MIMO U-model Based Control: Real-Time Tracking Control and Feedback Analysis Via Small gain Theorem

ALI, SYED SAAD AZHAR
Air University
Electrical Engineering Department
E-9, Islamabad
PAKISTAN
saadali@mail.au.edu.pk

FOUAD M. AL-SUNNI
King Fahd University of Petroleum & Minerals
Systems Engineering Department
31261, Dhahran
SAUDI ARABIA
alsunni@kfupm.edu.sa

MUHAMMAD SHAFIQ
Ghulam Ishaq Khan Institute
Department of Electronic Engineering
Topi, Swabi
PAKISTAN
mshafiq@giki.edu.pk

JAMIL M. BAKHASHWAIN
King Fahd University of Petroleum & Minerals
Electrical Engineering Department
31261, Dhahran
SAUDI ARABIA
jamilb@kfupm.edu.sa

Abstract: In this paper, MIMO U-model based IMC is used for the tracking control of multivariable nonlinear systems. The algorithm is implemented in real-time on a 2DoF robot arm. The stability and convergence issues for the control-oriented U-model are also discussed. In order to guarantee stability and faster convergence speeds, bounds are suggested for the learning rate of adaptation algorithm that estimate the parameters of U-model. The adaptation algorithm is first associated with a feedback structure and then its stability is investigated using l_2 stability and small gain theorem. The paper also discusses about the robustness of adaptation algorithm in the presence of noise and suggests optimal choices for faster convergence speeds.

Key–Words: U-Model, LMS, Convergence Speed, Learning Rate, Small Gain Theorem.

1 Introduction

U-model is a recently proposed [1] modelling framework for adaptive control and tracking of nonlinear systems. U-model is a control oriented model which comes up with an explicit controller design methodology on approximate inverse concept. U-model has a more general appeal as compared to the polynomial NARMAX model [2] and the Hammerstein model [3], [4]. U-model is a polynomial in the control input $u(t-1)$ and the parameters of this polynomial are function of $u(t-2), \ldots u(t-m)$ and $y(t-1), \ldots u(t-n)$, where $u(t)$ and $y(t)$ represent the plant inputs and outputs, respectively. Inverse finding of polynomials is straight forward in numerical techniques such as Newton-Raphson method, while finding the inverses of other nonlinear models for dynamic system leads to theoretical and implementation difficulties in this regards [20]. It has been used in several nonlinear adaptive control scheme for both SISO and MIMO systems such as pole placement control [1], learning feedforward control [6, 7], adaptive inverse control [8], etc.

Practically all control schemes require the model to be stable and converging to the actual plant. U-model methodology engage an online recursive parameter estimation loop to model the unknown nonlinear plant in different control schemes. An appropriate adaptation gain called learning rate is often needed for the recursive procedure. The learning rate should be within an optimum range. It should neither be too large which would drive the algorithm unstable, nor too small, that it slows down the training. In general practice, a suitable learning rate is selected after few experimental runs.

Generally a small learning rate is chosen to avoid instability in learning, that evidently slows down the learning procedure. Especially, for MIMO systems with many parameters to estimate and a large data, a small learning rate may require substantial amount of time and machine power.

Therefore, it should be analyzed to find an optimal learning rate to speed up the convergence and yet keeping the algorithm stable. In the robustness analysis of adaptive schemes [9] and [10], the authors have addressed the methods of selecting the learning rate, that guarantees robustness under noisy conditions and faster convergence [9] and [10].

The formulation in [9] and [10] accentuates a feedback structure for many of the adaptive algorithms and it depends on tools from system theory, signal processing and control such as feedback analysis, state-space description, small gain theorem, H^{∞} design and lossless systems [9]. The authors have stated that the feedback configuration can be provoked via energy arguments and is shown to consist of two major blocks: a time-variant $lossless$ ($i.e.$, energy preserving) feedforward path and a time-variant feedback path [9, 10].

We make use of the feedback structure to analyze robustness of U-model and find optimal choices for learning rate. We will associate the learning algorithm with the feedback structure of [9] and [10]. As an argument, choices for learning rate that guarantee robust performance, stability and yet faster convergence speeds for the adaptation of U-model parameters.

This paper is organized in eight sections. The structure of U-model, MIMO U-model and the the Newton Raphson based controller are presented in section 2. The preliminaries for the analysis are presented in section 3. Robustness issues are discussed in section 54while the optimal learning rate using the feedback structure is presented in section 5. The real-time and simulation results are presented in section 6 and the paper is concluded in section 7.

2 The U-model Structure

The SISO U-model used for internal model control of a SISO plant in [11] and [12], based on the basic U-model developed by Zhu $et.al$ [1], models a plant of NARMAX representation given by,

$$y(t) = f\big(y(t-1),\ldots,y(t-n),u(t-1),\ldots, u(t-n),e(t-1),\ldots,e(t-n)\big), \quad (1)$$

The U-model is obtained by expanding the non-linear function of the above equation as a polynomial with respect to $u(t-1)$ as follows:

$$y_m(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1) + e(t), \quad (2)$$

where M is the degree of model input $u(t-1)$, α_j is a function of past inputs and outputs $u(t-2),\ \ldots,\ u(t-n),\ y(t-1),\ \ldots,\ y(t-n)$ and errors $e(t),\ \ldots,\ e(t-n)$. The sampled data representation of many non-linear continuous time systems may also be represented by the above form.

In order to establish the controller $i.e.$ the inverse of the model, the model output is set equal to the controller input as,

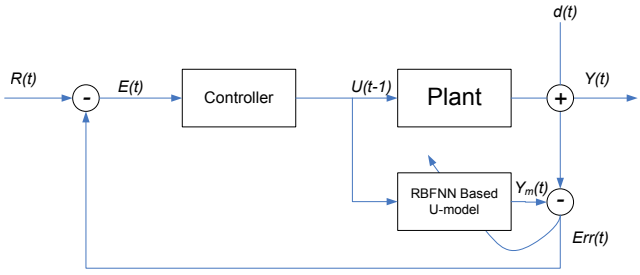

Figure 1: U-model based IMC scheme

$$y(t) = E(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1) + e(t). \quad (3)$$

Now the design of the control law transforms to a root solving problem that finds a control input $u(t-1)$, such that the controller-plant cascade produces a unity transfer function.

