the reactor pressure primarily with the reactor temperature or the purge rate (Ricker & Lee, 1995).

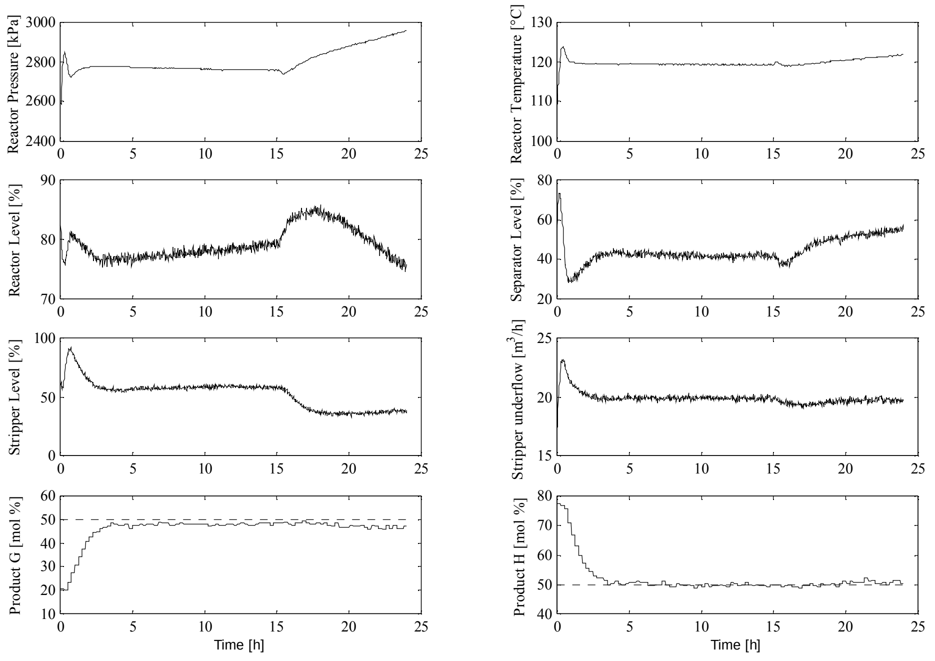


Figure 8-6 -Transient response of selected process variables to a loss in A feed at 15 [h] for SANE neurocontroller.

8.5 CONCLUSIONS

In comparison to the conventional plant-wide control (chapter 2), an ERL methodology required only an understanding of advanced genetic algorithms. A large number of different techniques needed to be mastered by the control engineer in an algorithmic plant-wide control design. An ERL methodology also required no iteration on the design procedure, as would be required in multi-loop SISO designs. SANE attained all the design objectives in a single comprehensive step.

From section 8.4, evolutionary reinforcement learning provides significant benefits for the development of plant-wide control strategies. In this case study, numerous difficulties associated with conventional plant-wide control (chapter 2) are surmounted by a coordinated dynamic optimisation and neurocontroller design methodology. All available process and manipulated variables were used in the controller design, since an ERL approach dealt effectively with the high dimensionality of the TE plant-wide control problem. The economic optimum was implicitly determined, which reduced the design effort in that no prior steady state optimisation was required. The ERL algorithm implicitly solved the non-trivial pairing of process and manipulated variables in high dimensional state space, which required considerable engineering judgement and iterative experimentation in the algorithmic approaches. Effective pairing allowed for superior disturbance rejection through generalisation.

9 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

9.1 CONCLUSIONS

Several criteria exist for evaluating process control methodologies. These are (1) generality, (2) applicability to non-linear systems, (3) controller independent, (4) quantitative, (5) method simplicity, effectiveness and efficiency and (6) the degree of theoretical development (Larsson & Skogestad, 2000). For neural network control, these criteria beg answers to the questions posed by Ydstie (1990): (1) How non-linear, robust controllers based on connectionist networks may be devised and (2) how the tuning algorithm should be modified to ensure global parameter convergence. The neurocontrol paradigm in this thesis addresses these criteria as follows:

- SMNE offers generality with no limitation on dynamic model structure. SMNE has been applied to both phenomenological and semi-empirical models of varying non-linearity. Unlike other non-linear control methodologies, SMNE is applicable to any class of non-linear system.
- SSA offers robust non-linear model development from historical and experimental plant data. Using SSA, state estimation is forthcoming without resorting to fundamental models. SMNE has equal applicability to partial models as confirmed by experimental verification for a MEBAD pilot plant.
- The non-linear structure of a SMNE neurocontroller effectively maps state representations to control actions for highly non-linear processes. Implicit fitness sharing and the synergetic nature of SMNE provide for global tuning of neurocontroller weights. SMNE imparted significant generalisation in the face of process uncertainty.
- Multi-input multi-output (MIMO) SMNE neurocontrollers deal with severe process interaction through optimal pairing of process variables with manipulated variables.
- SMNE and SSA offer a non-heuristic, quantitative approach that requires minimal engineering judgement or knowledge, making the methodology free of subjective design input.
- Both the SMNE and SSA methods are rapid development methods, incorporating a great deal of design simplicity. Coordinated flow sheet design, steady state optimisation and non-linear controller development encompass a comprehensive methodology.
- The SMNE methodology is not confined to set point regulation, but incorporates any control objective such as economic objectives.

