Brain Inspired Cognitive Systems
August 29 – September 1, 2004
University of Stirling, Scotland, UK

A New Non-linear Multi-variable Multiple-Controller incorporating a Neural Network Learning Sub-Model

Ali S. Zayed and Amir Hussain
Department of Computing Science and Mathematics, University of Stirling, FK9 4LA, Scotland, UK
Author's E-mail: ahu@cs.stir.ac.uk

ABSTRACT

A new non-linear multi-variable multiple-controller incorporating a neural network learning sub-model is proposed. The unknown multivariable non-linear plant is represented by an equivalent stochastic model consisting of a linear time-varying sub-model plus a non-linear neural-network based learning sub-model. The proposed multiple controller methodology provides the designer with a choice of using either a conventional Proportional-Integral-Derivative (PID) self-tuning controller, a PID based pole-placement controller, or a newly proposed PID based pole-zero placement controller through simple switching. The novel PID based pole-zero placement controller employs an adaptive mechanism, which ensures that the closed loop poles and zeros are located at their pre-specified positions. The switching decision between the different non-linear fixed structure controllers can be done either manually or by using Stochastic Learning Automata. Simulation results using a non-linear Multiple Input Multiple Output (MIMO) plant model demonstrate the effectiveness of the proposed multiple controller, with respect to tracking set-point changes. The aim is to achieve a desired speed of response, whilst penalizing excessive control action, for application to non-minimum phase and unstable systems.

INTRODUCTION

Conventional PID controllers have been proven to be robust, easy to implement using analogue or digital hardware, and remarkably effective in regulating a wide range of processes. For most simple processes, PID control can provide satisfactory closed loop performance. However, the problems of large time-delays, time-varying processes, large non-linearities and non-negligible disturbances call for more advanced control algorithms. In the last two decades much progress has been seen in the theory of self-tuning and other adaptive control systems, which automatically adjust controller parameters online in response to changes in the process or the environment.

Over a longer period of about three decades, a lot of attention was paid to the problem of designing pole-placement controllers and self-tuning regulators. Various self-tuning controllers based on classical pole-placement ideas were developed and employed in real applications, e.g. [1, 2].

The popularity of pole-placement techniques may be attributed to the following [3]:

1) In the regulator case they provide mechanisms to over-come the restriction to minimum-phase plants of the original minimum variance self-tuner of [1].
2) In the servo case, they provide the ability to directly introduce bandwidth and damping ratio as tuning parameters.

In many industrial sectors, machines and processes might be improved using optimal control and optimisation (e.g., the steel industry, food industry, chemical industry, textile industry). A difficult problem in the control of these industrial processes is due to the inherent non-linearities of their models and these problems cannot be solved by traditional "generalised minimum variance control" techniques. The application of linear control theory to these problems relies on the key assumption of a small range of operation in order for the linear model assumption to be valid. When the required operating range is large, a linear controller may not be adequate. For this reason, it seems appropriate to extend "generalised minimum variance" control to plants with non-linear models and with plant/model mismatch. A possible way this can be achieved is by incorporating the inherent non-linearity of the process into the control design process using a so-called *learning model*.

Over the last decade or so, there has been much progress in the modelling and control of non-linear processes, using black-box type learning models, such as neural networks [4, 5]. This is due to their proven ability to learn arbitrary non-linear mappings. Other advantages inherent in neural networks include their robustness, parallel architecture and fault tolerant capabilities. Neural networks have been shown to be very effective for controlling complex non-linear systems, when there is no complete model information, or when the controlled plant is considered to be a "black box" [5].

In the following a new control algorithm is developed which builds on the works of Zayed *et al*. [3, 6] and Zhu *et. al*. [7], in which the unknown non-linear plant is represented by an equivalent model, consisting of a linear sub-model plus a non-linear sub-model. Models of a similar type have previously been shown to be particularly useful in an adaptive pole-placement based control framework by Zhu *et al*.[7]. In this work, the parameters of the linear sub-model are identified by a standard recursive identification algorithm, and in addition a conventional multi-layered neural network is utilized as the non-linear learning sub-model (see figure (1)).

The main contribution of this paper is in the development of a new learning sub-model based multiple controller, which provides the designer with a choice of using either a multivariable PID self-tuning controller, a PID pole-placement controller, or a newly proposed multivariable PID pole-zero placement controller through switching. All controllers operate using the same adaptive procedure. The switching (transition) decisions between the different non-linear fixed structure controllers can be achieved by using the Stochastic Learning Automata of [8]. However, in the present case, a manual switching process has been adopted.

The paper is organized as follows: the derivation of the control law of a multiple controller is discussed in section 2. In section 3, a simulation case study is carried out in order to demonstrate the effectiveness of the proposed controller. Finally, some concluding remarks are presented in section 4.