The control law presented in [12] using the Newton-Raphson method is given by,

$$u_{i+1}(t-1) = u_i(t-1) - \frac{\sum_{j=0}^{M} \hat{\alpha_j}(t) u_i^j(t-1) - E(t)}{d\sum_{j=0}^{M} \hat{\alpha_j}(t) u_i^j(t-1)/du_i(t-1)}$$

2.1 The MIMO U-Model and The Newton-Raphson Based Controller

The proposed U-model based IMC for multivariable nonlinear system is shown in Fig. 1. The model structure is given by,

$$Y_m(t) = A_0 + A_1 \overset{1}{U}(t-1) + A_2 \overset{2}{U}(t-1) + \ldots, \quad (4)$$

or

$$Y_m(t) = \sum_{j=0}^{M} A_j \overset{j}{U}(t-1) = F\big(U(t-1)\big). \quad (5)$$

The model output $Y_m(t)$ is a function of the current control signal $U(t-1)$, where $U(t-1)$ and $Y_m(t)$ are the input and output vectors. $\overset{j}{U}$ is the vector with j^{th} power of the control inputs $u_i(t-1)$ as,

$$\overset{j}{U}(t-1) = [u_1^j(t-1)\ u_2^j(t-1)\ \ldots\ u_p^j(t-1)]^T. \quad (6)$$

A_j are matrices instead of simple scalars. The problem in the proposed adaptive control structure is solved by first estimating the model in terms of the parameters A_j to obtain the MIMO U-model, and then

establishing the controller $i.e.$ the inverse using the Newton-Raphson.

To obtain a controller that acts as an inverse of the plant, it is required that the input to the controller $E(t)$ to be set equal to the model output $Y_m(t)$ [13]. In this way, the output of the controller $U(t-1)$, which when fed to the plant and the plant model, generates $Y(t)$ and $Y_m(t)$. Therefore, setting,

$$E(t) = Y_m(t) \tag{7}$$

Eq. 7 is system of multivariable nonlinear equations. This system of equations can be solved by any recursive nonlinear equations solver, such as the Newton-Raphson method [14]. Starting from an initial approximate solution, for instance $U_k(t-1)$, a better solution $U_{k+1}(t-1)$ is sought with the correction vector $H = [h_1 \dots h_n]$ such that,

$$U_{k+1}(t-1) = U_k(t-1) + H, \tag{8}$$

and

$$F\big(U_{k+1}(t-1)\big) = F\big(U_k(t-1) + H\big) = E(t) \tag{9}$$

is satisfied.

Now having the Taylor series expansion of $F\big(U_k(t-1) + H\big)$ with only the linear terms,

$$F\big(U_k(t-1)+H\big) \approx F\big(U_k(t-1)\big)+F'\big(U_k(t-1)\big)H. \tag{10}$$

The term $F'\big(U_k(t-1)\big)$ is the $p \times p$ $Jacobian$ matrix with elements $\partial f_i/\partial u_{k_j}(t-1)$.

Using Eq. 9 in Eq. 10, the value of the correction vector H can be obtained as,

$$H = F'\big(U_k(t-1)\big)^{-1}\big(E(t) - F\big(U_k(t-1)\big)\big) \tag{11}$$

Hence, the Newton-Raphson solution for the controller will be,

$$U_{k+1}(t-1) = \tag{12}$$
$$U_k(t-1) + F'\big(U_k(t-1)\big)^{-1}\big(E(t) - F\big(U_k(t-1)\big)\big)$$

Remarks: The Newton-Raphson solution is conditioned with the existence of the inverse of the $Jacobian$ in Eq. 12. It is possible during the update process to have a singular $Jacobian$ matrix. This situation can be avoided using one of the following techniques:

1. Employing Pseudoinverse,
2. or using the inverse of $Jacobian$ matrix from the previous instant,
3. or adding a small number to the $Jacobian$ matrix to avoid singularity.

3 Analysis Preliminaries

U-model is shown to be capable of modelling any dynamic system that can be represented as the NARMAX representation [1],[11]. The NARMAX representation is given by [15],

$$\begin{aligned} y(t) \;=\; & f\big(y(t-1), \dots, y(t-n), u(t-1), \dots, \\ & u(t-n), v(t-1), \dots, v(t-n)\big), \end{aligned} \tag{13}$$

where $v(t)$ represents the error due to measurement noise, model mismatch, uncertain dynamics and plant variation. The model of the plant is obtained by expanding the non-linear function of the above equation as a polynomial with respect to $u(t-1)$ as already expressed in Eq. 15

The polynomial in Eq. 15 is defined as the SISO U-model, where M is the degree of model input $u(t-1)$. The parameters $\alpha_j(t)$ are functions of past inputs given by [1],

$$\alpha_j(t) = f_j\big(y(t-1), \dots, y(t-n), u(t-2), \dots, u(t-n)\big) \tag{14}$$

For convenience in future analysis we express the U-model in vector notation as,

$$y(t) = A(t)U(t-1) + v(t), \tag{15}$$

where $A(t) = [\alpha_0(t)\ \alpha_1(t)\ \dots\ \alpha_M(t)]$ and $U(t-1) = [\ 1\ \ u(t-1)\ \ u^2(t-1)\ \dots\ \ u^M(t-1)]^T$.