- SMNE allows for control strategy design beyond single unit operations. SMNE is equally applicable to processes with high dimensionality, developing plant-wide control strategies. Many of the difficulties in conventional plant-wide control may be circumvented in the biological approach of the SMNE algorithm.
- In addition, ANS allows adaptation to drifting process conditions and tracking of the economic optimum on-line, providing robust performance in the face of significant process uncertainty.

Evolutionary reinforcement learning thus offers significant advantages for developing high performance control strategies for the chemical, mineral and metallurgical industries. Symbiotic memetic neuro-evolution (SMNE), adaptive neural swarming (ANS) and singular spectrum analysis (SSA) present a response to Foss' critique.

9.2 RECOMMENDATIONS FOR FUTURE WORK

- Future work will focus on variations of the SMNE algorithm, including sub-population migration and culling in the particle swarms.
- Though the neurocontrol paradigm has been tested on both simulated and real processes, the method will be tested in other real-world tasks that require both an efficient exploration and an accurate fine-tuning of the controller solutions.
- Future work on SSA will consider the advantages of non-linear principal component analysis using auto-associative neural networks.
- The current incarnation of this neurocontrol paradigm has used sigmoidal activation functions exclusively. Several clear benefits of radial basis functions exist for adaptive neurocontrol and the validity of SSA models.

9.3 FINAL CONCLUDING REMARKS

In a broader sense this thesis has been about discovering the meaning of a moment. Yet, a moment is largely about the present, with almost no consideration for the distant past or the far future. A moment in time, even with a causal understanding of the underlying state variables, tells us almost nothing about living and dying; or about being born.

REFERENCES

Abarbanel, H.D.I., R. Brown, J.J. Sidorowich and L.S. Tsimring, "The analysis of observed chaotic data in physical systems," Reviews of Modern Physics, **65**(4), 1331-1392 (1993).

Agrawal, M., "A Systematic Classification of Neural-Network-Based Control," *IEEE Control Systems*, 75-93, (April 1997).

Agrawal, P., C. Lee, H.C. Lim and D. Ramkrishna, "Theoretical Investigations of Dynamic Behavior of Isothermal Continuous Stirred Tank Biological Reactors," *Chemical Engineering Science*, **37**(3), 453-462 (1982).

Amirthalingam, R., and J.H. Lee, "Subspace identification based infertential control of a continuous pulp digester," *Computers and Chemical Engineering*, **21**, S1143-1148 (1997).

Angeline, P.J., "Tracking Extrema in Dynamic Evironments," In: *Proceedings of the 6th International Conference on Evolutionary Programming*, Vol. 1213, 335-345 (1997).

Bahri, P.A., Bandoni, J.A., & Romagnoli, J.A. (1997), "Integrated Flexibility and Controllability Analysis in Design of Chemical Processes," AIChE Journal, 43(4), 997-1015.

Banerjee, A., and Y. Arkun, "Control Configuration Design Applied to the Tennessee Eastman Plant-wide Control Problem," *Computers and Chemical Engineering*, **19**(4), 453-480 (1995).

Bansal V., J.D. Perkins, EN Pistikopoulos, "A Case Study in Simultaneous Design and Control using Rigorous, Mixed-Integer Dynamic Optimization Models," *Industrial and Engineering Chemistry Research*, **41**(4), 760-778 (2002).

Barker, I.J. and D.G. Hulbert, "Dynamic Behaviour in the Control of Milling Circuits," In: *Proceedings of the 4th IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, T. Westerlund (ed.), Pergamon Press, Oxford, 139-152 (1983).

Bequette, B.W., "Nonlinear Control of Chemical Processes: A Review," *Industrial and Engineering Chemistry Research*, **30**, 1391-1413 (1991).

Bhat, N, and T.J. McAvoy, "Use of Neural Networks for Dynamic Modeling and Control of Chemical Process Systems," *Computers and Chemical Engineering*, **14**(4/5), 573-583 (1990).

Brengel, D.D., & Seider, W.D. (1989). Multistep Nonlinear Predictive Controller. *Industrial &Engineering Chemistry Research*, 28, 1821-1822.

Brengel, D.D., and W.D. Seider, "Coordinated Design and Control Optimization of Nonlinear Processes," *Computers and Chemical Engineering*, **16**(9), 861-886 (1992).

Broomhead, D.S., and G.P. King, "Exacting Qualitative Dynamics from Experimental Data," *Physica D*, **20**, 217-236 (1986).

Carlisle, A., and G. Dozier, "Adapting Particle Swarm Optimization to Bynamic Environments," *Proceedings ICAI 2000*, Las Vegas, Vol. I, 429-434 (2000).