2. DERIVATION OF THE CONTROL LAW

In deriving the multivariable control law for the proposed multiple controller it is assumed that the plant can be described by the following n input n output non-linear Hammerstein model [6, 7]:

$$\mathbf{A}(z^{-1})\mathbf{y}(t+k) = \mathbf{B}(z^{-1})\mathbf{u}(t) + \mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U}) \qquad (1)$$

where $\mathbf{y}(t) = [y_1(t), y_2(t), \ldots\ldots, y_n(t)]^T$ is the measured output vector with dimension $(n\times 1)$ and $\mathbf{u}(t) = [u_1(t), u_2(t), \ldots\ldots, u_n(t)]^T$ is the control input vector $(n\times 1)$ at the sampling instant $t = 1,2,\ldots$, and k is the integer-sample dead time of the process. The term $\mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U})$ is potentially a non-linear function and $\mathbf{y}(t)\in\mathbf{Y}$, and $\mathbf{u}(t)\in\mathbf{U}$. The resulting MIMO model is a combination of a linear time varying sub-model plus a non-linear sub-model as shown in figure (1). The polynomial matrices $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ are respectively of orders n_a and n_b. $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ are $(n\times n)$ polynomial matrices and expressed in terms of backwards shift operator, z^{-1}

$$\left.\begin{aligned} \mathbf{A}(z^{-1}) &= diag(1 + a_1^{ii} z^{-1} + \ldots. + a_{n_a}^{ii} z^{-n_a}) \\ \mathbf{B}(z^{-1}) &= diag(b_0^{ii} + b_1^{ii} z^{-1} + \ldots\ldots. + b_{n_b}^{ii} z^{-n_b}) \end{aligned}\right\} \qquad (2)$$

where $b_0^{ii} \neq 0$ and $i = 1,2,\ldots\ldots., n$.

The equivalent sub model is originated by assuming that no coupling relationship exits. The coupling effects and the other non-linear relationships are meanwhile accommodated in $\mathbf{f}_0(\mathbf{Y},\mathbf{U})$ [7].

In order simplify the analysis, the time delay is taken as $k = 1$ [6, 7]. For this case the multivariable non-linear system given by (1) can be written as:

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = z^{-1}\mathbf{B}(z^{-1})\mathbf{u}(t) + z^{-1}\mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U}) \qquad (3)$$

The generalised minimum variance controller of interest minimises the following cost function:

$$\begin{aligned}\mathbf{J} = \mathbf{E}\{[\mathbf{P}(z^{-1})\mathbf{y}(t+1) + \Delta\mathbf{Q}(z^{-1})\mathbf{u}(t) - \\ (\mathbf{R}(z^{-1})\mathbf{w}(t) + \mathbf{H}(z^{-1}))\mathbf{f}_{0,t}(.,.)]\end{aligned} \qquad (4)$$

where $\mathbf{w}(t)$ is the set point and $\mathbf{P}(z^{-1})$, $\mathbf{Q}(z^{-1})$, $\mathbf{R}(z^{-1})$ and $\mathbf{H}(z^{-1})$ are user-defined transfer functions in the backward shift operator z^{-1} and $\mathbf{E}\{.\}$ is the expectation operator.

Now we can define an auxiliary output $f(t+1) = \mathbf{P}y(t+1)$, where the optimal predictor $f^*(t+1/t)$ is derived in [7, 9]:

$$\begin{aligned} f^*(t+1/t) = \mathbf{F}'(z^{-1})\mathbf{y}(t) + \\ \Delta\mathbf{E}'(z^{-1})\mathbf{B}(z^{-1})\mathbf{u}(t) + \Delta\mathbf{E}'(z^{-1})\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (5)$$

The relation (5) can be obtained by using (6) and multiplying (1) by $\Delta\mathbf{E}'(z^{-1})$.

The $(n\times n)$ polynomial matrices $\mathbf{E}'(z^{-1})$ and $\mathbf{F}'(z^{-1})$ satisfy the following identity [9]:

$$\mathbf{P}(z^{-1}) = \mathbf{A}(z^{-1})\Delta\mathbf{E}'(z^{-1}) + z^{-1}\mathbf{F}'(z^{-1}) \qquad (6)$$

where $\mathbf{E}'(z^{-1})$, $\mathbf{F}'(z^{-1})$ and $\mathbf{P}(z^{-1})$ are $(n\times n)$ polynomial matrices in z^{-1}. where the orders of the polynomial matrices $\mathbf{E}'(z^{-1})$, $\mathbf{F}'(z^{-1})$ and $\mathbf{P}(z^{-1})$ are specified as follows:

$$\left.\begin{aligned} & n_{e'} = k-1 \;\text{ and }\; n_{f'} = n_a \\ & n_p = \max(n_a + n_{e'} + 1,\;\; k + n_{f'}) \end{aligned}\right\} \qquad (7)$$

Now we can see clearly from (4) and (5) that the solution which minimises $\mathbf{J}$ is then found to be:

$$\begin{aligned} f^*(t+1/t) = \mathbf{R}(z^{-1})\mathbf{w}(t) - \Delta\mathbf{Q}(z^{-1})\mathbf{u}(t) + \\ \mathbf{H}(z^{-1})\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (8)$$

Using (5) and (8), the non-linear control law which minimises $\mathbf{J}$ is obtained as:

$$\begin{aligned}\Delta(\mathbf{E}'(z^{-1})\mathbf{B}(z^{-1}) + \mathbf{Q}(z^{-1}))\mathbf{u}(t) = [\mathbf{R}(z^{-1})\mathbf{w}(t) - \\ \mathbf{F}'(z^{-1})\mathbf{y}(t) + (\mathbf{H}(z^{-1}) - \Delta\mathbf{E}'(z^{-1}))\mathbf{f}_{0,t}(.,.)]\end{aligned} \qquad (9)$$