3.1 Adaptation Algorithm and Error Quantities

Now consider the plant having a set of input vectors $\{u(t)\}$ with the corresponding desired set of output vectors $\{y(t)\}$ and assuming the plant can be modelled as a U-model of form given by Eq. 15. The noisy perturbations $v(t)$ can be contributed from noisy measurements or model uncertainties. This converts the problem of identifying the plant into finding the estimates of the U-model parameters $\alpha_j(t)$, for $0 \leq\ j\ \leq M$, such that

$$y_m(t) = \hat{A}(t)U(t-1), \tag{16}$$

where the vector $\hat{A}(t)$ is an estimate of $A(t)$ at time instant t. Starting with an initial guess $\hat{A}(0)$, the parameters are updated recursively based on the least mean square (LMS) principle as [17],

$$\hat{A}(t+1) = \hat{A}(t) + \mu(t)e(t)U^T(t-1) \tag{17}$$

where $\mu(t)$ is the learning and the error $e(t)$ is defined as,

$$\begin{aligned} e(t) &= y(t) - y_m(t) + v(t) \\ e(t) &= A(t)U(t-1) - \hat{A}(t)U(t-1) + v(t) \end{aligned} \tag{18}$$

Defining $a\ priori$ and $a\ posteriori$ error quantities as

$$e_a(t) = \tilde{A}(t)U(t-1) \quad (19)$$
$$e_p(t) = \tilde{A}(t+1)U(t-1) \quad (20)$$

where $\tilde{A}(t)$ is the parameter error vector symbolizing the difference between the actual parameter and its estimate as $\tilde{A}(t) = A(t) - \hat{A}(t)$. Therefore,

$$\begin{aligned} e_a(t) &= \big(A(t) - \hat{A}(t)\big)U(t-1), \\ &= A(t)U(t-1) - \hat{A}(t)U(t-1) \\ &= A(t)U(t-1) - y_m(t). \end{aligned}$$

and the parameter error update equation satisfies the following recursion,

$$\tilde{A}(t+1) = \tilde{A}(t) + \mu(t)e(t)U^T(t-1) \quad (21)$$

4 Robustness

It is imperative to bring up that a robust algorithm has consistent estimation errors independent of the disturbance type and may lead to minor estimation errors in the presence of minor disturbances [9]. Usually, this is not true for a particular adaptive algorithm. In an adaptive algorithm, even for small disturbances, the estimation errors can be large [16].

The robustness of U-model adaptation is analyzed deterministically without any information about the signal or noise. This approach is similar to [9] and [10] and is very useful for conditions where no prior information about the signal is known. The learning rate obtained as a result of this analysis would ensure an appropriate robustness level which is independent of the noise characteristics. In general, this implies that the disturbance energy will be upper bounded by the level of estimation energy level, or can be specifically stated as [9],

$$\frac{\text{estimation error energy}}{\text{disturbance energy}} \leq 1 \quad (22)$$

In the following section, the robustness methodology will be adopted to select a learning rate that ensures robustness behavior in the presence of noisy perturbations.

4.1 Optimal Learning Rate for Robustness

In this continuation, we will develop a contractive mapping from the t^{th} instant to $t + 1^{th}$ instant of the recursion.

Definition: *A linear map that transforms x to y, as $y = T[x]$, is said to be contractive mapping, if $\forall\ x$ we have $\|T[x]\|^2 \leq \|x\|^2$* [18].

This depicts that the output energy is always lesser than the input energy [9, 10]. The contractive mapping will relate the energies in such a way that the ratio in Eq. 22 is satisfied. More specifically, the Euclidean norm of the $\tilde{A}(t+1)$ and e_a at the $t + 1^{th}$ instant is compared with the Euclidean norms of $\tilde{A}(t+1)$ and $\tilde{v}(t)$ [9, 10].

$$\tilde{v}(t) = e(t) - e_a(t) \quad (23)$$

Now consider the parameter error recursion given by Eq. 21,

$$\tilde{A}(t+1) = \tilde{A}(t) - \mu(t)e(t)U^T(t-1)(t).$$

The squared norm (the energies), of the parameter error recursion equation can be computed as follows,

$$\begin{aligned} \|\tilde{A}(t+1)\|^2 &= \|\tilde{A}(t)\|^2 - 2\mu(t)e(t)\tilde{A}(t)U(t-1) \\ &\quad +\mu(t)^2e^2(t)\|U(t-1)\|^2, \\ &= \|\tilde{A}(t)\|^2 - 2\mu(t)\tilde{A}(t)U(t-1)(e_a(t)+\tilde{v}(t)) + \\ &\quad \mu(t)^2\|U(t-1)\|^2(e_a(t)+\tilde{v}(t))^2, \\ &= \|\tilde{A}(t)\|^2 - 2\mu(t)(e_a^2(t)+e_a(t)\tilde{v}(t)) + \\ &\quad \mu(t)^2\|U(t-1)\|^2(e_a^2(t)+2e_a(t)\tilde{v}(t)+\tilde{v}^2(t)), \\ &= \|\tilde{A}(t)\|^2 - 2\mu(t)e_a^2(t) - 2\mu(t)e_a(t)\tilde{v}(t) + \\ &\quad \mu(t)^2\|U(t-1)\|^2e_a^2(t) \\ &\quad + 2\mu(t)^2\|U(t-1)\|^2e_a(t)\tilde{v}(t) \\ &\quad + \mu(t)^2\|U(t-1)\|^2\tilde{v}^2(t), \end{aligned}$$

Rearranging terms we get,

$$\begin{aligned} \|\tilde{A}(t+1)\|^2 + 2\mu(t)e_a^2(t) - \mu(t)^2\|U(t-1)\|^2e_a^2(t) = \\ \|\tilde{A}(t)\|^2 - 2\mu(t)e_a(t)\tilde{v}(t) + \\ 2\mu(t)^2\|U(t-1)\|^2e_a(t)\tilde{v}(t) + \\ \mu(t)^2\|U(t-1)\|^2\tilde{v}^2(t). \end{aligned} \quad (24)$$

Introducing a parameter $\eta(t)$ as

$$\eta(t) = \frac{1}{\|U(t-1)\|^2} \quad (25)$$

$$\begin{aligned} \|\tilde{A}(t+1)\|^2 + 2\mu(t)e_a^2(t) - \frac{\mu(t)^2}{\eta(t)}e_a^2(t) = \\ \|\tilde{A}(t)\|^2 - 2\mu(t)e_a(t)\tilde{v}(t) + 2\frac{\mu(t)^2}{\eta(t)}e_a(t)\tilde{v}(t) \\ + \frac{\mu(t)^2}{\eta(t)}\tilde{v}^2(t). \end{aligned} \quad (26)$$