Chang, H.-C., and L.-H. Chen, "Bifurcation characteristics of nonlinear systems under conventional PID control," *Chemical Engineering Science*, **39**(7/8), 1127-1142 (1984).

Chen,F., and H.K. Khalil, "Adaptive Control of a Class of Nonlinear Discrete-Time Systems using Neural Networks," *IEEE Transactions on Automatic Contro*l, **40**(5), 791-801 (1995).

Chiu, S., "Developing Commercial Applications of Intelligent Control," IEEE Control Systems, 94-97 (April 1997).

Cho, C., R. Vance, N. Mardi, Z. Qian and K. Prisbey, "Neural Network-Based Nonlinear Model Predictive Control vs. Linear Quadratic Gaussian Control," *Minerals and Metallurgical Processing*, **14**(2), 43-46 (1997).

Collins, R.J., and D.R. Jefferson, "Selection in Massively Parallel Genetic Algorithms," In: Proceedings of the Fourth International Conference on Genetic Algorithms, R. Belew and L. Booker (eds.), Morgan Kaufmann, San Mateo CA., USA, 249-256 (1991).

Conradie, A.v.E., *Neurocontroller Development for Nonlinear Processes Utilising Evolutionary Reinforcement Learning*, M.Sc. Eng. Thesis (Chemical Engineering), University of Stellenbosch, Stellenbosch, South Africa (2000).

Conradie, A.v.E., R. Miikkulainen, and C. Aldrich, "Adaptive control utilizing neural swarming", In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, NY, 2002.

Conradie, A.v.E., R. Miikkulainen, and C. Aldrich, "Intelligent process control utilising symbiotic memetic neuro-evolution", In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002), vol. 1, pp.623-628, Hawaii, USA, 2002.

Conradie, A.v.E., I. Nieuwoudt and C. Aldrich, "Nonlinear neurocontroller development with evolutionary reinforcement learning," In: 9th National Meeting of SAIChE, Secunda, South Africa (2000).

Davidor, Y., "A Naturally Occurring Niche and Species Phenomenon: The Model and First Results," In: *Proceedings of the Fourth International Conference on Genetic Algorithms*, K. Belew and L.B. Booker (eds.), Morgan Kaufmann, San Mateo CA., USA, 257-263 (1991).

Desbiens, A., F. Flament and A. Pomerleau, "Distributed Control a the Kidd Creek Grinding Plant. Part II: Implementation," CIM Bulletin, 90(1008), 145-150 (1997).

Desbiens, A., A. Pomerleau and K. Najim, "Adaptive Predictive Control of a Grinding Circuit," *International Journal of Mineral Processing*, **41**, 17-31 (1994).

Dirion, J.-L., M. Cabassud, M.V. Le Lann and G. Casamatta, "Design of a Neural Controller by Inverse Modelling," *Computers and Chemical Engineering*, **19**(S), S797-S802 (1995).

Dirion, J.-L., M. Cabassud, M.V. Le Lann and G. Casamatta, "Development of adaptive neural networks for flexible control of batch processes," *The Chemical Engineering Journal*, **63**, 65-77 (1996).

Distefano, G.P., "Mathematical Modeling and Numerical Integration of Multicomponent Batch Distillation Equations," *AIChE Journal*, **14**(1), 191-199 (1968).

Downs, J.J, and E.F. Vogel, "A Plant-Wide Industiral Process Control Problem," *Computers & Chemical Engineering*, **17**(3), 245-255 (1993).

Eaton, J.W., & Rawlings, J.B. (1992). Model-Predictive Control of Chemical Processes. *Chemical Engineering Science*, **47**(4), 705 - 720.

Economou, C.G., & Morari, M. (1986),"Internal Model Control. 5. Extension to Nonlinear Systems," *Industrial & Engineering Process Des. Dev.* , **25**, 403 - 411.

Flament, F., A. Desbiens and R. del Villar, "Distributed Control at the Kidd Creek Grinding Plant. Part I: Control Strategy Design," *CIM Bulletin*, **90**(1008), 139-144 (1997).

Flament, F., J. Thibault and D. Houdouin, "Neural Network Based Control of Mineral Grinding Plants," *Minerals Engineering*, **6**(3), 235-249 (1993).

Forssell, U. and L. Ljung, "Closed-loop identification revisited," *Automatica*, **35**, 1215 - 1241 (1999).

Foss, A.S., "Critique of Chemical Process Control Theory," *AIChE Journal*, **19**(2), 209-215 (1973).

Foster, W.R., F. Collopy and L.H. Ungar, "Neural Network ForeCasting of Short, Noisy Time Series," *Computers and Chemical Engineering*, **16**(4), 293-297 (1992).

Furlonge, H.I., C.C. Pantelides and E. Sorensen, "Optimal operation of multivessel batch distillation columns," *AIChE Journal*, **45**(4), 781-801.

Ghanadan, R., "Adaptive PID control of nonlinear systems," M.Sc. thesis, University of Maryland, USA (1990).

