A NEW NEURAL NETWORKS BASED MULTIVARIABLE NONLINEAR PID POLE-ZERO PLACEMENT CONTROL DESIGN

Ali S. Zayed and Mahmood Elfandi*

Department of Electrical & Electronic Engineering, Zawia University, Sabrata, Libya
Email: Zayed_1964@yahoo.com
*Department of Electrical& Electronic Engineering, University of Tripoli, Libya.
Email: m.elfandi@ee.edu.ly

الملخص

تم في هذه الورقة البجثية عرض متحكم جديد من المتحكمات تلقائية التعديل للانظمة الغير خطية المتعددة المدخلات والمخرجات. المتحكم مبني على ما يسمى بالنمودج التدريبي العام وهو نموذج رياضي يشتمل على نموذج فرعي خطي ونموذج أخر غير خطي. حيث يتم الحصول على النموذج الغير خطي باستخدام الشبكة العصبية لاستخدامه في تعويض نقص الاداء الذي قد ينتج عن بعض المكونات غير الخطية المصاحبة للنطام المراد التحكم فيه.

الهدف الرئيسى من هذه الورقة هو تطوير متحكم يتضمن كل من الشبكات العصبية والمسيطر التناسبي والتفاضلي والتكاملي للانظمة المتعددة المدخلات والمخرجات بالاضافة الى مقدرته على التحكم في تموضع الاصفار والأقطاب. تم اختيار المسيطر المقترح على نمودج منظومة خزان مزدوج ومقارنة النتائج المتحصل عليها من المحكاءة مع بعض المسيطرات الاخرى.

ABSTRACT

In this paper, a new non-linear self-tuning controller, which is based on the Generalised Learning Model (GLM), is presented. The GLM comprises a linear sub-model plus a non-linear neural-networks based learning sub-model. The non-linear part is accommodated in the control law design so that the non-linearity is compensated effectively. The main contribution of this paper is in the development of using a new neural networks control scheme for designing an adaptive pole-zero placement controllers with a PID structure for class of multivariable non-linear GLM plants. Example simulation results using nonlinear multivariable double tank system is considered. Convergence results are provided under certain assumptions and simulation studies are used to demonstrate the effectiveness of the proposed algorithm compared to previous works.

INTRODUCTION

Since the inception of self-tuning control (STC) by Astrom and Wettenmark [1], and Clark and Gawthrop [2], many forms of self-tuning controllers have been presented in the literature based on linear minimum variance strategy. Linear control methods are frequently applied in non-linear control problems [3, 4]. This is usually done using an adaptive framework and designing the control law on the basis of a linearised model of the controlled system, which is re-estimated periodically to fit the actual operating point. The popularity of this methodology is due to the existence of well-established recursive identification techniques, which allow online estimation of 1inear models. However, such adaptive approaches may fail if abrupt changes in the operation point are requested by the control specifications or determined by the specific non-linearity of the system [4]. In addition, the adaptive control system performance is difficult to analyse in general terms since it strongly depends on the dynamics of the process. Therefore, it is necessary to accommodate the non-linear parts explicitly in the controller design so that they are compensated effectively [5].

For this reason, it seems appropriate to extend predictive control strategy 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 *General learning model* (GLM) [6].

GLM model has been shown to be a useful representation for a large class of engineering plants.

In such designs, which have been proposed by Zhu and Warwick [5] and which we term the General Learning Model (GLM), the process model can be split into two parts, namely linear and non-linear dynamical learning sub-model, so that this special structure allows the linear part of the controller to exploit classical linear theory. In addition, the coupling effects and the other relationships are accommodated in the non-linear learning sub-model allowing effective compensation.

In this paper, a new non-linear controller algorithm is developed which exploits the benefits of both the PID and pole-zero placement controllers within a GLM framework. The proposed design builds on the previous works of Zayed *et al.* [6, 7, 8], by investigating the applicability of the GLM based multiple-controller methodology for complex multiple input multiple output (MIMO) plants. The parameters of the linear MIMO sub-model are identified by a standard recursive identification algorithm, and a conventional multi-layered neural network is utilised as the non-linear learning MIMO sub-model.

The paper is organised as follows: the derivation of the control law for the proposed PID pole-zero placement controller is discussed in section II. In section III, the Generalized Learning Models (GLM) for identification of MIMO plantsIn is described. In section IV, a detailed simulation case study using multivariable nonlinear double tank system model is carried out in order to demonstrate the effectiveness of the proposed controller with respect to tracking set point changes. Finally, some concluding remarks are presented in section V.

DERIVATION OF CONTROL LAW

Consider the following representation for n input and n output nonlinear multivariable plant model [5, 6, 9]:

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

where $\mathbf{y}(t) = [y_1(t), y_2(t), ..., ...y_n(t)]$ is the measured output vector with dimension ($n \times 1$), $\mathbf{u}(t) = [u_1(t), u_2(t), ..., ...u_n(t)]$ is the control input vector with dimension ($n \times 1$) and $\xi(t) = [\xi_1(t), \xi_2(t), ..., ...\xi_n(t)]$ is uncorrelated sequence of random variables with zero mean vector with dimension ($n \times 1$) at the sampling instant $t = 1,2,...$, and k is the time delay of the process in the integer-sample interval. The term $\mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U})$ in equation (1) above, is potentially a non-linear function (which accounts for any unknown time-delays, uncertainty and non-linear coupling effects in the complex MIMO plant model). The overall MIMO plant model represented by equation (1) above, is also known as the Generalized Learning Model (GLM) [6], and can be seem as the combination of a linear sub-model and a non-linear (learning) sub-model as shown in Figure (1).Also, in equation (1), we define $\mathbf{y}(t) \in \mathbf{Y}$, and $\mathbf{u}(t) \in \mathbf{U}$; $\{\mathbf{Y} \in R^{n_a}; \mathbf{U} \in R^{n_b}\}$ and $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ as ($n \times n$) diagonal polynomial matrices with orders n_a and n_b, respectively, which can be expressed in terms of the backwards shift operator, z^{-1} as:

$$\mathbf{A}(z^{-1}) = diag(1 + a_1^{ii} z^{-1} + \ldots,\ldots + a_{n_a}^{ii} z^{-n_a}) \tag{2a}$$

$$\mathbf{B}(z^{-1}) = diag(b_0^{ii} + b_1^{ii} z^{-1} + \ldots,\ldots + b_{n_b}^{ii} z^{-n_b}) \tag{2b}$$

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

It can be seen from equations (2a) and (2b) that the equivalent linear MIMO sub-model is devised by assuming that no coupling relationships exist. As stated earlier, the coupling effects and the other non-linear relationships are meanwhile accommodated in the nonlinear part $\mathbf{f}_{0,t}(.,.)$.

In order to simplify the analysis, the time delay is taken as $k = 1$ [5, 9]. For this case the non-linear system represented by equation (1) can be written as [5, 6, 9]:

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

The generalised minimum variance controller of interest minimises the following cost function [5, 6]:

$$J_N = \mathrm{E}\{[\boldsymbol{\phi}(t+1)]\} \tag{4}$$

where: $\phi(t+1) = [\mathbf{P}(z^{-1})\mathbf{y}(t+1) + \mathbf{Q}(z^{-1})\mathbf{u}(t) - \mathbf{R}(z^{-1})\mathbf{w}(t) - \mathbf{H}_N(z^{-1})\mathbf{f}_{0,t}(.,.)]$ (5)

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

Next, we can introduce the following identity [6, 10, 11]:

$$\mathbf{P}_n(z^{-1}) = \mathbf{A}(z^{-1})\mathbf{E}(z^{-1})\mathbf{P}_d(z^{-1}) + z^{-1}\mathbf{F}(z^{-1}) \tag{6}$$

where $\mathbf{E}(z^{-1})$, $\mathbf{F}(z^{-1})$, $\mathbf{P}_n(z^{-1})$ and $\mathbf{P}_d(z^{-1})$ are ($n \times n$) diagonal polynomial matrices in z^{-1}.

where $\mathbf{P}_n(z^{-1})$ and $\mathbf{P}_d(z^{-1})$ are the numerator and denominators of the polynomial matrix $\mathbf{P}(z^{-1})$.