If we assume: $\mathbf{H}(z^{-1}) = \Delta(\mathbf{H}_0(z^{-1}) + \mathbf{E}'(z^{-1}))$ (10)

and set the transfer function $\mathbf{Q}(z^{-1})$ such that the following relation is satisfied:

$$(\mathbf{E}'(z^{-1})\mathbf{B}(z^{-1}) + \mathbf{Q}(z^{-1})) = [\mathbf{V}]^{-1}\mathbf{q}'(z^{-1}) \qquad (11)$$

Then equation (9) becomes:

$$\Delta\mathbf{q}'\mathbf{u}(t) = \mathbf{VRw}(t) - \mathbf{VF}'\mathbf{y}(t) + \Delta\mathbf{VH}_0\mathbf{f}_{0,t}(.,.) \qquad (12)$$

where $\mathbf{V} = diag(v^{ii})$ is an $(n \times n)$ diagonal gain matrix [2, 10] and $\mathbf{q}'(z^{-1})$ is a diagonal polynomial matrix in z^{-1} and having the following form:

$$\mathbf{q}'(z^{-1}) = diag(1 + q_1'^{ii} z^{-1} + ...+ q_{n_{q'}}'^{ii} z^{-n_{q'}}) \qquad (13)$$

It can clearly be seen from (11) that the polynomial $\mathbf{q}'(z^{-1})$ and the gain $\mathbf{V}$ are also user defined parameters since they depend on the user transfer function $\mathbf{Q}(z^{-1})$. It is clear from (10) that $\mathbf{H}_0$ is also a user defined parameter because it depend on the transfer function $\mathbf{H}(z^{-1})$.

We can see clearly from (12) that the controller denominator has now conveniently been split into two parts:

1) An integrator action part (Δ) required for PID design.
2) An arbitrary compensator ($\mathbf{q}'$) that may be used for pole-placement and pole-zero placement design.

2.1 Multiple Controller Mode 1: Self-tuning PID controller

In this mode, the so-called multiple controller operates as a conventional self-tuning PID controller, which can be expressed in the most commonly used velocity form [2, 10] as:

$$\begin{aligned}\Delta\mathbf{u}(t) = \mathbf{K}_I\mathbf{w}(t) - [\mathbf{K}_P + \mathbf{K}_I + \mathbf{K}_D]\mathbf{y}(t) - \\ [-\mathbf{K}_P - 2\mathbf{K}_D]\mathbf{y}(t-1) - \mathbf{K}_D\mathbf{y}(t-2)\end{aligned} \qquad (14)$$

If we assume that the degree of $\mathbf{F}'(z^{-1})$ is equal to 2 and set

$$\mathbf{q}'(z^{-1}) = \mathbf{I} \text{ and } \mathbf{R}(z^{-1}) = \mathbf{F}'(1) \qquad (15)$$

and make use of (14), (15) and (12), a non-linear self-tuning controller with PID structure is obtained, where:

$$\begin{aligned}\Delta\mathbf{u}(t) = \mathbf{VR}(z^{-1})\mathbf{w}(t) - \mathbf{VF}'(z^{-1})\mathbf{y}(t) + \\ \mathbf{VH}_0(z^{-1})\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (16)$$

$$\left.\begin{aligned}\mathbf{K}_P &= -diag(v^{ii} f_1'^{ii} + v^{ii} 2 f_2'^{ii}) \\ \mathbf{K}_I &= diag(v^{ii} f_0'^{ii} + v^{ii} f_1'^{ii} + v^{ii} f_2'^{ii}) \\ \mathbf{K}_D &= diag(v^{ii} f_2'^{ii})\end{aligned}\right\} \qquad (17)$$

The polynomial matrix $\mathbf{F}'(z^{-1})$ is computed from (6) by selecting suitable user-defined polynomial matrix $\mathbf{P}(z^{-1})$ and $i = 1,2,......., n$. It can also clearly be seen from (7) that the order of $\mathbf{F}'(z^{-1})$, which indicates the type of the controller (PI or PID), is governed by the polynomial matrix $\mathbf{A}(z^{-1})$. If the polynomial matrix $\mathbf{F}'(z^{-1})$ is of first order then a PI controller is obtained. A PID controller occurs if the polynomial matrix $\mathbf{F}'(z^{-1})$ is of second order. In what follows, the z^{-1} notation will be omitted from the various polynomials to simplify the presentation.

Substituting for $\mathbf{u}(t)$ given by (12) into the process model described by (3) and making use of equations (6) and (18), obtain:

$$\begin{aligned}(\Delta\mathbf{A} + z^{-1}\mathbf{VBF}')\mathbf{y}(t) = z^{-1}\mathbf{VBRw}(t) + \\ z^{-1}(\mathbf{V}\Delta\mathbf{H}_0\mathbf{B} + \Delta)\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (18)$$

To eliminate the effect, in the steady state, of the non-linear part the transfer function $\mathbf{H}_0$ can then be chosen as:

$$\mathbf{H}_0 = -\mathbf{q}'(1)\mathbf{V}^{-1}\mathbf{B}^{-1}(1) \qquad (19)$$

The controller is tuned by a selection of the user-defined polynomial matrix $\mathbf{P}$ and the diagonal gain matrix $\mathbf{V}$. However, the main disadvantage of many PID self-tuning based minimum variance control designs (see for example [2, 10]) is that the tuning parameters must be selected using a trial and error procedure. Alternatively, these tuning parameters can be automatically and implicitly set on line by specifying the desired closed loop poles [2, 6, 10].