Gomez, F. and R. Miikkulainen, "Incremental Evolution of Complex General Behaviour," *Adaptive Behavior*, **5**, 317-342 (1997).

Gossman, G.I., and A. Buncombe, "The application of a Microprocessor-Based Multivariable Controller to a Gold Milling Circuit," In: *Proceedings of the 4th IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, T. Westerlund (ed.), Pergamon Press, Oxford, 177-188 (1983).

Greffenstette, J.J., C.L. Ramsey and A.C. Schultz, "Learning Sequential Decision Rules Using Simulation Models and Competition," *Machine Learning*, **5**, 355-381 (1990).

Gupta, M.M. and D.H. Rao, "Neuro-Control Systems: A Tutorial," In: *NeuroControl Systems: Theory and Application*, M.M. Gupta and D.H. Rao. (eds), IEEE Press, ? , 1-43 (1994).

Harris, K.R., and A. Palazoglu, "Studies on the Analysis of Nonlinear Processes via Functional Expansions - III: Controller Design," *Chemical Engineering Science*, **53**(23), 4005-4022 (1998).

Hecht-Nielsen, R., "Neurocomputing: picking the human brain," *IEEE Spectrum*, **25**(3), 36-41 (1988).

Henson, M.A., and D.E. Seborg, "Nonlinear Process Control," Prentice Hall, New Jersey, 1995.

Henson, M.A., "Nonlinear Model Predictive Control: Current Status and Future Directions," *Computers and Chemical Engineering*, **23**, 187-202 (1998).

Herbst, J.A., and K. Rajamani, "Evaluation of Optimising Control Strategies for Closed Circuit Grinding," In: *Developments in Mineral Processing*, D.W. Fuerstenau (ed.), International Mineral Processing Congress, Warsaw, 1642-1670 (1979).

Herbst, J.A., K. Robertson and K. Rajamani, "Mill Speed as a Manipulated Variable for Ball Mill Grinding Control," In: *Proceedings of the 4th IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, T. Westerlund (ed.), Pergamon Press, Oxford, 153-160 (1983).

Hernadez, E., and Y. Arkun, "Study of the control-relevant properties of backpropagation neural network models of nonlinear dynamic systems," *Computers and Chemical Engineering*, **16**(4), 227-240 (1992).

Hodouin, D., S.-L. Jämsä-Jounela, M.T. Carvalho and L. Bergh, "State of the Art and Challenges in Mineral Processing Control," *Control Engineering Practice*, **9**, 995-1005 (2001).

Horn, J., D.E. Goldberg and K. Deb, "Implicit niching in the Learning Classifier System: Nature's Way," *Evolutionary Computation*, **2**(1), 37-66 (1994).

Horn, J., and D.E. Goldberg, "Natural Niching for Evolving Cooperative Classifiers," In: *Genetic Programming - Proceedings of the First Annual Conference*, The MIT Press, Cambridge MA., USA, 553-564 (1996).

Hornik, K., M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, **2**, 359-366 (1989).

Hoskins, J.C., and D.M. Himmelblau, "Process Control via Artificial Neural Networks and Reinforcement Learning," *Computers & Chemical Engineering*, **16**(4), 241-251 (1992).

Hrycej, T., "Neurocontrol: Towards an industrial control methodology," John Wiley & Sons, New York, 223-242 (1997).

Hulbert, D.G., J. Koudstaal, M. Braae and G.I. Goosen, "Multivariable Control of an Industrial Grinding Plant," In: *Proceedings of the 3rd IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, J. O'Shea and M. Polis (eds.), Pergamon Press, Oxford, 311-322 (1980).

Hulbert, D.G., I.K. Craig, M.L. Coetzee and D. Tudor, "Multivariable Control of a Run-of-Mine Milling Circuit," *Journal of South African Institute of Mining and Metallurgy*, **90**(7), 173-181 (1990).

Hunt, K.J., D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, "Neural Networks for Control Systems - A Survey," *Automatica*, **28**(6), 1083-1112 (1992).

Hussain, M.A., "Review of the applications of neural networks in chemical process control - simulation and online implementation," *Artificial Intelligence in Engineering*, **13**, 55-68 (1999).

Hussain, M.A., and L.S. Kershenbaum, "Implementation of an inverse-model-based control strategy using neural networks on a partially simulated exothermic reactor," *Chemical Engineering Research and Design*, **78**(A2), 299-311 (2000).

Jang, J.S.R., and C.T. Sun, " Functional equivalence between radial basis function networks and fuzzy inferential systems," *IEEE Transactions on Neural Networks*, **4**(1), 156-159 (1993).

Jämsä, S.-L., H. Melama and J. Penttinen, "Design and Experimental Evaluation of a Multivariable Grinding Circuit Control System," In: *Proceedings of the 4th IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, T. Westerlund (ed.), Pergamon Press, Oxford, 189-198 (1983).