The orders of the polynomial matrices $\mathbf{E}(z^{-1})$, $\mathbf{F}(z^{-1})$ and $\mathbf{P}_n(z^{-1})$ in the equations (6) are specified as follows:

$$\left.\begin{aligned} n_e &= k-1 \\ n_f &= (n_{P_d} + n_a - 1) \\ n_{p_n} &= \max(n_a + n_{p_d} + \\ &\qquad n_e,\ k + n_f) \end{aligned}\right\} \tag{7}$$

where, n_f, n_{p_n} and n_{p_d} represents the degrees of $P_d(z^{-1})$, $\mathbf{P}_n(z^{-1})$ and $\mathbf{P}_d(z^{-1})$ respectively.

Multiplying (3) by $\mathbf{P}_d(z^{-1})\mathbf{E}(z^{-1})$ and substitute for $(\mathbf{E}(z^{-1})\mathbf{A}(z^{-1}))$ from equation (6) gives:

$$[\mathbf{P}_d]^{-1}\mathbf{P}_n\mathbf{y}(t+1) = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{Q}(z^{-1})\mathbf{u}(t) - \mathbf{R}(z^{-1})\mathbf{w}(t) - \mathbf{H}_N(z^{-1})\mathbf{f}_{0,t}(.,.)] \tag{8}$$

Adding $\mathbf{Q}(z^{-1})\mathbf{u}(t) - \mathbf{R}(z^{-1})\mathbf{w}(t) - \mathbf{H}_N(z^{-1})\mathbf{f}_{0,t}(.,.)$ to both sides of equation (8) and using equation (6), yields:

$$\begin{aligned}\phi(t+1) &= [\mathbf{P}_d(z^{-1})]^{-1}\mathbf{F}(z^{-1})\mathbf{y}(t) + (\mathbf{Q}(z^{-1}) + \mathbf{B}(z^{-1})\mathbf{E}(z^{-1}))\mathbf{u}(t) - \mathbf{R}(z^{-1})\mathbf{w}(t) + \\ &\quad (\mathbf{E}(z^{-1}) - \mathbf{H}_N(z^{-1}))\mathbf{f}_{0,t}(.,.) + \mathbf{E}(z^{-1})\xi(t+1)\end{aligned} \tag{9}$$

In the rest of this section, the argument z^{-1} will be omitted from various polynomials and transfer functions in order to simplify the notation and only be used where required for clarification purposes.

Now we can define the optimal predictor $\phi^*(t+1|t)$ and the prediction error $\tilde{\phi}(t+1|t)$ as follows:

$$\phi_y^*(t+1|t) = [\mathbf{P}_d]^{-1}\mathbf{F}\mathbf{y}(t) + (\mathbf{Q} + \mathbf{E}\mathbf{B})\mathbf{u}(t) + [\mathbf{E} - \mathbf{H}_N]\mathbf{f}_{0,t}(.,.) \tag{10}$$

$$\tilde{\boldsymbol{\phi}}(t+1|t) = \mathbf{E}\boldsymbol{\xi}(t+1) \tag{11}$$

If we set $\boldsymbol{\phi}^*(t+1|t) = 0$ in equation (10) and after some arrangement, the generalised minimum variance control law for non-linear systems is obtained as:

$$\mathbf{P}_d(\mathbf{E}\mathbf{B} + \mathbf{Q})\mathbf{u}(t) = [\mathbf{P}_d\mathbf{R}\mathbf{w}(t) - \mathbf{F}\mathbf{y}(t) + \mathbf{P}_d(\mathbf{H}_N - \mathbf{E})\mathbf{f}_{0,t}(.,.)] \tag{12}$$

Now, if we set:

$$\left.\begin{aligned}\mathbf{R} &= [\mathbf{P}_d]^{-1}\mathbf{H}_0 \\ \mathbf{H}_N &= ([\mathbf{P}_d]^{-1}\Delta\mathbf{H}'_N + \mathbf{E})\end{aligned}\right\} \tag{13}$$

next, if we set the transfer function $\mathbf{Q}(z^{-1})$ such that the following relation is satisfied [6, 10, 11]:

$$\mathbf{P}_d(\mathbf{E}\mathbf{B} + \mathbf{C}\mathbf{Q}) = \mathbf{V}^{-1}\Delta\mathbf{q}' \tag{14}$$

then, equation (12) becomes:

$$\Delta\mathbf{q}'\mathbf{u}(t) = [\mathbf{V}\mathbf{H}_0\mathbf{w}(t) - \mathbf{V}(\mathbf{F})\mathbf{y}(t) + \Delta\mathbf{V}\mathbf{H}'_N\mathbf{f}_{0,t}(.,.)] \tag{15}$$

where $\mathbf{V}$ is a user defined gain matrix [6, 10, 11] and $\mathbf{q}'$ is a polynomial in z^{-1} having the following form:

$$\mathbf{V} = diag(v^{ii}) \tag{16a}$$

$$\mathbf{q}'(z^{-1}) = diag(1 + q_1'^{ii}z^{-1} + \ldots\ldots + q_{n_{q'}}'^{ii}z^{-n_{q'}}) \tag{16b}$$

where $n_{q'}$ is the degree of the polynomial $\mathbf{q}'$ and $i = 1,\ldots.n$.

We can see clearly from equations (14) and (15) 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 zero placement design.

It can be seen from equation (14) that the polynomial matrix $\mathbf{q}'(z^{-1})$ and the gain matrix $\mathbf{V}$ are user-defined parameters since they depend on the user transfer function $\mathbf{Q}(z^{-1})$. It is also clear from equation (13) that $\mathbf{H}_0$ and $\mathbf{H}'_N$ are user-defined parameter because they depend on the transfer functions $\mathbf{R}(z^{-1})$ and $\mathbf{H}_N$ respectively.

Now, if we set:

$$\mathbf{H}_0 = \widetilde{\mathbf{H}}[\widetilde{\mathbf{H}}(1)]^{-1}\mathbf{F}(1) \tag{17}$$

and combine equations (17) and (15), then we can readily obtain:

$$\Delta\mathbf{q}'\mathbf{u}(t) = \mathbf{V}\widetilde{\mathbf{H}}[\widetilde{\mathbf{H}}(1)]^{-1}\mathbf{F}(1)\mathbf{w}(t) - \mathbf{V}(\mathbf{F})\mathbf{y}(t) + \Delta\mathbf{V}\mathbf{H}'_N\mathbf{f}_{0,t}(.,.) \tag{18}$$

where $\widetilde{\mathbf{H}}$ in equation (18) is a user-defined polynomial which can be used to introduce arbitrary closed loop zeros for explicit pole-zero placement controller and has the following form:

$$\widetilde{\mathbf{H}}(z^{-1}) = diag(1 + \tilde{h}_1^{ii}z^{-1} + ... + \tilde{h}_{n_{\tilde{h}}}^{ii}z^{-n_{\tilde{h}}}); \tag{19}$$

The above equation (18) represents the final expression of the control law for the proposed non-linear MIMO multiple controller.