2.2 Multiple Controller mode 2: PID based Pole placement design

This form of the controller is obtained by setting (see equation (11)) $(\mathbf{E}'(z^{-1})\mathbf{B}(z^{-1}) + \mathbf{Q}(z^{-1})) = [\mathbf{V}]^{-1}\mathbf{q}'(z^{-1})$ instead and substitute for $u(t)$ given by (12) into the process model described by (3), we obtain

$$\begin{aligned}(\mathbf{q}'\Delta\mathbf{A} + z^{-1}\mathbf{VBF}')y(t+1) = \\ \mathbf{VBRw}(t) + \Delta(\mathbf{VH}_0\mathbf{B} + \mathbf{q}')\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (20)$$

If we assume $\overline{\mathbf{B}} = \mathbf{VB}$ and $\overline{\mathbf{A}} = \Delta\mathbf{A}$, then (18) becomes:

$$\begin{aligned}(\mathbf{q}'\overline{\mathbf{A}} + z^{-1}\mathbf{F}'\overline{\mathbf{B}})\mathbf{y}(t) = z^{-1}\mathbf{VBRw}(t) + \\ \Delta z^{-1}(\mathbf{VBH}_0 + \mathbf{q}')\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (21)$$

We can now introduce the identity:

$$(\mathbf{q}'\overline{\mathbf{A}} + z^{-1}\mathbf{F}'\overline{\mathbf{B}}) = \mathbf{T} \qquad (22)$$

where $\mathbf{T}(z^{-1})$ is the desired closed loop poles and zeros and $\mathbf{q}'(z^{-1})$ is the controller polynomial. For (22) to have a unique solution, the order of the regulator polynomials and the number of the desired closed loop poles have to be [3, 6]:

$$\left.\begin{aligned}n_{f'} &= n_{\bar{a}} - 1 = n_a \\ n_{q'} &= n_{\bar{b}} \text{ and } n_T \le n_{\bar{a}} + n_{\bar{b}}\end{aligned}\right\} \qquad (23)$$

where $n_{\bar{a}}$, $n_{\bar{b}}$, and $n_{q'}$ are the orders of the polynomials $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$ and $\mathbf{q}'$, respectively, n_T denotes the number of desired closed loop poles. Also, $n_{\bar{b}} = n_b$ and $n_{\bar{a}} = n_a + 1$. By using (21) and (22) obtain:

$$\mathbf{Ty}(t+1) = \mathbf{VBRw}(t) + \Delta(\mathbf{q}' + \mathbf{VH_0B})\mathbf{f}_0(.,.) \qquad (24)$$

If we assume $\mathbf{R} = \mathbf{F}'(1)$ and $\mathbf{H}_0 = -[\mathbf{VB}(1)]^{-1}\mathbf{q}'(1)$ (25)

then (24) becomes:

$$\begin{aligned}\mathbf{Ty}(t) = z^{-1}\mathbf{VBF}'(1)\mathbf{w}(t) + \\ \Delta z^{-1}(\mathbf{q}' - \mathbf{q}'(1)\mathbf{B}^{-1}(1)\mathbf{B})\mathbf{f}_{0,t}(.,.)\end{aligned} \qquad (26)$$

2.3 Multiple Controller mode 3: PID based Pole-zero placement design

In this controller mode, an arbitrary desired zeros polynomial can be used to reduce excessive control action which can result from set point changes when pole placement is used. If we assume

$$\mathbf{R} = \mathbf{q}_\mathrm{d}\mathbf{F}'(1)[\mathbf{q}_\mathrm{d}(1)]^{-1} \qquad (27)$$

Then equation (26) becomes:

$$\begin{aligned}\mathbf{Ty}(t) = z^{-1}\mathbf{VB}(\mathbf{q}_\mathrm{d})\mathbf{F}'(1)[\mathbf{q}_\mathrm{d}(1)]^{-1}\mathbf{w}(t) + \\ \Delta z^{-1}(\mathbf{q}' - \mathbf{q}'(1)\mathbf{B}^{-1}(1)\mathbf{B})\mathbf{f}_0(.,.)\end{aligned} \qquad (28)$$

where $\mathbf{q}_\mathrm{d}$ is the desired closed loop systems.
In this case:

$$\left.\begin{aligned}\mathbf{T} &= diag(1 + t_1^{ii} z^{-1} + \ldots\ldots + t_{n_T}^{ii} z^{-n_T}) \\ \mathbf{q}_\mathrm{d} &= diag(1 + q_{d_1}^{ii} z^{-1} + \ldots\ldots + q_{d\,n_{q_d}}^{ii} z^{-n_{q_d}})\end{aligned}\right\} \qquad (29)$$