Jarmulak, J., P. Spronck, E.J.H. Kerckhoffs, "Neural networks in process control: model-based and reinforcement trained controllers," *Computers and Electronics in Agriculture*, **18** , 149-166 (1997).

Kaelbling, L.P., M.L. Littman, A.W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, **4**, 237-285 (1996).

Kavchek, M., and H. Budman, "Adaptive neural network structures for non-linear process estimation and control," *Computers and Chemical Engineering*, **23**(9), 1209-1228 (1999).

Kershenbaum, L, "Experimental testing of advanced algorithms for process control: When is it worth the effort?," Chemical *Engineering Research and Design*, **78**(A4), 509-521 (2000).

Khalid, M. and S. Omatu, "A Neural Network Controller for a Temperature Control System," IEEE Control Systems Magazine, 58-64 (June 1992).

Kim, S., Lee, M., Park, S., Lee, S. & Park, C.H. (1997). A Neural Linearizing Control Scheme for Nonlinear Chemical Processes. *Computers and Chemical Engineering*, 21(2), 187 - 200.

Kim, K.J., and U.M. Diwekar, "New era in batch distillation: Computer aided analysis, optimal design and control," *Reviews in Chemical Engine*ering, **17**(2), 111-164 (2001).

Krishnapura, V.G., and A. Jutan, "A neural adaptive controller," *Chemical Engineering Science*, **55**, 3803-3812 (2000).

Kugiumtzis, D., "State space reconstruction parameters in the analysis of chaotic time series - the role of the time window length*," Physica D*, **95**, 13-28 (1996).

Kumar, S.S., and A. Guez, "ART Based Adaptive Pole Placement for Neurocontrollers," *Neural Networks*, **4**, 319-335 (1991).

Kurooka, T., H. Nishitani, S. Hasebe and I. Hashimoto, "Energy Conservation by Multi-Effect Batch Distillation System," *Journal of Chemical Engineering Japan*, **34**(9), 1141-1146 (2001).

Kuttisupakorn, P., M.A. Hussain and J. Petcherdask, "Studies on the Use of Neural Networks in Nonlinear Control Strategies," *Journal of Chemical Engineering Japan*, **34**(4), 453-465 (2001).

Larsson, T., K. Hestetun, E. Hovland and S. Skogestad, "Self-Optimization of a Large Scale Plant: The Tennessee Eastman Process," Industrial and Engineering Chemistry Research, **40**(22), 4889-4901 (2001).

Larsson, T., and S. Skogestad, "Plantwide control - A Review and a New Design Procedure," *Modeling, Identification and Control*, **21**(4), 209-240 (2000).

Leonard, J.A., M.A. Kramer and L.H. Ungar, "A Neural Network Architecture that Computes its Own Reliability," *Computers and Chemical Engineering*, **16**(9), 819-835 (1992).

Lo, Y.C., A.E. Oblad and J.A. Herbst, "Cost Reduction in Grinding Plants through Process Optimization and Control," *Minerals and Metallurgical Processing*, **13**(1), 19-21 (1996).

Luyben, M.L., and C.A. Floudas, "Analyzing the Interaction of Design and Control - 1. A Multiobjective Framework and Application to Binary Distillation Synthesis," *Computers and Chemical Engineering*, **18**(10), 933-969 (1994a).

Luyben, M.L., and C.A. Floudas, "Analyzing the Interaction of Design and Control - 2. Reactor-Separator-Recycle System," Computers and Chemical Engineering, 18(10), 971-994 (1994b).

Luyben, M.L., B.D. Tyreus and W.L. Luyben, "Plantwide Control Design Procedure," *AIChE Journal*, **43**(12), 3161-3174 (1997).

Lyman, P.R., and C. Georgakis, "Plant-wide Control of the Tennessee Eastman Problem," *Computers and Chemical Engineering*, **19**(3), 321-331 (1995).

Mahfoud, S.W., "Crowding and Preselection Revisited," In: *Parallel Problem Solving from Nature*, R. Manner and B. Manderick (eds.), Elsevier Science Publishers, Amsterdam, The Netherlands, **2**, 27-36 (1992).

Martin, G., and S. McGarel, "Nonlinear Mill Control," *ISA Transactions*, **20**, 369-379 (2001).

McAvoy, T.J., and N. Ye, "Base Control for the Tennessee Eastman Problem," *Computers & Chemical Engineering*, **18**(5), 383-413 (1994).

McAvoy, T.J., "Synthesis of Plantwide Control Systems using Optimization," *Industrial and Engineering Chemistry Research,* **38**, 2984-2994 (1999).

McCrone, J. (2002*), How the brain works*, Dorling Kindersley, London.

McQueston, P., and R. Miikkulainen, "Culling and Teaching in Neuro-evolution," Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-97), Morgan Kaufmann, San Francisco, 760-767 (1997).

Megan, L., and D.J. Cooper, "Neural Network Based Adaptive Control via Temporal Pattern Recognition," *The Canadian Journal of Chemical Engineering*, **70**, 1208-1219 (1992).