Multiple Controller Mode 1: Conventional non-linear 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 [10, 11] as:

$$\Delta\mathbf{u}(t) = \mathbf{K_I}\mathbf{w}(t) - [\mathbf{K_P} + \mathbf{K_I} + \mathbf{K_D}]\mathbf{y}(t) - [-\mathbf{K_P} - 2\mathbf{K_D}]\mathbf{y}(t-1) - \mathbf{K_D}\mathbf{y}(t-2) \tag{20}$$

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

$$\mathbf{F}(z^{-1}) = diag(f_0^{ii} + f_1^{ii}z^{-1} + f_2^{ii}z^{-1}) \tag{21}$$

and switch both of pole-placement polynomial $\mathbf{q}'$ given by equation (16b) and zero-placement polynomial $\widetilde{\mathbf{H}}$ given by (19) off by setting:

$$\left.\begin{array}{l} \mathbf{q}' = diag(1), (\text{i.e. } q_1'^{ii} = q_2'^{ii} = ..,.. = q_{n_{q'}}^{ii} = 0) \\ \widetilde{\mathbf{H}} = diag(1), (\text{i.e.} \tilde{h}_1^{ii} = \tilde{h}_2^{ii} = ...,... = \tilde{h}_{n_{\tilde{h}}}^{ii} = 0) \end{array}\right\} \tag{22a}$$

next, if we set:

$$\mathbf{H}'_N = -(\mathbf{B}(1)\mathbf{V})^{-1}\mathbf{q}'(1) \tag{22b}$$

then a self-tuning controller with PID structure is obtained, where

$$\Delta\mathbf{u}(t) = [\mathbf{V}\mathbf{F}(1)\mathbf{w}(t) - \mathbf{V}(\mathbf{f}_0 + \mathbf{f}_1 z^{-1} + \mathbf{f}_2 z^{-1})\mathbf{y}(t) + \Delta\mathbf{V}\mathbf{H}'_N\mathbf{f}_{0,t}(.,.) \tag{23}$$

$$\mathbf{K_P} = -diag(v^{ii}f_1 + 2v^{ii}f_2) \tag{24a}$$

$$\mathbf{K_I} = diag(v^{ii} f_0^{ii} + v^{ii} f_1^{ii} + v^{ii} f_2^{ii}) \tag{24b}$$

$$\mathbf{K_D} = diag(v^{ii} f_2^{ii}) \tag{24c}$$

It can be seen from the above equations (23), (24a), (24b) and (24c) that the PID control parameters $\mathbf{K_P}$, $\mathbf{K_I}$ and $\mathbf{K_D}$ depend on the polynomial matrix $\mathbf{F}(z^{-1})$ and the gain matrix $\mathbf{V}$ [6, 10, 11]. In this case the parameters of the polynomial matrix $\mathbf{F}(z^{-1})$, f_0^{ii}, f_1^{ii} and f_2^{ii} are computed directly from the equation (6) by selecting suitable user-defined polynomials $\mathbf{P}_d$ and $\mathbf{P}_n$ which are selected in trial and error basis [10, 11].

It can also clearly be seen from equation (7) that the order of $\mathbf{F}(z^{-1})$ which indicates the type of the controller (PI or PID) is governed by the polynomial $\mathbf{A}$ and $\mathbf{P}_d$ [11, 12].

As stated above the multiple controller mode 1 described by equations (20)-(24c) is tuned by a selection of the polynomials $\mathbf{P}_n$ and $\mathbf{P}_d$, and the gain $\mathbf{V}$. However, the main disadvantage of many PID self-tuning based minimum variance control designs (see for example [10, 11]) 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 [5, 6, 12].

Multiple Controller Mode 2: Nonlinear PID based pole placement controller

Substituting for $\mathbf{u}(t)$ given by equation (18) into the process model described by equation (3), the closed loop system is obtained as:

$$(\tilde{\mathbf{A}}\mathbf{q}' + z^{-1}\tilde{\mathbf{B}}\mathbf{F})\mathbf{y}(t) = z^{-1}\mathbf{B}\mathbf{V}\tilde{\mathbf{H}}[\mathbf{H}(1)]^{-1}\mathbf{F}(1)\mathbf{w}(t) + \Delta(z^{-1}\mathbf{B}\mathbf{V}\mathbf{H}'_N + \mathbf{q}')\mathbf{f}_{0,t} + \Delta\mathbf{q}'\xi(t) \tag{25}$$

where

$$\left.\begin{array}{l}\tilde{\mathbf{A}} = \Delta\mathbf{A}\\ \tilde{\mathbf{B}} = \mathbf{V}\mathbf{B}\end{array}\right\} \tag{26}$$

We can now introduce the identity:

$$(\mathbf{q}'\tilde{\mathbf{A}} + z^{-1}\mathbf{F}\tilde{\mathbf{B}}) = \mathbf{T} \tag{27}$$

where $\mathbf{T}$ is the desired closed loop poles and $\mathbf{q}'$ is the controller polynomial. For equation (27) to have a unique solution, the order of the regulator polynomials and the number of the desired closed loop poles must be set as [6, 12]:

$$\left.\begin{array}{l}n_{f'} = n_{\tilde{a}} - 1 = n_a\\ n_{q'} = n_{\tilde{b}} + k - 1\\ n_t \le n_{\tilde{a}} + n_{\tilde{b}} + k - 1\end{array}\right\} \tag{28}$$

where $n_{\tilde{a}}$, $n_{\tilde{b}}$, and $n_{q'}$ are the orders of the polynomials $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ and $\mathbf{q}'$, respectively, and n_t denotes the number of desired closed loop poles. Also, $n_{\tilde{b}} = n_b$ and $n_{\tilde{a}} = n_a + 1$.

Combining equations (25) and (27), gives:

$$\mathbf{T}\mathbf{y}(t) = z^{-1}\mathbf{B}\mathbf{V}\tilde{\mathbf{H}}[\mathbf{H}(1)]^{-1}\mathbf{F}(1)\mathbf{w}(t) + \Delta(z^{-1}\mathbf{B}\mathbf{V}\mathbf{H}'_N + \mathbf{q}')\mathbf{f}_{0,t} + \Delta\mathbf{q}'\xi(t) \tag{29}$$

If the explicit zero placement polynomial given by (19) is switched off by setting:

$$\widetilde{\mathbf{H}} = diag(1), (\text{i.e.} \, \widetilde{h}_1^{ii} = \widetilde{h}_{2\;2}^{ii} = \ldots,\ldots = \widetilde{h}_{n_{\widetilde{h}}}^{ii} = 0) \tag{30}$$

And, if we set:

$$\mathbf{H}'_N = -(\mathbf{B}(1)\mathbf{V})^{-1}\mathbf{q}'(1) \tag{31}$$

then the closed loop function of equation (29) becomes:

$$\mathbf{Ty}(t) = z^{-1}\mathbf{BVF}(1)\mathbf{w}(t) + \Delta(z^{-1}\mathbf{BVq}'(1)[-(\mathbf{B}(1)\mathbf{V})^{-1}] + \mathbf{q}')\mathbf{f}_{0,t}\Delta\mathbf{q}'\xi(t) \tag{32}$$

In this case:

$$\mathbf{T}(z^{-1}) = diag(1 + t_1^{ii}z^{-1} + \ldots. + t_{n_t}^{ii}z^{-n_t}) \tag{33}$$

where $n_{\widetilde{h}}$ and n_t in equations (30) and (33) represent orders of the polynomials $\widetilde{\mathbf{H}}(z^{-1})$ and $\mathbf{T}(z^{-1})$ respectively.