In practice the order of $\mathbf{T}$ and $\mathbf{q}_\mathrm{d}$ are most of the time selected as 1 or 2 [3, 6]

It can be seen from (12) that the polynomial q' may be considered as an arbitrary compensator that can be used to achieve the desired closed loop system poles. Clearly, from (28) the closed loop poles and zeros are in their desired locations. Clearly from (6), (7), (9), (10), (11) and (12) the transfer functions $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{H}$ change at every sampling instant in order to satisfy the condition in (19), (22), (23), (25) and (27). However, it is not necessary to explicitly calculate these design polynomials [4, 6, 10]. This does of course suggest the cost index has time varying weightings in this problem.
As can be seen in figure (1), a recursive least squares algorithm is initially used to estimate the parameters $\mathbf{A}$ and $\mathbf{B}$ (equation (1)) of the linear sub-model. Then a Back Propagation (BP) network is used to approximate the non-linear part $\hat{f}_0$.

It is clear from equation (3) that the i^{th} measured output $y_{i(t)}$ can be obtained as follows:

$$y_i(t+1) = \varphi_i^T(t)\theta_i(t) + f_{0,i}(Y,U),\ i = 1,2,\ldots\ldots,n \qquad (30)$$

where $\mathbf{y}(t) = [y_1(t), y_2(t),\ldots\ldots, y_n(t)]^T$ is the measured output vector with dimension $(n\times1)$ and $\mathbf{u}(t) = [u_1(t), u_2(t),\ldots\ldots, u_n(t)]^T$ is control input vector.

where θ_i is the parameter vector and $\varphi_i \in \Re^m$ is the data vector as follows:

$$\left.\begin{aligned}\theta_i(t) &= [-a_{1,i},\ldots\ldots,-a_{n_a,i}, b_{0,i},\ldots\ldots, b_{n_b,i}]^T \\ \varphi_i^T(t) &= [y_i(t),\ldots., y_i(t-n_a), u_i(t),\ldots., u_i(t-n_b)]\end{aligned}\right\} \qquad (31)$$

Equation (3) and (30) can also be presented as:

$$y_i(t) = \hat{\hat{y}}_i(t) + \tilde{f}_{0,i}(.,.) \qquad (32)$$

It is clear from figure (1) that $\hat{\hat{y}}_i(t) = j_i^T(t)\hat{q}_i(t)$ is the linear sub-model output and $\tilde{f}_{0,i}(.,.) = y_i(t) - \hat{\hat{y}}_i(t)$ is the difference between the actual output $y_i(t)$ and the linear sub-model output $\hat{\hat{y}}_i(t)$.

From figure (1) we can also see clearly that $\tilde{f}_{0,i}(.,.)$ can be expressed as:

$$\tilde{f}_{0,i}(.,.) = \hat{f}_{0,i} + \hat{e}_i(t) \qquad (33)$$

From the above equation and figure (1) the estimation of non-linear function $\hat{f}_{0,i}$ is obtained using neural network, whereas the identification error $e_i(t)$ is used to update the weights and thresholds of the learning neural network. The neural networks used in the proposed control scheme are three-layered types. The schematic diagram of the i^{th} neural network is shown in figure (2). The estimated non-linear function $\hat{f}_{0,t}$ can be detected by respectively using the following equations:

$$\hat{f}_{0,i} = \hat{f}_0^i = (1 + \exp[-\beta_{2l}^i(\sum_{j=1} w_{2lj}^i o_{1j}^i + b_{2l}^i)])^{-1} \qquad (34)$$

$$S_1^i = \eta^i\beta_{2l}^i\,\hat{f}_0^i[\tilde{f}_0^i - \hat{f}_0^i][1 - \tilde{f}_0^i] \qquad (35)$$

$$w_{2lj}^i(t) = w_{2lj}^i(t-1) + S_1^i \qquad (36)$$

$$o_{1j}^i(t) = [1 + \exp(-\beta_{1j}^i \sum_{i'} w_{1ji'}^i x_{i'} + b_{1j}'^i)]^{-1} \qquad (37)$$

$$S_2^i = S_1^i w_{2lj}^i(t)[o_{1j}^i(1 - o_{1j}^i)] \text{ where } l = 1 \qquad (38)$$

$$w_{1ji'}^i(t) = w_{1ji'}^i(t-1) + S_2^i x_{i'}^i \qquad (39)$$

where $w_{1ji'}$ and b_{1j} are the weights and thresholds between the input layer and the hidden layer, and w_{2lj} and b_{2l} are the weights and thresholds between hidden layer and the out put layer.

2.3 Non-linear Multiple-Controller Algorithm summary:

The proposed multiple controller algorithm can be summarised as:

Step 1. Select the desired closed-loop system poles and zeros polynomials $\mathbf{T}$ and $\mathbf{q}_d$.

Step 2. Select $\mathbf{P}$ and $\mathbf{V}$ for conventional PID control.

Step 3. Read the new values of $\mathbf{y}(t)$ and $\mathbf{w}(t)$.

Step 4. Estimate the process parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ using least squares algorithm.