Merz, P., "Memetic algorithms for combinatorial optimization problems," Ph.D. thesis, University of Siegen, Germany, 2000.

Minsky, M.L., "Steps Toward Artificial Intelligence," *Proceedings of the Institute of Radio Engineers*, **49**, 8-30.

Mohideen, M.J., Perkins, J.D., & Pistikopoulos E.N. (1997), "Robust stability considerations in optimal design of dynamic systems under uncertainty," Journal of Process Control, 7(5), 371-385.

Mohideen, M.J., Perkins, J.D., & Pistikopoulos E.N. (1996), "Optimal Design of Dynamic Systems under Unicertainty," *AIChE Journal*, **42**(8), 2251-2272.

Morari, M., Y. Arkun and G. Stephanopoulos, "Studies in the Synthesis of Control Structures for Chemical Processes Part I: Formulation of the Problem. Process Decomposition and the Classification of the Control Tasks. Analysis of the Optimizing Control Structures," *AIChE Journal*, **26**(2), 220-232 (1980).

Morari, M., and J.H. Lee, "Model Predictive Control: Past, Present and Future," *Computers and Chemical Engineering*, **23**, 667-682 (1999).

Moriarty, D.E., and R. Miikkulainen, "Efficient Reinforcement Learning through Symbiotic Evolution," Machine Learning, **22**, 11-32 (1996a).

Moriarty, D.E., and R. Miikkulainen, "Hierarchical Evolution of Neural Networks," Technical Report AI96-242, Department of Computer Sciences, The University of Texas at Austin, USA (1996b).

Moriarty, D.E., *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*, Ph.D. Thesis (Computer Science), University of Texas at Austin, Austin, Texas, USA (1997).

Moriarty, D.E., and R. Miikkulainen, "Forming Neural Networks through Efficient and Adaptive Coevolution," *Evolutionary Computation*, **5**(4), 373-399 (1998).

Moriarty, D.E., A.C. Schultz and J.J. Grefenstette, "Evolutionary Algorithms for Reinforcement Learning," *Journal of Artificial Intelligence Research*, **11**, 241-276 (1999).

Narenda, K.S., "Neural Networks for Control: Theory and Practice," *Proceedings of the IEEE*, **84**(10), 1385-1406 (1996).

Narendra, K.S., and K. Parthasarathy, "Identification and Control of Dynamic Systems using Neural Networks," *IEEE Transactions on Neural Networks*, **1**(1), ?, (1990).

Narenda, K.S., and J. Balakrishnan, "Adaptive Control Using Multiple Models," *IEEE Transactions on Automatic Control*, **42**(2), 171-187 (1997).

Nguyen, D.H., and B. Widrow, "Neural Networks for Self-Learning Control Systems," *IEEE Control Systems Magazine*, April 1990, 18-23.
Noda, M., T. Chida, S. Hasebe, and I. Hashimoto, "On-line optimization system of pilot scale multi-effect batch distillation system," *Computers and Chemical Engineering*, **24**, 1577-1583 (2000).

Nolfi, S., J.L. Elman and D. Parisi, "Learning and evolution in neural networks," *Adaptive Behavior*, **3**(1), 5-28 (1994).

Parlos, A.G., S. Parthasarathy, and A.F. Atiya, "Neuro-predictive process control using on-line controller adaptation," *IEEE Transactions on Control Systems Technology*, **9**(5), 741-755 (2001).

Palancar, M.C., J.M. Aragón and J.S. Torrecilla, "pH-Control System Based on Artificial Neural Networks," *Industrial and Engineering Chemistry Research*, **37**, 2729-2740 (1998).

Pearl, J., "The New Challenge: From a Century of Statistics to an Age of Causation," IASC Second World Congress, Pasadena CA., USA, ?, (1997).

Piché, S., B. Sayyar-Rodsari, D. Johnson and M. Gerules, "Nonlinear Model Predictive Control Using Neural Networks," *IEEE Control Systems Magazine*, **20**(3), 53-62 (2000).

Polani, D. and R. Miikkulainen, "Eugenic Neuro-Evolution for Reinforcement Learning," *Proceedings of the Genetic and Evolutionary Computation Conference*

(GECCO 2000), Las Vegas, Nevada, Morgan Kaufmann, San Francisco, 1041-1046 (2000).

Pomerleau, A., D. Hodouin, A. Desbiens and E. Gagnon, "A Survey of Grinding Control Methods: From Decentralized PID Controllers to Multivariable Predictive Controllers," *Powder Technology*, **108**, 103 - 115 (2000).

Powell, M.J.D. (1964), "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *Comput. J.*, **7**, 155-162.

Potgieter, T.I., S.T.L. Harrison and C.L.E. Swartz, "The Role of Fundamental Modelling in Fermentation Optimisation," SAIChE 2001, Secunda, 2001.

Pottmann, M., and M.A. Henson, "Compactly supported radial basis functions for adaptive process control," *Journal of Process Control*, **7**(5), 345-356 (1997).