It can be seen from equation (32) that the closed loop poles are placed at their desired positions which is pre-specified by the user through the use of the polynomial $\mathbf{T}(z^{-1})$.

Multiple Controller Mode 3: Nonlinear PID based zero-pole placement controller

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 the zero-placement polynomial ($\widetilde{\mathbf{H}}$) given by equation (19) is switched on then the closed loop given by equation (25) is again obtained and can be simplified as follows: then the closed loop function of equation (32) becomes:

$$\mathbf{Ty}(t) = z^{-1}\mathbf{BV}[\widetilde{\mathbf{H}}(1)]^{-1}\widetilde{\mathbf{H}}\mathbf{F}(1)\mathbf{w}(t) + \Delta(z^{-1}\mathbf{BVq}'(1)[-(\mathbf{B}(1)\mathbf{V})^{-1}] + \mathbf{q}')\mathbf{f}_{0,t} + \Delta\mathbf{q}'\xi(t) \tag{34}$$

Note that in practice, the order of $\mathbf{T}(z^{-1})$ and $\widetilde{\mathbf{H}}(z^{-1})$ are most of the time selected to equal 1 or 2 [9, 16].

It can be seen from equations (34) that the closed loop poles and zero are placed at their desired positions which pre-specified by using the polynomials $\mathbf{T}(z^{-1})$ and $\widetilde{\mathbf{H}}(z^{-1})$.

Clearly from equations (6)-(7) and (12)-(16) above, the user defined transfer functions $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{H}_N$ must change at every sampling instant in order to satisfy the conditions specified by equations (21), (22), (24a), (24b) and (24c) for achieving self-tuning PID control (multiple controller mode 1). On the other hand, the above under defined transfer functions must change in order to satisfy equations (27), (28) (30) (31) and (33) for achieving pole-placement control (multiple controller mode 2). Finally, for achieving pole-zero placement control (multiple controller mode 2) these user-defined transfer functions must change automatically in order to satisfy equations (27), (28), (31) and (19).

However, note that it is not necessary to explicitly calculate these user defined design transfer functions [6,10,11]. This does of course; suggest that 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 *A* and *B* (equation (3)) of the linear sub-model. Then a Back Propagation (BP) network is used to approximate the non-linear part $\hat{f}_{0,t}$.

In the next section, the identification of complex non-linear MIMO plants using the GLM framework is discussed.

Generalized Learning Models (GLM) for identification of MIMO plants

As can be seen in Figure (1), a recursive least squares (RLS) algorithm is initially used in the proposed multiple-controller framework in order to estimate the parameters *A* and *B* of the linear MIMO sub-model. As described in this section, a Back Propagation (BP) or multi-layered perceptron (MLP) neural network is then used to approximate the non-linear part $\mathbf{f}_{0,t}(\mathbf{Y},\mathbf{U})$ (which accounts for any uncertainty, time-varying and non-linear multi-variable interactions in the complex plant model)

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

$$y_i(t+1) = \varphi_i^T(t)\theta_i(t) + f_{0,i}(\mathbf{y},\mathbf{u}) \tag{35}$$

Where

$$i = 1,2,......,n \tag{36}$$

where $\mathbf{y}(t) = [y_1(t), y_2(t),......, y_n(t)]^T$ is the measured output vector with dimension $(n \times 1)$ and $\mathbf{u}(t) = [u_1(t), u_2(t),......, u_n(t)]^T$ is control input vector.

where $\boldsymbol{\theta}_i$ is the parameter vector and $\varphi_i \in \Re^m$ is the data vector as follows:

$$\left.\begin{array}{l} \theta_i(t) = [-a_{1,i},.......,-a_{n_{a,i}}, b_{0,i},......, b_{n_{b,i}}]^T \\ \varphi_i^T(t) = [y_i(t-1),..., y_i(t-n_a), u_i(t-1),..., u_i(t-n_b)] \end{array}\right\} \tag{37}$$

Equation (3) and (35) can also be presented as:

$$y_i(t) = \hat{\bar{y}}_i(t) + \bar{f}_{0,i}(.,.) \tag{38}$$

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

From Figure (1) it can be also seen that $\bar{f}_{0,i}(.,.)$ can be expressed as:

$$\bar{f}_{0,i}(.,.) = \hat{f}_{0,i} + \varepsilon_i(t) \tag{39}$$

Using the above equation (39) and as can be seen in Figure (1), a neural network is used for estimating the non-linear function $\hat{f}_{0,i}$, with the identification error $\varepsilon_i(t)$ being used to update the weights and thresholds of the learning neural network model. The neural network model employed in the proposed control scheme is chosen to be a three-layered type multi-layered perceptron (MLP). The schematic diagram of the i^{th} neural network is shown in Figure (2). The non-linear function $\hat{f}_{0,t}$ is adaptively estimated by using the following equations [5, 6]:

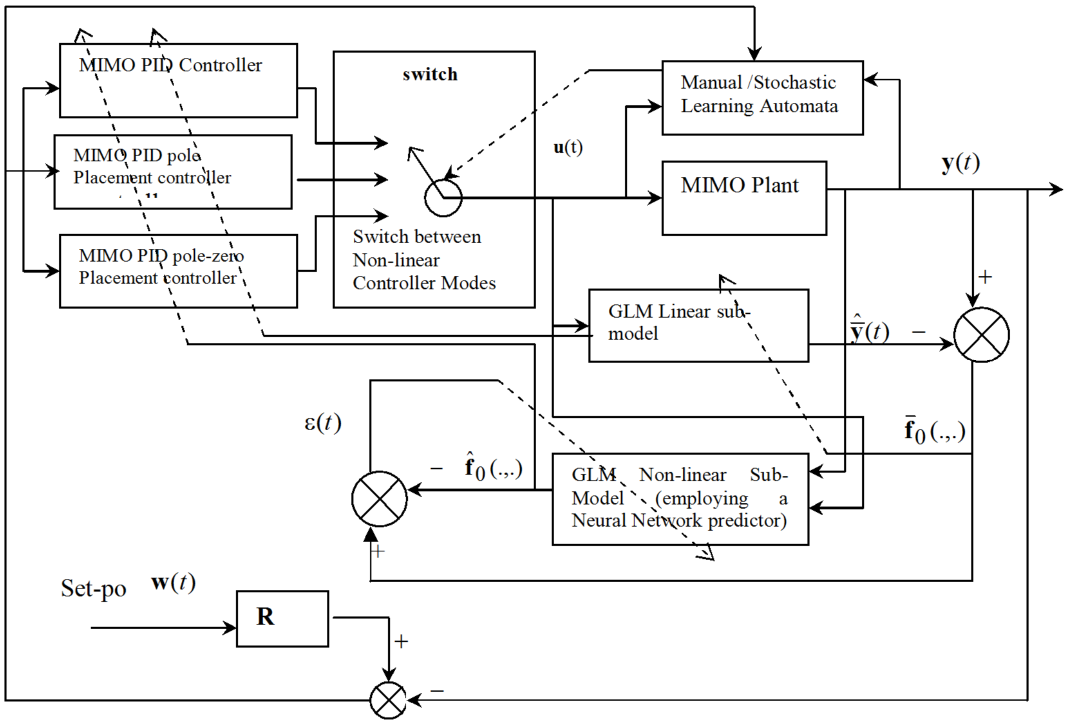
MIMO PID Controller
MIMO PID pole Placement controller
MIMO PID pole-zero Placement controller
switch
Switch between Non-linear Controller Modes
u(t)
Manual /Stochastic Learning Automata
MIMO Plant
y(t)
+
GLM Linear sub-model
ŷ(t)
−
f̄0(.,.)
ε(t)
− f̂0(.,.)
GLM Non-linear Sub-Model (employing a Neural Network predictor)
+
Set-po
w(t)
R
+
−