Step 5. Compute $\hat{\mathbf{F}}'$ and $\hat{\mathbf{q}}'$ using (22). If we want to switch to the conventional PID controller, then use the polynomial $\mathbf{P}$ and $\mathbf{V}$ selected in step 2, set $\mathbf{q}'=1$, compute $\mathbf{F}'$ using equation (6) and then set $\mathbf{R}=\mathbf{F}'(1)$.

Step 6. Compute $\mathbf{R}$ using (25) for PID pole-placement. In the situation of using Pole-zero placement $\mathbf{R}$ is computed using (27).

Step 7. Compute $\mathbf{H}_0$ using (19) for all controllers.

Step 8. Compute $\tilde{\mathbf{f}}_0(.,.)=\mathbf{y}(t)-\hat{\tilde{\mathbf{y}}}(t)$, where $\hat{\tilde{\mathbf{y}}}(t)$ is the output of the linear sub-model.

Step 9. Apply the BP learning network to obtain $\hat{\mathbf{f}}_0(.,.)$ by using equations (34-39).

Step 10. Compute the control input using (12).

Steps 1 to 10 are to be repeated for every sampling instant.

3. SIMULATION RESULTS

The objective of this section is to study the performance and the robustness of the closed loop system using the techniques proposed in Section 2. A simulation case study will be carried out in order to demonstrate the ability of the proposed algorithm to locate the closed-loop poles and zeros at their desired locations under set point changes.

In chemical process industries one of the commonly occurring control problems is the control of fluid levels in storage tanks or reaction vessels. In this example the proposed controller was applied to a real world system model shown in Fig. 3 and described in [3].

The main objective of the control problem is to adjust the inlet flows f_{L1} and f_{L2} as to maintain the levels in the two tanks h_1 and h_2 as close to a desired set point. The fluid flow rates into tank 1 (f_{L1}) and thank 2 (f_{L2}) are supplied by two pumps.

To measure these flow rates, two flow meters are inserted between pumps and tanks. The flow of water from tank 2 to the reservoir (f_{L0}) is controlled by an adjustable tap. The maximum diameter of this tap is 0.70 cm. The depth of fluid is measured using parallel track depth sensors which are located in tank 1 and 2.

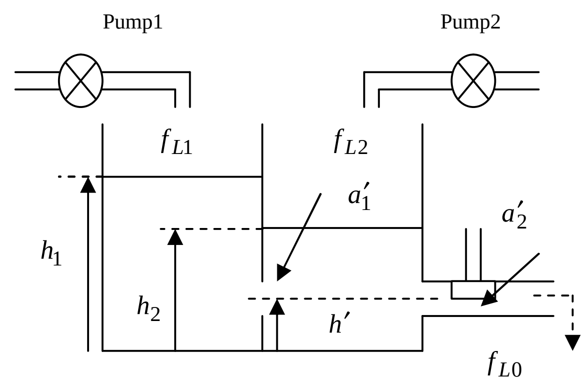

Figure 3. Coupled tank system.

The non-linear model can be presented as follows:

$$A\frac{dh_1}{dt}=f_{L1}-a_1'\sigma_1\sqrt{2g(h_1-h_2)}$$

$$A\frac{dh_2}{dt}=f_{L2}+a_1'\sigma_1\sqrt{2g(h_1-h_2)}-a_2'\sigma_2\sqrt{2g(h_2-h')}$$

where a_1' and a_2', are respectively the cross section area of orifice 1 and cross section area of orifice 2, and A is cross-sectional area of tank 1 and tank 2. s_1 and σ_2 are the discharge coefficient (0.6 for a sharp edged orifice) and $g=9.81N/m^2$. The diameter of orifice 1 is adjusted to 0.317cm and drain valve is fully open.

This simulation example is performed over 600 samples with the set point changing every 100 sampling instants as follows:

1) $\mathbf{w}_1(t)$ changes from 0.13m to 0.15m and from 0.15m to 0.13m.

3) $\mathbf{w}_2(t)$ changes from 0.08m to 0.1m and from 0.1m to 0.08m.

A first-order non-linear model is used. It is clear from equation and (10) that a PI control structure is achieved.

In this simulation only switching from PI pole-placement to PI pole-zero placement is considered.

In order to demonstrate clearly the closed loop performance of the multiple controller, we manually arrange the multiple-controller to work in the following two control modes, namely as a PI pole-placement controller and as a PI pole-zero placement controller, as described below:

a) The PI pole placement controller is switched on from 0th to 250th sampling times.

b) The PI pole-zero placement controller is switched on from 251th to 600th sampling times.

The closed loop poles and zeros are respectively selected as:

$$\mathbf{T} = \mathbf{I} + \begin{bmatrix} -0.5 & 0 \\ 0 & -0.7 \end{bmatrix} z^{-1} \text{ and } \tilde{\mathbf{h}} = \mathbf{I} + \begin{bmatrix} 0.8 & 0 \\ 0 & 0.85 \end{bmatrix} z^{-1}.$$

The closed-loop system outputs h_1 and h_2 (in meters) are shown in the Fig. 4a, whereas, the control inputs f_{L1} and f_{L2} in (m^3/sec) are respectively shown in Fig. 4b and Fig. 4c.