If we set $\mu(t) = \eta(t)$, we come up to the following equality, where the energy bounds are always

satisfied as estimation energy = disturbance energy.

$$\begin{aligned}\|\tilde{A}(t+1)\|^2 + 2\mu(t)e_a^2(t) - \mu(t)e_a^2(t) = \\ \|\tilde{A}(t)\|^2 - 2\mu(t)e_a(t)\tilde{v}(t) + \\ 2\mu(t)e_a(t)\tilde{v}(t) + \mu(t)\tilde{v}^2(t), \\ \|\tilde{A}(t)\|^2 + \mu(t)e_a^2(t) = \|\tilde{A}(t)\|^2 + \mu(t)\tilde{v}^2(t). \quad (27)\end{aligned}$$

Therefore, we can conclude to the results for the energy bounds depending upon the learning rate [9, 10].

$$\frac{\|\tilde{A}(t+1)\|^2 + \mu(t)e_a^2(t)}{\|\tilde{A}(t)\|^2 + \mu(t)\tilde{v}^2(t)} \begin{Bmatrix} \leq 1 \; if \; 0 < \mu(t) < \eta(t) \\ = 1 \; if \; \mu(t) = \eta(t) \\ \geq 1 \; if \; \mu(t) > \eta(t) \end{Bmatrix} \quad (28)$$

Remarks

1. The first two inequalities in the statement of Eq. 28 ascertain that if the learning rate is selected such that $\mu(t) \leq \eta(t)$, then a contractive mapping is achieved from the signals $\{\tilde{A}(t), \sqrt{\eta(t)}e_p(t)\}$ to the signals $\{\tilde{A}(t+1), \sqrt{\eta(t)}e_a(t)\}$ [9, 10]. Therefore, a local energy bound is deduced that highlights robustness of the update recursion.

2. The energy bound is independent of the noise signal $\tilde{v}(t)$ and the parameter error vector $\tilde{A}(t)$. The sum of energies $\|\tilde{A}(t+1)\|^2 + \mu(t)e_a^2(t)$ will always be lesser than or equal to the sum of energies $\|\tilde{A}(t)\|^2 + \mu(t)\tilde{v}^2(t)$ and the algorithm will show robust behavior in the presence of noisy perturbations and load variations [9, 10].

3. Moreover, the adaptation algorithm looses its robustness properties if the the learning rate obeys the third condition in the Eq. 28.

5 Feedback Structure

The bounds of the statement given by Eq. 28 can be illustrated in an alternative form that establishes the feedback structure. Initially, the recursive parameter update equation has to be written as a function of $a\ priori\ error$ and $a\ posteriori\ error$. The $a\ posteriori\ error$ s defined in Eq. 20 as,

$$\begin{aligned} e_p(t) &= \tilde{A}(t+1)U(t-1) \\ &= [\tilde{A}(t) - \mu(t)U^T(t-1)e(t)]U(t-1) \\ &= e_a(t) - \mu(t)\|U(t-1)\|^2 e(t) \\ &= e_a(t) - \frac{\mu(t)}{\eta(t)}e(t) \quad (29) \end{aligned}$$

$$\begin{aligned} \eta(t)e_p(t) &= \eta(t)e_a(t) - \mu(t)e(t) \\ \eta(t)\big(e_a(t) - e_p(t)\big) &= \mu(t)e(t). \quad (30) \end{aligned}$$

Hence, the recursive parameter update Eq. 17 can be written as

$$\hat{A}(t+1) = \hat{A}(t) + \eta(t)\big(e_a(t) - e_p(t)\big)U^T(t-1).$$

Similarly, the parameter error recursion Eq. 21 can be reformulated as,

$$\tilde{A}(t) = \tilde{A}(t) - \eta(t)U^T(t-1)\big(e_a(t) - e_p(t)\big) \quad (31)$$

The squared norm of Eq. 31 leads to the same statement as 28, except that the disturbance error $\tilde{v}(t)$ is replaced by the negative of $a\ posteriori\ error\ -e_p(t)$ and the learning rate is set to $\eta(t)$ as follows,

$$\begin{aligned}\|\tilde{A}(t+1)\|^2 = \\ \|\tilde{A}(t)\|^2 - 2\eta(t)\tilde{A}(t)U(t-1)\big(e_a(t) - e_p(t)\big) \\ +\eta(t)^2\|U(t-1)\|^2\big(e_a(t) - e_p(t)\big)^2 \\ = \|\tilde{A}(t)\|^2 - 2\eta(t)e_a(t)\big(e_a(t) - e_p(t)\big) + \\ \eta(t)^2\frac{1}{\eta(t)}e_a^2(t) - 2\eta(t)^2\frac{1}{\eta(t)}e_a(t)e_p(t) + \\ \eta(t)^2\frac{1}{\eta(t)}e_p^2(t) \\ = \|\tilde{A}(t)\|^2 - 2\eta(t)e_a^2(t) + 2\eta(t)e_a(t)e_p(t) + \\ \eta(t)e_a^2(t) - 2\eta(t)e_a(t)e_p(t) + \eta(t)e_p^2(t)\end{aligned}$$

Therefore,

$$\begin{aligned}\|\tilde{A}(t+1)\|^2 = \|\tilde{A}(t)\|^2 - 2\eta(t)e_a^2(t) + \\ \eta(t)e_a^2(t) + \eta(t)e_p^2(t),\end{aligned}$$

or

$$\|\tilde{A}(t+1)\|^2 + \eta(t)e_a^2(t) = \|\tilde{A}(t)\|^2 + \eta(t)e_p^2(t),$$

expressing in a ratio form as,

$$\frac{\|\tilde{A}(t+1)\|^2 + \eta(t)e_a^2(t)}{\|\tilde{A}(t)\|^2 + \eta(t)e_p^2(t)} = 1. \quad (32)$$

Hence, the energy ratio given in Eq. 32 is true for all possible learning rates. This implies that the mapping $\overline{T}_i$ from the signals $\{\tilde{A}(t), \sqrt{\eta(t)}e_p(t)\}$ to the signals $\{\tilde{A}(t+1), \sqrt{\eta(t)}e_a(t)\}$ is lossless [9] and [10].