Figure 1: Proposed non-linear adaptive multiple controller framework incorporating a Neural Network based Generalized Learning Model (GLM) for MIMO systems

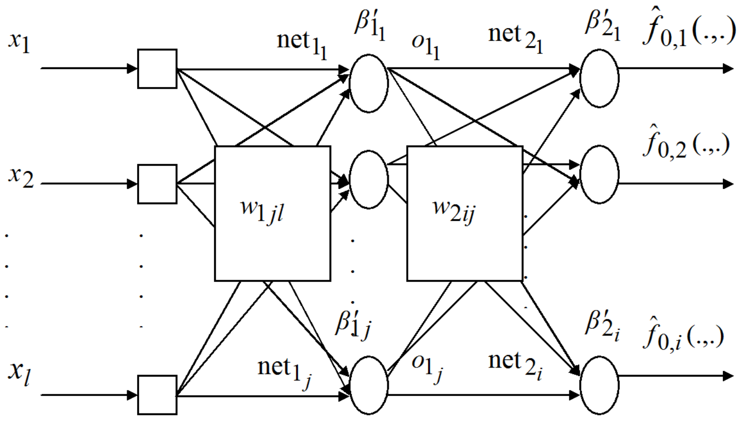

Figure 2: Neural network learning model to approximate the non-linear function $f_0(.,.)$

$$\hat{f}_{0,i} = \frac{1}{1+\exp[-\beta'_{2i}\mathrm{net}_{2_i}]} \tag{40}$$

$$\mathrm{net}_{2_i} = \sum_{j=1} w_{2ij}o_{1_j} + b'_{2_i} \tag{41}$$

$$o_{1_j}(t) = \frac{1}{1+\exp[-\beta'_{1j}\mathrm{net}_{1_j}]} \tag{42}$$

$$\mathrm{net}_{1_j} = \sum_{l=1} w_{2jl}(x_l) + b'_{1_j} \tag{43}$$

$$S_{1_i} = \eta\beta'_{2_i}\hat{f}_{0,i}[\bar{f}_{0,i} - \hat{f}_{0,i}][1 - \bar{f}_{0,i}]o_{1_j} \tag{44}$$

$$S_2 = \sum_{i=1}^{n} S_{1_i} w_{ij}[\beta'_{1_j}(1 - o_{1_j})]x_l \tag{45}$$

$$w_{2ji}(t) = w_{2ji}(t-1) + S_{1_i} \tag{46}$$

$$w_{1lj}(t) = w_{1lj}(t-1) + S_2 \tag{47}$$

where w_{1jl} and β'_{1_j} are the weights and activation factors between the input layer and the hidden layer, and w_{2ij} and β'_{2_j} are the weights and activation factors between hidden layer and the output layer, where b'_{2_i} is the threshold value for the i^{th} neuron in the output layer and b'_{1_j} is the threshold value for the j^{th} neuron in the hidden layer.

The overall adaptive multiple controller framework for MIMO systems is illustrated in Figure (1).

Nonlinear MIMO multiple-controller algorithm summary

The proposed multiple controller algorithm for MIMO systems can now be summarised as:

Step 1. Select the desired closed-loop system poles and zeros polynomials $\mathbf{T}$ and $\tilde{\mathbf{H}}$ (for explicit pole-zero control).

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

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

Step 4. Estimate the process parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ using a conventional exponentially weighted RLS algorithm [1,15].

Step 5. In order to switch to the conventional PID controller (multiple controller mode 1), then following steps must be followed:

Step (5.1) Compute $\hat{\mathbf{F}}$ from equation (6) using the polynomials $\mathbf{P}_d$ and $\mathbf{P}_n$ and the gain $\mathbf{V}$ selected in step 2.

Step (5.2) set $\hat{\mathbf{q}}' = diag(1)$, $\tilde{\mathbf{H}} = diag(1)$ and $\mathbf{H}'_N = -(\mathbf{B}(1)\mathbf{V})^{-1}\mathbf{q}'(1)$ using equations (22a) and (22b).

Step 6. In order to switch to the PID pole-placement controller (multiple controller mode 2) then compute $\hat{\mathbf{F}}$ and $\hat{\mathbf{q}}'$ using (26) and (27) and switch off the explicit zero placement polynomial by setting $\tilde{\mathbf{H}} = diag(1)$ using (30).

Step 7. In order to switch to the PID pole-zero placement controller (multiple control mode 3) then compute $\hat{\mathbf{F}}$ and $\hat{\mathbf{q}}'$ using (26) and (27), and switch on the pole-zero placement polynomial $\tilde{\mathbf{H}}$ using (19).

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

Step 9. Apply the BP learning network to obtain $\hat{\mathbf{f}}_0(.,.)$ by using equations (40)-(47).

Step 10. Compute the control input using (18).

Steps 3 to 10 are to be repeated for every sampling instant.

SIMULATION RESULTS

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

In chemical process industries, one of the most commonly occurring control problems is that of controlling the fluid levels in storage tanks or reaction vessels. In this example, the proposed controller is applied to a real world system model shown in Figure (3) and described in detail in [12]. The coupled tank system illustrated in Fig. 3 comprises one container with a centre partition resulting in two tanks. Both tanks are 10cm long, 10 cm deep and 30cm high. At the base of the partition four holes are provided to allow flow of water between the tanks. These holes are at the height of 3 cm (i.e. $h'_s = 0.03$ m) with different diameters of 1.27cm, 0.95cm, 0.635cm 0.317 and together form orifice 1, which is adjustable by plugging one or more of the holes. The main objective of the control problem is to adjust the inlet flows f_{L1} and f_{L2} in order to maintain the levels in the two tanks h_{s_1} and h_{s_2} as close to a desired set point. The fluid flow rates into tank 1 (f_{L1}) and tank 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.

The non-linear model can be presented as follows [12]:

$$A\frac{dh_{s1}}{dt} = f_{L1} - a'_1\sigma_1\sqrt{2g(h_{s1} - h_{s2})} \tag{48a}$$

$$A\frac{dh_{s2}}{dt} = f_{L2} + a'_1\sigma_1\sqrt{2g(h_{s1} - h_{s2})} - a'_2\sigma_2\sqrt{2g(h_{s2} - h'_s)} \tag{48b}$$

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. σ_1 and σ_2 are the discharge coefficient (0.6 for a sharp edged orifice) [15], and $g = 9.81 N/m^2$. The diameter of orifice 1 is adjusted to 0.95cm and drain valve is fully open.

The set point $\mathbf{w}(t)$ changes every 100 as follows:

1) $\mathbf{w}_1(t)$ changes from 0.08 m to 0.09 m and from 0.09 m to 0.08 m.

2) $\mathbf{w}_2(t)$ changes from 0.14 m to 0.15 m and from 0.15 m to 0.14 m.

A first order linear model $[\mathbf{I} + \hat{\mathbf{A}}_1 z^{-1}]\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = z^{-1}\hat{\mathbf{B}}_0\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$ is used to identify the parameters of the MIMO process using RLS estimator. Three-layer back propagation neural network shown in the Fig. 2 is used to approximate the non-linear function $(\mathbf{f}_{0,t}(.,.))$.