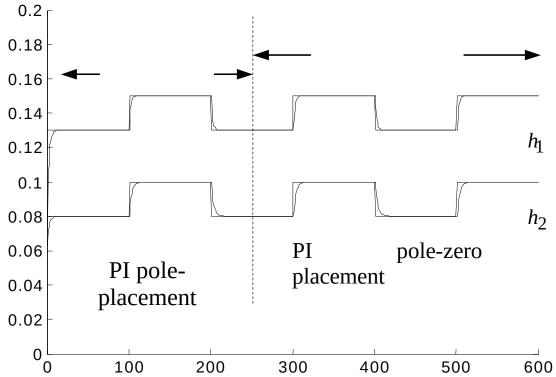

Figure 4a. The out puts h_1 and h_2.

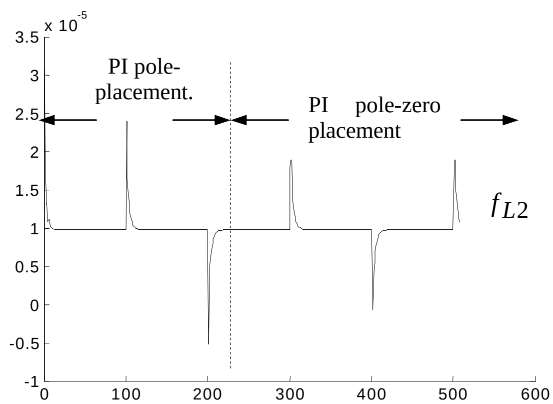

Figure 4c. The control input f_{L2}.

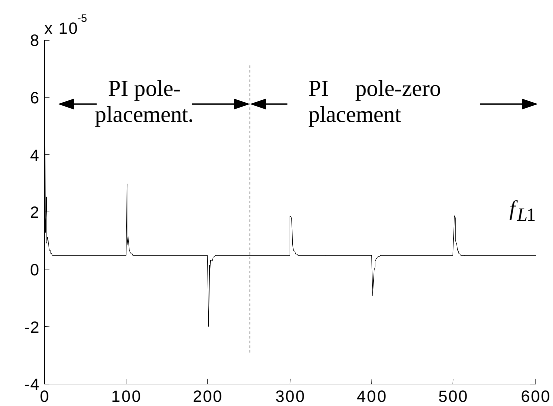

Figure 4b. The control input f_{L1}.

It is clear from these figures (4a), (4b) and (4c) that, the transient response is significantly shaped by the choice of the polynomial $\mathbf{T}$ when either a PI pole-placement controller or a PI zero-pole placement is used. It can also clearly be seen from figure (4b) and (4c) that excessive control action, which resulted from set-point changes, is tuned most effectively when the new PI zero-pole placement controller is on-line (during the sampling interval 251-600)

4 Concluding Remarks

A new robust non-linear self-tuning PID multiple-controller incorporating a neural network based learning sub-model for multivariable system has been described. The design, which exhibits enhanced robustness over similar techniques, overcomes certain limitations exhibited in other self-tuning PID control designs. Nonetheless, it retains the simplicity of adaptive mechanisms used elsewhere (e.g. [2, 10]). The proposed methodology provides the designer with the choice of using a conventional PID self-tuning controller, a PID pole placement controller or the proposed PID pole-zero placement controller. All of these controllers operate using the same adaptive procedure. The switching (transition) decision between these different fixed structure controllers was achieved here manually. In the future it will be implemented by using Stochastic Learning Automata of [8].

In the proposed design the unknown non-linear plant is represented by an equivalent model composed of a simple linear sub-model plus a non-linear sub-model. The parameters of the linear sub-model are identified by a standard recursive least squares algorithm, whereas the non-linear sub-model is approximated using a multi-layer BP neural network. The results indicate that of the proposed multiple controllers, the PID pole-zero placement controller best tracks set point changes with the desired speed of response. This penalises the excessive control action most effectively, and it can also deal with non-minimum phase systems. The controller's transient response is shaped by the choice of the pole polynomial $\mathbf{T}(z^{-1})$, while the zero polynomial $\mathbf{q}_{\mathrm{d}}(z^{-1})$ can be used to reduce the magnitude of control action, or to achieve better set point tracking [3, 6], compared to the computationally less expensive PID pole-placement and conventional self-tuning PID controllers.

ACKNOWLEDGEMENTS

This work is funded by the UK Engineering & Physical Sciences Research Council (EPSRC) Cluster Project *Towards Multiple Model based Learning Control Paradigms for Complex Systems* (Research Grant Ref. GR/S63779/01). The authors are also grateful to Prof. Mike Grimble of the University of Strathclyde in Glasgow for his extremely helpful comments and suggestions.

REFERENCES

[1] K. J. Astrom and B. Wittenmark, *On self-tuning regulators*, Automatica, 9 (1973) 185-199

[2] R. Yusof, S. Omatu and & M. Khalid, *Self-tuning PID control: a multivariable derivation and application*, Automatica, 30 (1994) 1975-1981

[3] A. Zayed, A. Hussain and L. Smith, *A New Multivariable Generalised Minimum-variance Stochastic Self-tuning with Pole-zero Placement*, International Journal of Control and Intelligent Systems, 32 (1) (2004) 35-44

[4] A. Hussain, J. Soraghon and T. Durrani *A new adaptive Functional-link Neural Network Based DFE for Overcoming Co-Channel Interference*, IEEE Transactions on Communications, 45 (10) (1997) 1358-1362

[5] W. Yu and A. Pozilyak, *Indirect adaptive control via parallel dynamic neural networks*, IEE. Proc. Control theory Appl., 146 (4) (1999) 25-30

[6] A. Zayed, A. Hussain and L. Smith, *A New Non-linear Self-tuning pole-zero placement incorporating Neural Networks*, Proceedings of the third International NASO Symposium on Engineering of Intelligent Systems Conference, Malaga Spain, 24-27 Dec 2002, pp 107.paper no 130 in CD.