Now if we apply the mean-value theorem to the U-model output $y(t)$, we can write

$$A(t)U(t-1) - \hat{A}(t)U(t-1) = y'(\tau)e_a(t) \quad (33)$$

for some point τ along the segment connecting $A(t)U(t-1)$ and $\hat{A}(t)U(t-1)$ during estimation procedure. Therefore, combining Eq. 18 and Eq. 29,

$$
\begin{aligned}
e_p(t) &= e_a(t) - \frac{\mu(t)}{\eta(t)}e(t) \\
e_p(t) &= e_a(t) - \\
&\quad \frac{\mu(t)}{\eta(t)}\big(A(t)U(t-1) - \tilde{A}(t)U(t-1) + v(t)\big) \\
e_p(t) &= e_a(t) - \frac{\mu(t)}{\eta(t)}\big(y'(\tau)e_a(t) + v(t)\big) \\
e_p(t) &= [1 - \frac{\mu(t)}{\eta(t)}y'(\tau)]e_a(t) - \frac{\mu(t)}{\eta(t)}v(t) \\
-\sqrt{\eta(t)}e_p(t) &= \frac{\mu(t)}{\sqrt{\eta(t)}}v(t) - \\
&\quad [1 - \frac{\mu(t)}{\eta(t)}y'(\tau)]\sqrt{\eta(t)}e_a(t) \quad (34)
\end{aligned}
$$

In [9] and [10], the authors have incorporated a feedback structure to relate the overall mapping from the disturbance errors to the estimation errors. The relation in Eq.34 can also be shown as a feedback structure similar to the one presented in [9] and [10]. The feedback structure depicting the overall mapping from the $\sqrt{\eta(t)}v(t)$ to $\sqrt{\eta(t)}e_a(t)$ is shown in Fig. 2. In order to analyze the stability of the feedback structure the authors have utilized the small gain theorem [9, 10]. We will also make use of the small gain theorem theorem to study the stability of the feedback structure shown in Fig. 2 for the case of U-model adaptation. Conditions on the learning rate $\mu(t)$ will be derived that guarantee a robust, stable and yet faster training algorithm.

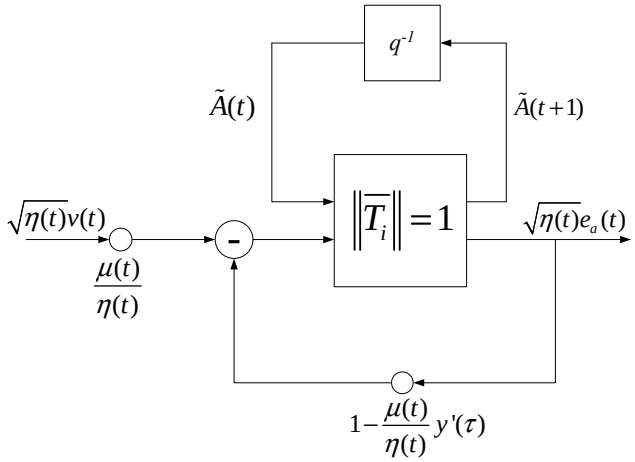

Figure 2: A lossless mapping in feedback structure for the learning algorithm of U-model

The conditions will be developed in such a way that the feedback configuration shown in Fig. 2 remains l_2 stable. This implies that a finite-energy input noise sequence including the noiseless case as well) $\{\sqrt{\eta(t)}v(t)\}$ is mapped into a finite-energy $a\ priori$ error sequence $\{\sqrt{\eta(t)}e_a(t)\}$ [9, 10].

5.1 Optimal Learning Rate for U-model via Small Gain Theorem

Now to obtain the optimal learning rate using small gain theorem, define the gain for the feedback loop in Fig. 2 as $\Delta(N)$,

$$\Delta(N) \quad = \quad \max_{0\le t\le N} \ |1 - \frac{\mu(t)}{\eta(t)}y'(\tau)| \ . \qquad (35)$$

According to the definition in Eq. 35, $\Delta(N)$ is the maximum absolute gain of the feedback loop over the interval $0 \le t \le N$.

According to small gain theorem, the condition for l_2 stability of the feedback configuration shown in Fig. 2 is that the product of norms of the feedforward and feedback maps be strictly bounded by one [9, 10].

Since, we have already assumed a lossless case, this implies that the norm of the feedforward map is equal to one. The norm of the feedback map is defined as $\Delta(N)$. Therefore, to keep the product of the norms to be less than one, it is required that $\Delta(N) < 1$.

Now the learning rate should be selected that results in $\Delta(N)$ to be less than one and guarantees the bound

$$0 < \mu(t)y'(\tau) < 2\eta(t) = \frac{2}{\|u(t)\|^2} \qquad (36)$$

In [19], the authors have presented a number of choices for learning rate. They based the selection of learning rate on the availability of the derivative function $y'(\tau)$. For the case of U-model it is straight forward to obtain the estimate of the derivative function as,

$$
\begin{aligned}
y(\tau) &= \sum_{j=0}^{M} \alpha_j(\tau)u^j(t-1), \\
y'(\tau) &= \sum_{j=1}^{M} j\alpha_j(\tau)u^{j-1}(t-1). \qquad (37)
\end{aligned}
$$

Hence, the derivative in Eq. 37 can be used to find the optimal learning rate to speed up the convergence as,

$$\mu(t) \quad < \quad 2\eta(t)y'(\tau),$$

$$\boxed{\mu(t) < \eta(t) \sum_{j=1}^{M} j\alpha_j(\tau) u^{j-1}(t-1).} \quad (38)$$

Remarks

The inequality given by Eq. 38 not only ensures the stability of the update recursion of the U-model parameters but also guarantees faster convergence speeds.