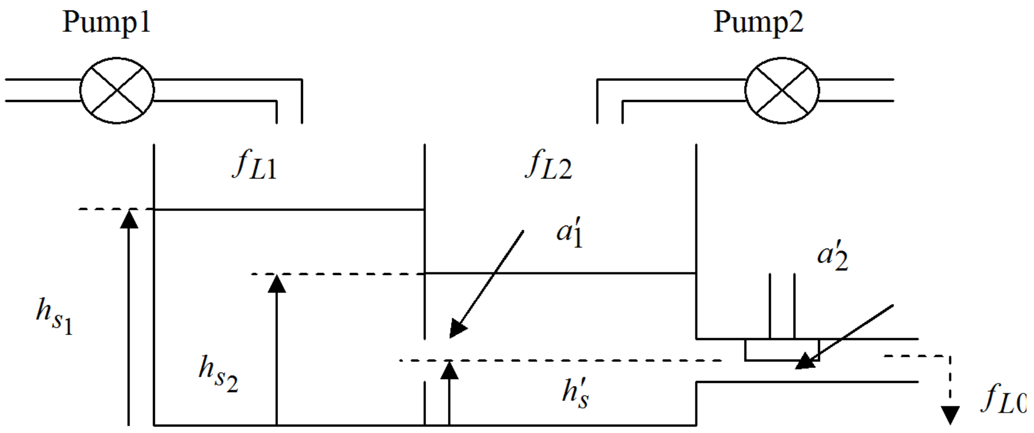

Figure 3: Coupled-tank system

In order to implement the multiple controller algorithm, the steps summarised in section (2.5) are followed. The linear sub-model is approximated by a first order linear model. It can be seen from equation (7), A PI controller is achieved, if $n_a = n_{p_d} = 1$. Whereas, the PID control structure is obtained by selecting the polynomial matrix $\mathbf{P}_d$ to be of order two. However, a PI controller seems to be more appropriate for like this type of systems. The user-defined gain matrix **V** and polynomial matrices $\mathbf{P}_d$ and $\mathbf{P}_n$ can initially be approximated using the systematic approach of [13] or the modified versions proposed by [11, 12].

Thus, the user defined gain matrix and the user-defined polynomial matrices are respectively selected as:

$\mathbf{P}_d(z^{-1}) = \mathbf{I} + \mathbf{P}_{d_1} z^{-1}$ and $\mathbf{P}_n(z^{-1}) = \mathbf{I} + \mathbf{p}_{n_1} z^{-1}$

$\mathbf{V} = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.001 \end{bmatrix}$, $\mathbf{P}_d(z^{-1}) = \mathbf{I} + \mathbf{P}_{d_1} z^{-1}$ and.

where $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{P}_{d_1} = \begin{bmatrix} -0.8 & 0 \\ 0 & -0.9 \end{bmatrix}$ and $\mathbf{P}_{n_1} = \begin{bmatrix} -0.5 & 0 \\ 0 & -0.6 \end{bmatrix}$.

A first order desired closed pole matrix **T** is selected to satisfy the time domain specification. The values of t_1^{11} and t_1^{22} depend on the parameters of the linear sub-model (for more details see [14]).

The closed loop poles and zeros are respectively selected as:

$\mathbf{T} = \mathbf{I} + \begin{bmatrix} -0.6 & 0 \\ 0 & -0.7 \end{bmatrix} z^{-1}$ and $\tilde{\mathbf{H}} = \mathbf{I} + \begin{bmatrix} 0.8 & 0 \\ 0 & 0.85 \end{bmatrix} z^{-1}$. It clear from equations (6), (7), (28), (24a), (24b) and (24c) that a PI control structure is achieved. It is also clear from equation (28) that the polynomial $\mathbf{q}'$ is of zero degree.

The neural networks learning model is shown in Figure (4). It can be seen from Figure (4) that, in the simulation experiment, the network was set up with the neuron complement as: $l = 4$, $j = 3$ and $i = 2$ for the input layer, hidden layer and output layer, respectively, and the learning rate of ($\eta = 0.02$). The size of network can be specified by the designer depending on the complicity of the non-linearity of the process. Also the weights and thresholds are initially selected to be less than one. The activation factors can be selected to be more than one. (for more details refer to reference [15]).

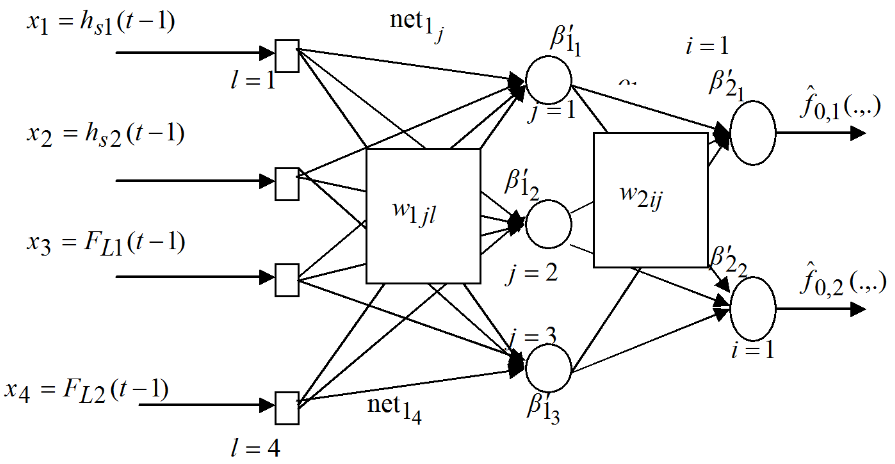

Figure 4: Neural network learning model for coupled tank system

The weights and activation factors between hidden layer and the output layer w_{2ij}, β'_{2_i} are selected as follows:

$$W_2 = \begin{bmatrix} w_{2_{11}} & w_{2_{12}} & w_{2_{13}} \\ w_{2_{21}} & w_{2_{22}} & w_{2_{23}} \end{bmatrix} = \begin{bmatrix} 0.05 & 0.02 & 0.1 \\ 0.03 & 0.12 & 0.06 \end{bmatrix} \text{and}\, \beta'_{2_1} = 2,\ \beta'_{2_2} = 1.9 .$$

The threshold values for the i^{th} neurons in the output layer are set as follows: $b'_{2_1} = 0.3$ and $b'_{2_2} = 0.4$. Where w_{2ij} is an element in the matrix W_2. Whereas, the weights and activation factors between hidden layer and the input layer w_{1jl}, β'_{1j} are selected as follows:

$$W_1 = \begin{bmatrix} w_{1_{11}} & w_{1_{12}} & w_{1_{13}} & w_{1_{14}} \\ w_{1_{21}} & w_{1_{22}} & w_{1_{23}} & w_{1_{24}} \\ w_{1_{31}} & w_{1_{32}} & w_{1_{33}} & w_{1_{34}} \end{bmatrix} \quad W_1 = \begin{bmatrix} 0.007 & 0.02 & 0.1 & 0.04 \\ 0.09 & 0.4 & 0.06 & 0.06 \\ 0.1 & 0.08 & 0.02 & 0.066 \end{bmatrix} \text{ and}$$

$\beta'_{1_1} = 1.8$ $\beta'_{1_2} = 2.06$ $\beta'_{1_3} = 2.2$

Where w_{1jl} is an element in the matrix (W_1).

The threshold values for the jth neurons in the input layer are set as follows: $b'_{1_1} = 0.025$ and $b'_{1_2} = 0.05$. In this simulation experiment the sigmoid function is used. In order to demonstrate the closed loop performance of the multiple controller we arrange manually the PI based multiple-controller to work in all three control modes: namely as a PI pole-placement controller, a PI pole-zero placement controller and as a PI self-tuning controller as described below:

a) From 0th up to 250th sampling time, the PI pole-placement controller is selected to operate on-line.

b) The PI pole-zero placement controller is switched on from 251st to 550th sampling time.

c) The conventional PI self-tuning controller is switched on from 551st to 800th sampling instant.