Poznyak, A.S., W. Yu, E.N. Sanchez, and J.P. Perez, "Nonlinear Adaptive Trajectory Tracking using Dynamic Neural Networks," *IEEE Transactions on Neural Networks*, **10**(6), 1402-1411 (1999).

Price, R.M., Lyman, P.R., and Georgakis, C., "Throughput Manipulation in Plantwide Control Structures," *Industrial and Engineering Chemistry Research*, 33, 1197-1207 (1994).

Psichogios, D.C., & Ungar, L.H. (1991). Direct and Indirect Model based Control Using Artificial Neural Networks. *Industrial & Engineering Chemistry Research*, 30, 2564 - 2573.

Rajamani, R.K. and J.A. Herbst, "Optimal Control of a Ball Mill Grinding Circuit - I. Grinding Circuit Modelling and Dynamic Simulation," *Chemical Engineering Science*, **46**(3), 861-870 (1991a).

Rajamani, R.K. and J.A. Herbst, "Optimal Control of a Ball Mill Grinding Circuit - II. Feedback and Optimal Control," *Chemical Engineering Science*, **46**(3), 871-879 (1991b).

Raju, G.K., and C.L. Cooney. "Active Learning for Process Data," AIChE Journal, 44(10), 2199-2211 (1998).

RayChaudhuri, T., L.G.C. Hamey, R.D. Bell, "From conventional control to autonomous intelligent methods," *IEEE Control Systems Magazine*, **16**(5), 78-84 (Oct. 1996).

Ricker, N.L., "Optimal Steady-State Operation of the Tennessee Eastman Challenge Process," *Computers & Chemical Engineering*, **19**(9), 949-959 (1995).

Ricker, N.L., and J.H. Lee, "Nonlinear Model Predictive Control of the Tennessee Eastman Challenge Process," *Computers & Chemical Engineering*, **19**(9), 961-981 (1995).

Ricker, N.L., and J.H. Lee, "Nonlinear modeling and state estimation for the Tennessee Eastman challenge process," *Computers & Chemical Engineering*, **19**(9), 983-1005 (1995b).

Ricker. N.L., "Decentralized Control of the Tennessee Eastman Challenge Process," *Journal of Process Control*, **6**(4), 205-221 (1996).

Robinson D., R. Chen, T. McAvoy and P.D. Schelle, "An Optimal Control Based Approach to Designing Plantwide Control System Architectures," *Journal of Process Control*, **11**, 223-236 (2001).

Rowe, D.A., Perkins, J.D., & Walsh, S.P. (1997), "Integrated Design of Responsive Chemical Manufacturing Facilities, " *Computers and Chemical Engineering*, 21, S101-S106.

Saerens, M., and A. Soquet, "Neural controller based on back-propagation algorithm," *IEE Proceedings-F*, **138**(1), 55-62 (1991).

Sanner, R.M., & Slotine, J-J. E., "Gaussian Networks for Direct Adaptive Control," IEEE Transactions on Neural Networks, 3(6), 837-863 (1992).

Savkovic-Stevanovic, J., "Neural Net Controller by Inverse Modeling for a Distillation Plant," *Computers and Chemical Engineering*, **20**(S), S925-S930 (1996).

Seborg, D.E., T.F. Edgar and D.A. Mellicamp, "Process Dynamics and Control," John Wiley & Sons, New York (1989).

Seider, W.D., and D.D. Brengel, "Nonlinear Analysis in Process Design," *AIChE Journal*, **37**(1), 1-38 (1991).

Seider, W.D., D.D. Brengel, A.M. Provost and S. Widagdo, "Nonlinear Analysis in Process Design. Why Overdesign to Avoid Complex Nonlinearities," *Industrial and Engineering Chemistry Research*, **29**(5), 805-818 (1990).

S'equim, C.H., and R.D. Clay, "Fault tolerance in artificial neural networks," *International Joint Conference on Neural Networks*, San Diego, Vol. 1, 703-708 (1990).

Shi, Y., and R.C. Eberhart, "Empirical study of particle swarm optimization," In: Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Service Centre, Piscataway, NJ, 1945-1950 (1999).

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P-Y. Glorennec, H. Hjalmarsson and A. Juditsky, "Nonlinear Black-box Modeling in System Identification: a Unified Overview," *Automatica*, **31**(12) 1691-1724 (1995).

Skogestad, S., "Plantwide Control: The Search for the Self-Optimizing Control Structure," *Journal of Process Control*, **10**, 487 - 507 (2000a).

Skogestad, S., "Self-optimizing control: the missing link between steady-state optimization and control," *Computers and Chemical Engineering*, **24**, 569-575 (2000b).

Smith, R.E., S. Forrest and A.S. Perelson, "Searching for Diverse, Cooperative Populations with Genetic Algorithms," *Evolutionary Computation*, **1**(2), 127-149.

Sriniwas, G.R., and Y. Arkun, "Control of the Tennessee Eastman process using input-output models," *Journal of Process Control*, **7**(5), 387-400 (1997).