6 Results

6.1 Real-Time Implementation on a 2-Degree of Freedom Robot Manipulator

To test and verify the behavior and robustness of the proposed algorithm, we have developed a 2-degree of freedom robot manipulator shown in Fig. 3.

Figure 3: The Real-Time setup for the 2 link Robot

The first link (named $primary$) is 30cm and the second link (named $secondary$) is 19cm long. For a varying load, the link is connected by elastic strings on both sides, such that the tension in the string is variable according to the angular position of the link. Tension in the string increases with increasing rotation angle. The geometry of the 2 link robot is shown in Fig. 4

The feedback signals $i.e.$ the angles of the links are measured by two 0-50KΩ potentiometers. Due to the physical limitations, the primary link is constrained to have a maximum rotation of $\pm 60^o$ from the central position. However, the secondary link can manoeuver the whole $\pm\ 180^o$ rotation.

The objective of the 2 link robot manipulator is; given any coordinates in the workspace, the $end-effector$ will be driven to those desired coordinates in the robot workspace within finite time, practically in shortest time. This is achieved by rotating the robot links to corresponding angles. The problem of finding the angles of the robot links given any coordinates in the workspace is called inverse kinematics. The control algorithm treats the values of the angles in radians as the setpoints and attempts to track the angles of the links to those setpoints. The actual position of the end effector can be obtained using the forward kinematics.

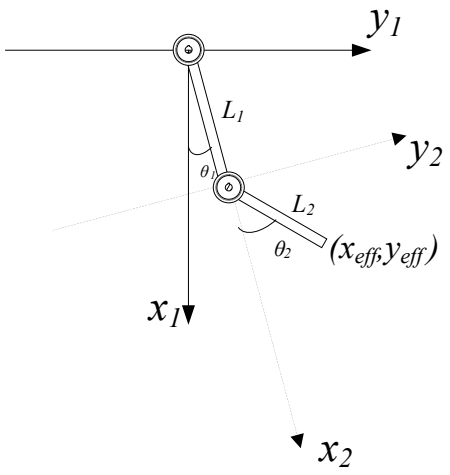

Figure 4: The 2 link Robot Geometry

For the 2 link robot having lengths L_1 and L_2, the inverse kinematics problem is defined as, given the desired x_d and y_d coordinate of the end effector, find the angles for the primary and secondary links. This can be achieved by the following equations.

Defining B as the distance of the end effector from the origin of the base frame.

$$B = \sqrt{x^2 + y^2} \quad (39)$$

The angles θ_1 and θ_2 are calculated by,

$$\theta_1 = Atan2(y_d/x_d) + cos^{-1}[(L_1^2 - L_2^2 + B^2)/2L_1B], \quad (40)$$

$$\theta_2 = cos^{-1}[(L_1^2 + L_2^2 - B^2)/2L_1L_2], \quad (41)$$

The function $Atan2(Y/X)$ finds the proper quadrant for the angle (There could be more than one solution to even a single link as the inverse of cosine generates $\pm$ angles, so it is necessary to find the correct quadrant).

The position of the end effector is calculated using the forward kinematics of the 2 link robot. Given the angles θ_1 and θ_2, the end effector coordinates x_{eff} and y_{eff} are,

$$x_{eff} = L1\cos(\theta_1) + L2\cos(\theta_1 + \theta_2), \quad (42)$$

$$y_{eff} = L1\sin(\theta_1) + L2\sin(\theta_1 + \theta_2). \quad (43)$$

The real-time code is built in SIMULINK on Intel Pentium III 933MHz Computer with 256MB RAM. The interfacing is done using the Advantech PCI-1711 I/O card.

The reference signal is set to be a uniformly distributed random signal with a step time of 4 seconds. Using a tuned standard PID controller at no load; the tracking behavior is shown in Fig. 5. The figure shows an acceptable steady-state tracking even though there are high overshoots at the transition and occasional mismatch in the tracking. The proposed U-model

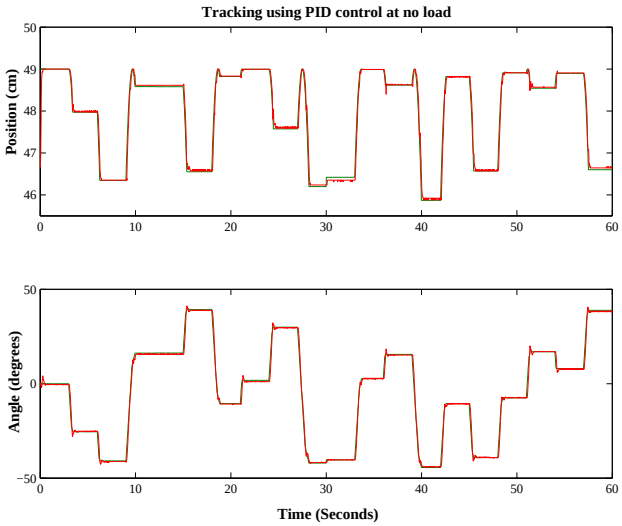

Figure 5: Tracking using a PID controller at no load

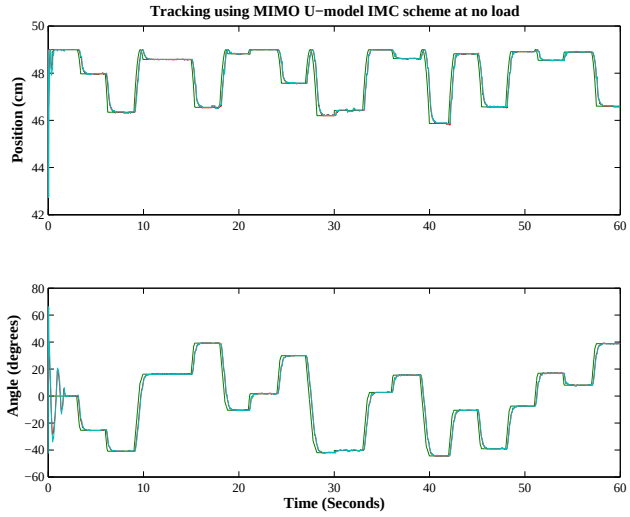

Figure 6: Tracking using the proposed U-model scheme at no load

based adaptive tracking scheme is applied to the 2 link robot using a $3rd$ order U-model and a 2 input 2 output RBFNN with 2 neurons for the A_o. The width of the Gaussian basis functions is kept as 1 to cover a large input range. The weights of the RBFNN and the matrix parameters A_j are updated using the LMS principle with a learning rate of 0.05. The tracking is shown in Fig. 6. Fine tracking performance can be observed with no overshoot and mistracking as compared to the standard PID controller.