The closed-loop system outputs h_{s_1} and h_{s_2} (in meters) are shown in Figure (4a), whereas, the control inputs f_{L1} and f_{L2} in $(m^3/\sec)$ are respectively shown in the Figure (5b) and Figure (5C).

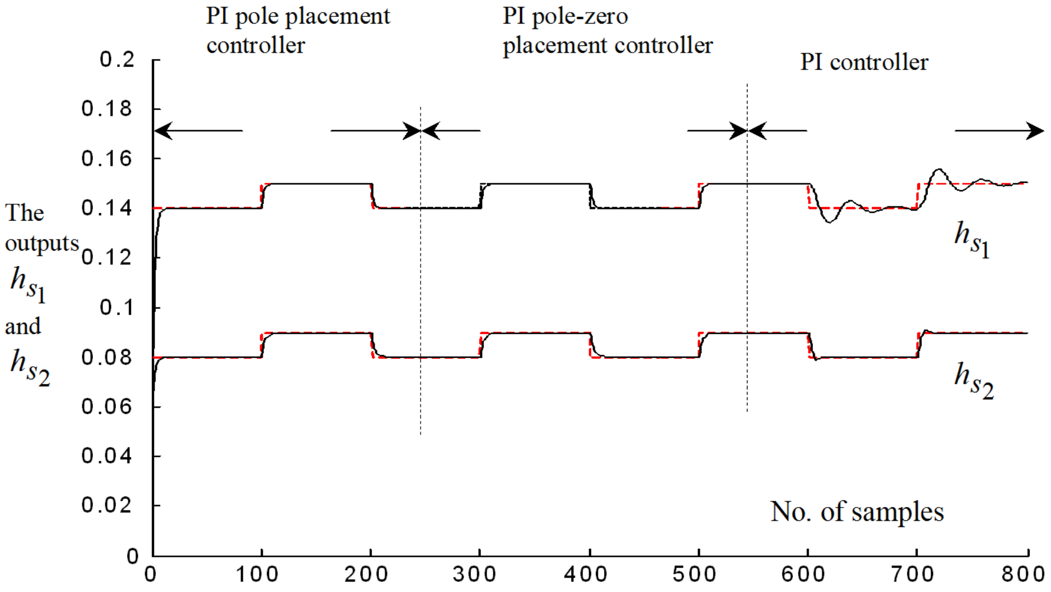

Figure: 5a. The outputs h_{s_1} and h_{s_2}

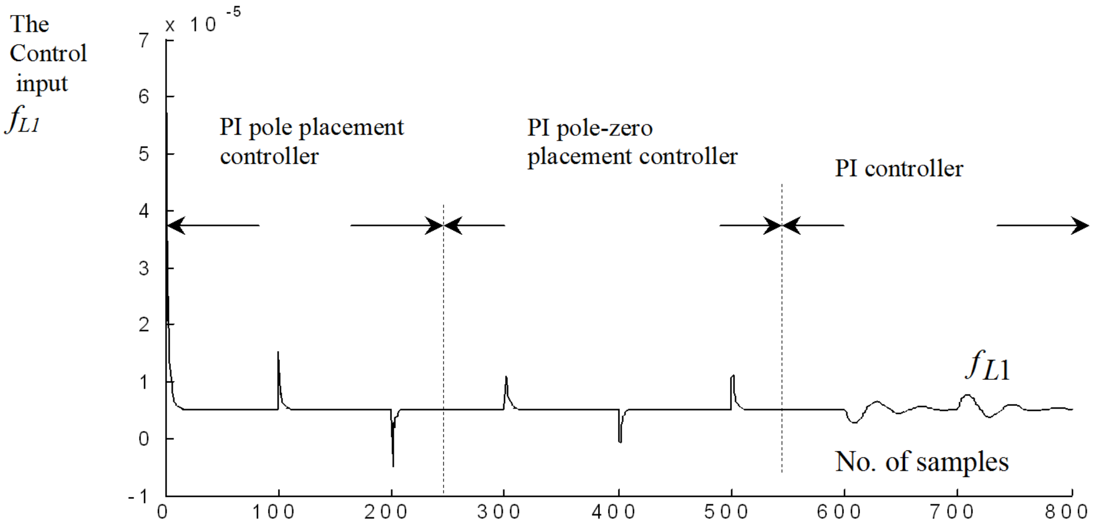

Figure 5b: The control input f_{L1}

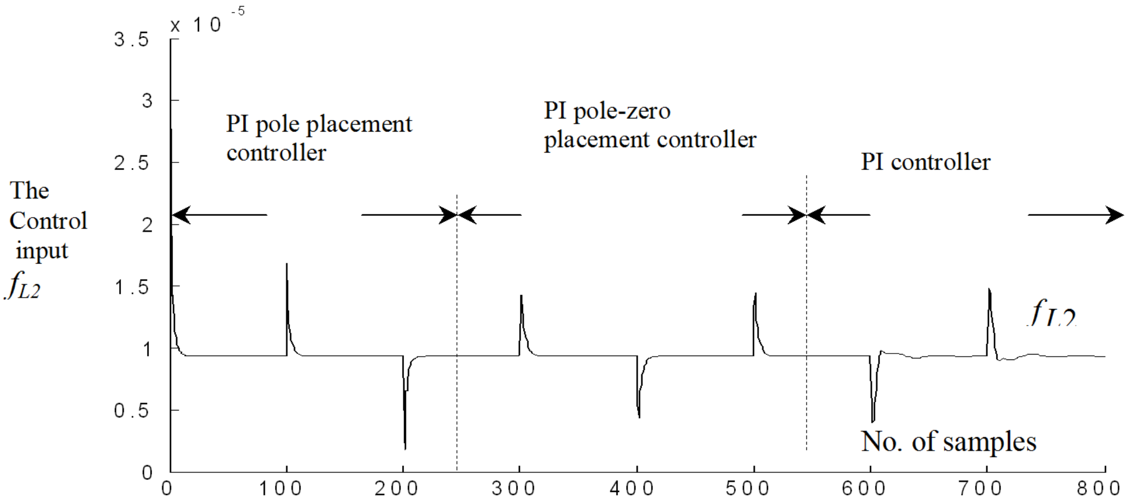

Figure 5c: The control input f_{L2}

It can be seen from these Figure (5a), Figure (5b) and Figure (5c) that, the transient responses are 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 Figure (4c) that excessive control action, which resulted from set-point changes, is tuned most effectively when the user selected PI zero-pole placement controller is operating on-line (during the sampling interval 251-550) Also note that during the last 250 samples (550-800 sampling times), where the conventional self tuning PI controller is operating, small oscillations can be seen in the control input and closed loop output, hence exhibiting the worst performance as expected, due to its inherent limitations. The other disadvantage of the self-tuning PID controller is that the tuning parameters must be selected using a trial and error procedure**.** The Final values of the PI pole-placement controller parameters **($\mathbf{K_P}$** and $\mathbf{K_I}$), and the pole-placement compensator matrices $\mathbf{q}'$ are as follows:

$\mathbf{K_P} = \begin{bmatrix} 0.0024 & 0.0001 \\ -0.0001 & 0.0026 \end{bmatrix}$, $\mathbf{K_I} = 10^{-3}\begin{bmatrix} 1 & 0 \\ 0 & 0.75 \end{bmatrix}$ and $\mathbf{q}' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Whereas, the final values of conventional PI controller matrices are obtained as follows:

$\mathbf{K_P} = 10^{-3}\begin{bmatrix} 0.08 & 0 \\ 0 & 0.9 \end{bmatrix}$ and $\mathbf{K_I} = 10^{-3}\begin{bmatrix} 0.05 & 0 \\ 0 & 0.4 \end{bmatrix}$.