[7] Q. Zhu, Z. Ma and K. Warwick, *Neural network enhanced generalised minimum variance self-tuning controller for nonlinear discrete-time systems*, IEE Proc. Contol Theory Appl., 146(4) (1999) 319-326

[8] Narendra K., Thathachar M.A.L., *Learning Automata, An Introduction*, Prentice-Hall, 1989.

[9] M. Tokuda and T. Yamamoto, 2002, A *neural-Net Based Controller supplementing a Multiloop PID Control System*, IEICE Trans. Fundamentals, Vol.E85-A, No. 1, 256-261

[10] A. Zayed, *MinimumVariance Based Adaptive PID Control Design*, M.Phil Thesis, Industrial Control Centre, University of Strathclyde, Glasgow, U.K., 1997.

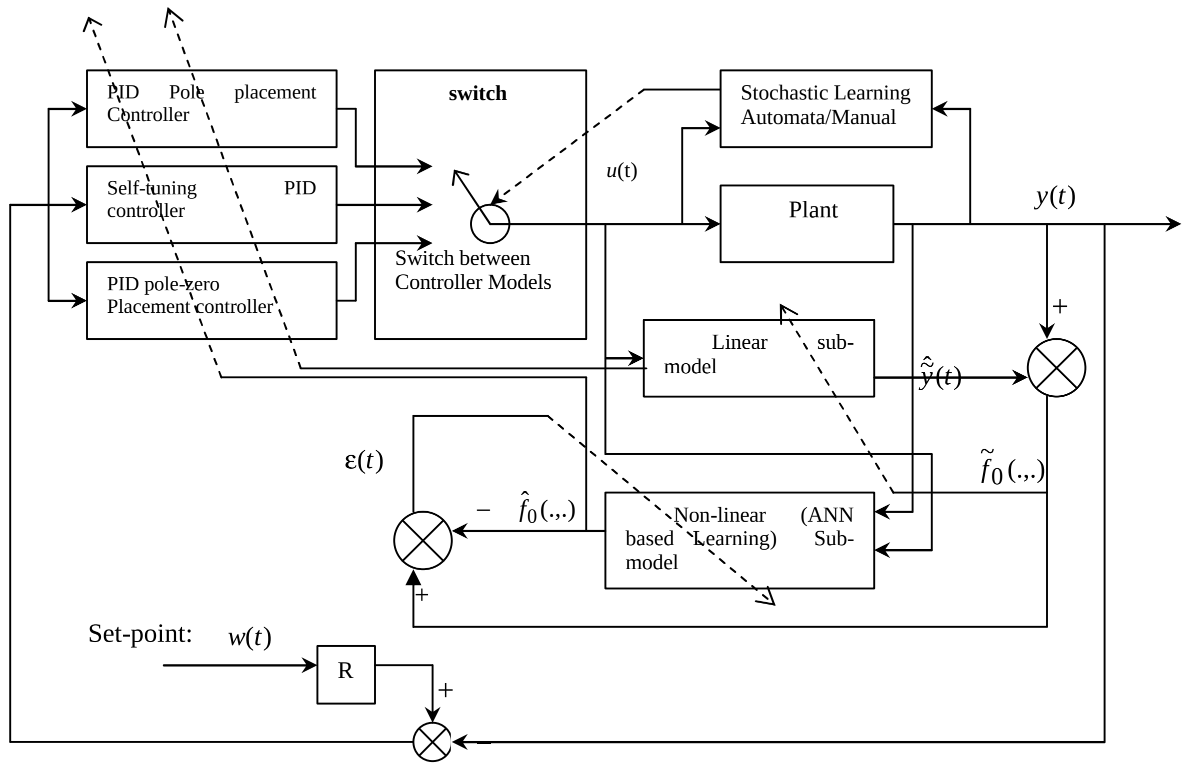

Figure 1: Non-linear Multiple Controller incorporating artificial neural networks (ANN) based learning sub-model

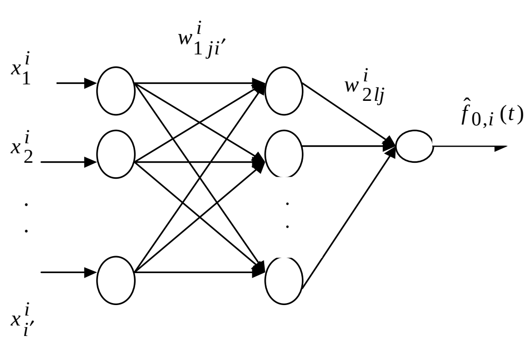

Figure 2: ANN learning sub-model to approximate NL function $\hat{f}_0$