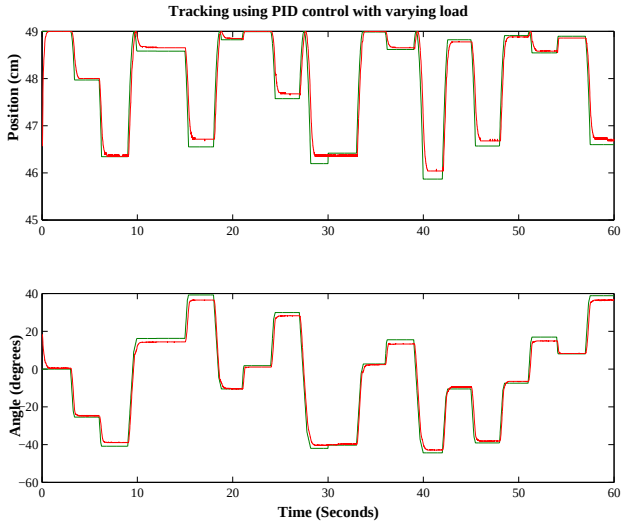

Figure 7: Tracking with variable load using a PID controller tuned for no load

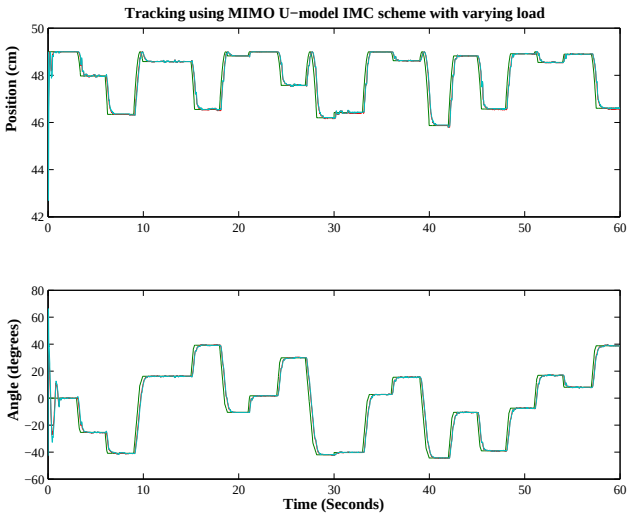

Figure 8: Tracking with variable load using the proposed U-model scheme

Using the same setup, with a varying load, the performance of the standard PID controller tuned at no load is shown in Fig. 7 and it is obvious that the PID controller tuned at no load was not able to track the reference signal. When the proposed scheme was applied to the varying load setup, very similar tracking results were observed as depicted in Fig. 8. This shows the robustness of the adaptive scheme that is able to perform even with load variation. A possible way to avoid the overshoots is to employ the three step input shaping technique proposed in [20]

6.2 Simulation for Adaptive Learning Rate

A hammerstein system is considered to verify the performance of the U-model scheme with a constant and with an adaptive learning rate obtained in Eq. 38. The hammerstein system considered is a 2-input 2-output heat exchanger. The two similar static nonlinearities in heat exchanger are defined as [21].

$$\begin{aligned} x(t) &= -31.549u(t) + 41.732u^2(t) \\ &\quad + 24.201u^3(t) + 68.634u^4(t). \end{aligned}$$

The linear dynamic block is considered as,

$$\begin{aligned} y_1(t) &= -0.82y_1(t-1) - 0.75y_1(t-2) \\ &\quad + 0.81x_1(t) + 0.53x_2(t), \\ y_2(t) &= -0.61y_2(t-1) - 0.53y_2(t-2) \\ &\quad + 0.62x_1(t) + 0.22x_2(t). \end{aligned}$$

The reference signal is set to be a random piece-wise continuous signal with a step time of 3 seconds. A third order U-model with $M = 3$ is used. The learning rate is set to 0.05 after few trial runs. The output is made noisy with additive white gaussian noise of 20dB SNR. The tracking behavior is shown in Fig 9. It can be seen that a constant learning behavior is observed. However, The tracking behavior can be different for a different learning rate. Now, the same system

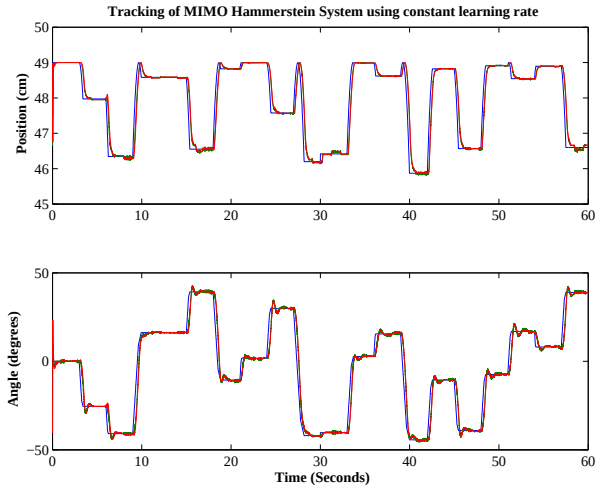

Figure 9: Tracking using a constant learning rate

is used and the learning is made adaptive using the criteria in Eq. 38. The learning rate $\mu(t)$ is kept 95% of the right hand side. The tracking behavior is shown in Fig. 10. It can be seen that although initially the performance was not acceptable, the adaptive learning rate eventually set the learning process in stable region and finer tracking behavior is followed after some time.