The zero pole-placement PID control matrices are always identical to those of PID pole-placement control. However, an additional control action caused by introducing the closed-loop zeros polynomial matrix ($\tilde{\mathbf{H}}$) into the design is obtained.

The final updated weights between hidden layer and the output layer are expressed in the following matrix form: $W_2 = \begin{bmatrix} w2_{11} & w2_{12} & w2_{13} \\ w2_{21} & w2_{22} & w2_{23} \end{bmatrix} = \begin{bmatrix} 0.04 & -0.02 & 0.1 \\ 0.12 & 0.16 & 0.06 \end{bmatrix}$

The final updated weights between hidden layer and the input layer are:

$$W_1 = \begin{bmatrix} w_{1_{11}} & w_{1_{12}} & w_{1_{13}} & w_{1_{14}} \\ w_{1_{21}} & w_{1_{22}} & w_{1_{23}} & w_{1_{24}} \\ w_{1_{31}} & w_{1_{32}} & w_{1_{33}} & w_{1_{34}} \end{bmatrix} \quad W_1 = \begin{bmatrix} 0.01 & -0.002 & 0.1 & 0.4 \\ 0.09 & 0.014 & 0.06 & -0.16 \\ 0.1 & -0.03 & 0.2 & 0.036 \end{bmatrix}.$$

CONCLUSIONS

In this paper, a new non-linear multi-variable multiple-controller algorithm is developed which combines the benefits of both the PID and pole-zero placement controllers within a novel GLM framework. 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 developed controllers operate using the same adaptive procedure and are selected on the basis of the required performance measure. The switching (transition) decision between these different fixed structure controllers was achieved manually in the present case. In the future it will be implemented using the Stochastic Learning Automata criterion proposed by [16]. It was shown in equation (7) that a PI controller is achieved if ($n_f = 1$). Whereas, a PID controller is obtained by setting ($n_f = 2$). Without lose of the generality (because of the case steady considered in this work), the PI control structure is used. However, The PID control structure can be obtained by selecting $n_{p_d} = 2$.

A further research can also be performed to investigate the closed loop stability analysis. Sample simulation results using a nonlinear multivariable double tank plant model are used to demonstrate its effectiveness. Furthermore, the results indicate that the PID pole-zero placement controller option provides the best set point tracking performance with the desired speed of response, but at the expense of a relatively greater computational requirement (compared to the other controller options). This particular controller option 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 $\tilde{\mathbf{H}}(z^{-1})$ can be used to reduce the magnitude of control action, or to achieve better set point tracking [12], compared to the computationally less expensive PID pole-placement and conventional self-tuning PID controllers.

ACKNOWLEGMENT

The authors are indebted to Professor Amir Hussain of The University of Stirling, Stirling (UK) for highly stimulating discussions and his insightful comments and suggestions.

REFERENCES

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

[2] Clarke D. W and Gawthrop P.J., Self tuning control, Proc. Inst. Electrical Engineering, Part D, 126 (1979) 633-640.

[3] Tokuda M.and Yamamoto T., A neural-Net Based Controller supplementing a Multiloop PID Control System, IEICE Trans. Fundamentals, Vol. E85-A, 1, 2002, 256-261.

[4] Bittanti S.and Piroddi L., Neural implementation of GMV control schemes based on affine input/output models, IEE Proc. Contol Theory and Applications, 144(6), 1997, 521-530.

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

[6] Zayed A., Hussain A., Grimble M., A Non-linear PID-based Multiple Controller incorporating a Multi-Layered Neural Network Learning Sub-model, Control & Intelligent Systems, , 2006, Vol.34, No.3

[7] Elfllah, A., Zayed S.,and Elfandi, M., A new Adaptive PID Controller Incorporating A Radial Basis Function Neural Network for non-linear Systems, The fifth Libyan Arab International Conference On Electrical and Electronic Engineering, 23-26 October, 2010, Tripoli University, Libya, Pp 307-316.

[8] Zayed A., Elflah A., A. Gawedar, and M. Rizg, A new neural networks adaptive PID pole-placement controller for non-linear systems, 10th International Conference on Science and Techniques of Automatic control and computer Engineering, CD Proceeding, 2009, Hammamat, Tunisia, December, 20-22.

[9] Zhu Q., and Warwick, K. A neural network enhanced generalised minimum variance self-tuning proportional, integral and derivative control algorithm for complex dynamic systems, Journal of systems and Control Engineering, 2002, 216, part 1, 265-273.

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

[11] Yusof R.and Omatu, S. A multivariable self-tuning PID controller, Int. J. Control. 57 (1993) 1387-1403.

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

[13] Cameron F.and. Seborg, D. E A self-tuning controller with a PID structure, Int. J. Control, 38, 1983, 401-417.

[14] Wellstead] P. and Zarrop M., Self-tuning systems: Control and Signal Processing (John Wiley and Sons, U.K., 1995).

[15] Haykin S., Neural Networks: A comprehensive foundation, Second edition, (Prentice Hall, New Jersey, 1999)

[16] Narendra K.and Thathachar M.A.L., Learning Automata, An Introduction, (Prentice-Hall, 1989).

NOMENCLATURES

A, B	Polynomials of orders n_a and n_b, respectively.
$\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{B}}$	Polynomial matrices depending on **A**, and **B,** respectively.
$\mathbf{y}(t), \mathbf{u}(t), \mathbf{w}(t)$	System output, control input set-point vectors with dimension ($n \times 1$)
$\xi(t)$	Uncorrelated sequence of random variables with zero mean vector with dimension ($n \times 1$)
$\mathbf{f}_{0,t}(\mathbf{Y}, \mathbf{U})$	Non-linear function.

$t = 1,2,...$	Sampling instant.
k	System time delay.
$\mathbf{P}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{H}_N$, $\mathbf{H}_0$	Cost (user defined transfer functions).
$\mathbf{P}_n$, $\mathbf{P}_d$	User-defined polynomials of degrees n_{p_n} and n_{p_d} , respectively.
$\mathbf{E}$, $\mathbf{F}$, $\mathbf{q}'$	Controllers polynomial matrices.
$\mathrm{E}\{.\}$	Expectation operator.
J_N	Design cost function.
$\tilde{\mathbf{H}}$	Arbitrary closed loop zeros polynomial matrix.
$\mathbf{T}$	Arbitrary closed loop poles polynomial matrix.
$\phi(t)$	Generalised output function.
$\mathbf{\Delta}$	Difference operator.
$\mathbf{K_P}$, $\mathbf{K_I}$, $\mathbf{K_D}$	PID control parameters
$\mathbf{V}$	Gain matrix.
$\phi^*(t+1\|t)$	Optimal predictor.
$\tilde{\phi}(t+1\|t)$	Prediction error.
f_{L1} , f_{L2}	Inlet flows.
h_{s_1} , h_{s_2}	Levels in the two tanks.
w_{1jl} , β'_{1_j}	weights and activation factors between the input layer and the hidden layer.
w_{2ij} , β'_{2_i}	are the weights and activation factors between hidden layer and the output layer.
b'_{2_i}	Threshold value for the i^{th} neuron in the output layer.
b'_{1_j}	Threshold value for the j^{th} neuron in the hidden layer.