Stanley, K.O., and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evolutionary Computation*, **10**(2), 99-127 (2002).

Stephanopoulos G., "Synthesis of Control Systems for Chemical Plants - A Challenge to Creativity," *Computers and Chemical Engineering*, **7**(4), 331-365 (1983).

Stephanopoulos, G., & C. Han, "Intelligent Systems in Process Engineering: A Review," *Computers and Chemical Engineering*, **20**(6/7), 743-791 (1996).

Stephanopoulos and Ng, "Perspectives on the Synthesis of Plant-wide Control Structures," *Journal of Process Control*, **10**, 97-111 (2000).

Sutton, R.S., and A.G. Barto (1998), *Reinforcement Learning - An Introduction*, 1st ed., A Bradford Book, Cambridge Massechusetts.

Tanese, R., "Distributed Genetic Algorithms," In: Proceedings of the Third International Conference on Genetic Algorithms, J.D. Schaffer (ed.), Morgan Kaufmann, San Mateo CA., USA, 434-439.

Temeng, K.O., P.D. Schenelle and T.J. McAvoy, "Model predictive control of an industrial packed bed reactor using neural networks," *Journal of Process Control*, **5**(1), 19-27 (1995).

Ungar, L.H., B.A. Powell and S.N. Kamens, "Adaptive Networks for Fault Diagnosis and process control," *Computers and Chemical Engineering*, **14**(4/5), 561-572 (1990).

Ungar, L.H., "A Bioreactor Benchmark for Adaptive Network-based Process Control," In: *Neural Networks for Control*, W.T. Miller, R.S. Sutton and P.J. Werbos (eds.), The MIT Press, Cambrigde Massechusetts, 387-402 (1991).

Vautard, R., P. Yiou and M. Ghil, "Singular-spectrum analysis: A toolkit for short, noisy chaotic signals," *Physica D*, **58**, 95-126 (1992).

Walters, F.H., "Sequential simplex optimization," CRC Press, Florida, 55-60 (1991).

Weeks, E.R., and J.M. Burgess, "Evolving Artificial Neural Networks to Control Chaotic Systems," *Physical Review E*, **56**(2), 1531-1540 (1997).

Whitehead, S.D., and L. Lin, "Reinforcement Learning of Non-Markov Decision Processes," *Artificial Intelligence*, **73**, 271-306.

Whitley, D., S. Dominic, R. Das and C.W. Anderson, "Genetic Reinforcement Learning for Neurocontrol Problems," *Machine Learning*, **13**, 259-284 (1993).

Whitley, D., T. Starkweather, "Optimizing Small Neural Networks using a Distributed Genetic Algorithm," In: *Proceedings of 1990 International Joint Conference on Neural Networks*, ?, Lawrence Erlbaum, Hillsdale NJ, 206-209 (1990).

Whitley, D., T. Starkweather and C. Bogart, "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity," *Parallel Computing*, **14**, 347-361.

Willis, M.J., G.A. Montague, C. Di Massimo, M.T. Tham and A.J. Morris, "Artificial Neural Networks in Process Estimation and Control," *Automatica*, **28**(6), 1181-1187 (1992).

Wills, M.J., C. DiMassimo, G.A. Montague, M.T. Tham, and A.J. Morris, "Artificial neural networks in process engineering," IEE Proceedings-D, 138(3), 257-266 (1991).

Wittgens, B., and Skogestad, S. (2000), "Closed Operation of Multivessel Batch Distillation: Experimental Verification," *AIChE Journal*, **46**(6), 1209-1217 (2000).

Yao, X., "Evolving Artificial Neural Networks," *Proceedings of the IEEE*, **87**(9), 1423-1447 (1999).

Ydstie, B.E., "Forecasting and Control using Adaptive Connectionist Networks," *Computers and Chemical Engineering*, **14**(4/5), 583-599 (1990).

Ylinen, R., T. Iivarinen and A.J. Niemi, "A Linear Quadratic-Gaussian Control Algorithm for Sulphide Ore Grinding," In: *Proceedings of the 4th IFAC Symposium on Automation in Mining, Mineral and Metal Processing*, T. Westerlund (ed.), Pergamon Press, Oxford, 199-205 (1983).

Zadeh, L.A., "Causality is Undefinable -- Toward a Theory of Hierarchical Definability," In: *Proceedings of the Third International Conference on Intelligent Processing and Manufacturing of Materials*, J.A. Meech, S.M. Veiga, M.M. Veiga, S.R. LeChair and J.F. Maguire (eds.), IPMM, Vancouver (2001).

Zheng, A., Mahajanam, R.V., and Douglas, J.M., "Hierarchical Procedure for Plantwide Control System Synthesis," *AIChE Journal*, **45**(6), 1255-1265 (1999).

Zhu, Y.-H., T. Rajalahti and S. Linko, "Application of neural networks to lysine production," *The Biochemical Engineering Journal*, **62**, 207-214 (1996).