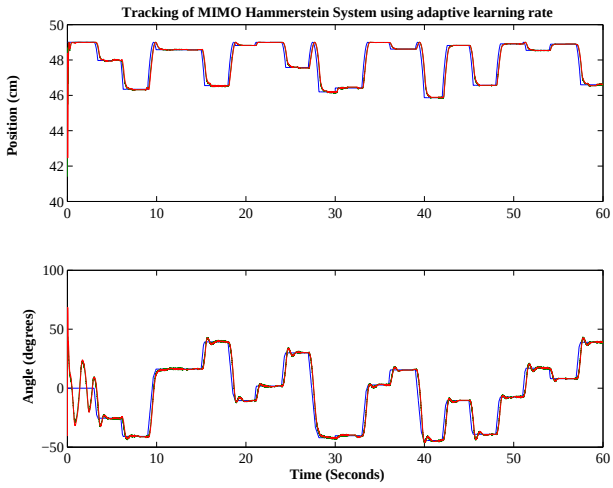

Figure 10: Tracking using an adaptive learning rate

7 Conclusions

In this paper a feedback analysis of the learning algorithm of U-model is presented. The U-model adaptation is related to a feedback structure of [9]. The stability of the adaptation is studied via small gain theorem. Choices for suitable learning rates are suggested that ensure robust behavior under noisy conditions model uncertainties. In order to speed up the convergence, bounds for the optimal learning rate are presented.

Acknowledgements: The research was supported by the King Fahd University of Petroleum & Minerals Project SABIC 2006-11 and Higher Education Commission Pakistan Grant 58-11/1/HEC (R&D)/08.

References:

[1] Zhu, Q. M. and Guo. L. Z, *"A pole placement controller for nonlinear dynamic plants,"* Journal of Systems and Control Engineering, Vol. 216 (part I), pp. 467 - 476 , 2002.

[2] Sales, K. R. and Billings, S. A., *"Self-tuning control of nonlinear ARMAX models,"* Intrenational Journal of Control, Vol. 51, no. 4, pp. 753-769, 1990.

[3] Hussain N. Al-Duwaish and Ali, S. Saad Azhar, *"Identification of hammerstein model using radial basis functions neural networks,"* Intrenational Conference on Artificial Neural Networks, Vol. 1, no. 1 pp. 951-956, August 2001.

[4] Syed Saad Azhar Ali, Hussain N. Al-Duwaish and M. Moinuddin, "Radial basis functions

neural networks based self tunning regulator," *WSEAS Transactions on Systems*, No. 9 Vol.3 pp. 2777-2781, Nov. 2004.

[5] Jaroslav Hlava and Bohmil Sulc, "Modelling and simulation of hybrid systems using a laboratory-scale plant," *WSEAS Transactions on Systems*, No. 9 Vol.3 pp. 2777-2781, Nov. 2004.

[6] Ali, S. Saad Azhar, Fouad M. AL-Sunni and Shafiq, M., "*U-model based adaptive tracking scheme for MIMO bilinear systems*," 1st IEEE International conference on Industrial Electronics and Applications, Singapore, May 2006.

[7] Ali, S. Saad Azhar, Fouad M. AL-Sunni, Shafiq, M. and Jamil M. Bakhashwain, "*Learning Feedforward control of MIMO nonlinear systems using U-model*," 9th IASTED conference on Control and Applications,Montreal, Quebec, May 2007.

[8] Shafiq, M. and Khan, T., "*Newton-Raphson based adaptive inverse control scheme for tracking of nonlinear dynamic plants* ," 1st International Symposium on Systems and Control in Aerospace and Astronautics, China, Feb 2006.

[9] Ali H. Sayed and M. Rupp, "A time-domain feedback analysis of filtered-error adaptive algorithms via the samll gain theorem," *Proceedings of SPIE,*, Vol. 2563, pp. 458-469, June 1996.

[10] M Rupp and Ali H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE Transactions on Signal Processing*, Vol. 44, pp. 1428-1439, July 1995.

[11] Shafiq, M. and Haseebuddin, M., "*Internal model control for nonlinear dynamic plants using U-model*," 12th Mediterranean Conference on Control and Automation, Turkey, June 2004.

[12] Shafiq, M. and Butt, N. R., "U-model based adaptive IMC for nonlinear dynamic plants," *10th IEEE International Conference on emerging technologies and factory automation* , Sep. 2005, Italy

[13] Ali, S. Saad Azhar, Fouad M. AL-Sunni, Shafiq M. and Bakhashwain Jamil M., "Learning feedforwad control of MIMO nonlinear systems using U-model," 9^{th} *IASTED International Conference on Control and Applications*, May 2007.

[14] Cheney, W. and Kincaid, D., "Numerical Mathematics and Computing, 5th Edition ," *Brooks/Cole Publishing Company*, 2004.

[15] Chen, S. and Billings, S. A., "Representations of nonlinear systems: the narmax model," *International Journal of Control*, Vol 4, pp. 10131032, 1988.

[16] Hasibi, B., Ali H. Sayed and Kailath, T., "H_∞ optimality of the LMS algorithm," *IEEE Transactions on Signal Processing*, Vol. 44, pp. 267-280, Feb. 1996.

[17] Haykin, S, "Neural Networks: A Comprehensive Foundation II," *Macmillan / IEEE press*, 1994, 1999.

[18] Khalil, H., "Nonlinear Systems," *MacMillan*, 1992.

[19] Rupp, M. and Ali H. Sayed, "Supervised learning of perceptron and output feedback dynamic networks: A feedback analysis via the small gain theorem," *IEEE Transactions on Neural Networks*, Vol. 8, pp. 612-622, May 1997.

[20] Sirri Sunay Gurleyuk, Ozgur Bahadir, Yunus Turkkan Meb and Hakan Usenti, "Performance analysis of three step positive input shaping," *10th WSEAS Intl. Conf. on Automatic Control Modelling and Simulation, Istanbul, Turkey*, May 2008.

[21] E. Eskinat, S. H. Johnson, and W. L. Luyben., "Use of hammerstein models in identification of nonlinear systems," *American Institute of Chemical Engineering Journal*, Vol.37 pp. 255-268, 1991.