# U-model Based Adaptive Internal Model Control of Unknown MIMO Nonlinear Systems: A Case study on 2-Link Robotic Arm ⋆

**Ali, S. Saad Azhar** * **Muhammad Shafiq** **
**Jamil M. Bakhashwain** *** **Fouad M. AL-Sunni** ****

* *Electrical Engineering Department, Air University, E-9 Islamabad, Pakistan, (email: saadali@mail.au.edu.pk).*
** *Department of Electronics Engineering, Ghulam Ishaq Khan Institute, Topi, Pakistan, (email: mshafiq@giki.edu.pk).*
*** *Electrical Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia, (email: jamilb@kfupm.edu.sa).*
**** *Systems Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia, (email: alsunni@kfupm.edu.sa).*

**Abstract:** In this paper, we propose a more generalized controller design methodology for a class of nonlinear plants. This design procedure is based on MIMO U-model structure. The U-model significantly simplifies the online synthesis of the control law. The proposed technique is applied for the internal model control of a 2-link robot manipulator. The performance of the proposed U-model based internal model controller is compared to standard PID controller under different conditions.

KEY WORDS

MIMO Nonlinear systems, IMC, U-model, RBFNN.

## 1. INTRODUCTION

The increasingly complex dynamical systems with significant uncertainty have limited the use of linear model based controllers. Consequently, a very large variety of control techniques have been developed based on a non-linear model of the system. Internal model control (IMC) is an advanced model based control arithmetic and has gained increasingly more attention since it has been proposed in [1] and [2]. IMC provides a simple yet effective framework for the analysis and synthesis of control system performance, especially its robustness properties. Additionally, it is easy to design and regulate on-line, and capable to eliminate unmeasured disturbance. Although IMC is used in linear processes successfully, it is difficult to obtain satisfactory control performance when IMC is directly introduced in nonlinear process. The nonlinear version of the IMC was presented in [3]. During recent years, researchers have adverted their eyes to nonlinear modelling technique, intelligent methods such as neural networks [4] and fuzzy logic [5] are adopted in nonlinear IMC design. An adaptive IMC scheme based on adaptive finite impulse response filters is presented in [6]. A neural network based multi-model IMC structure is presented in [7] for the adaptive tracking of plants with strong nonlinear characteristics. In most of these methods the plant model is a highly nonlinear complex system, that further complicates the design of the inverse *i.e.*, the synthesis of the control law using the model is very difficult.

In [8] and [9], U-model is used for the internal model control of SISO unknown nonlinear dynamic plants and in [10] for MIMO bilinear systems. The U-model proposed by [13] is a control oriented model that not only has the nonlinear approximation capability but also simplifies the synthesis of the control law. The U-model has a more general appeal as compared to the polynomial NARMAX model [14] and the Hammerstein model. The U-model exhibits a polynomial structure in the current control $U(t-1)$ and due to its polynomial structure, the control design is simply the solution of the polynomial equation, unlike other models which lead to complex non-linear algebraic equations. A simplified U-model has been used for MIMO bilinear systems in [10] and U-model based learning feedforward control for MIMO nonlinear systems is presented in [11].

We propose an IMC scheme for the unknown MIMO nonlinear plants using radial basis function neural networks (RBFNN) based MIMO U-model and Newton-Raphson based controller. The plant is identified online using the U-model and the inverse of the model is established using the U-model methodology.

This paper is organized as follows. The problem is stated in section 2. The U-model structure is briefed in section 3 along with the necessary background. Section 4 presents

⋆ This work is sponsored by King Fahd University of Petroleum & Minerals/SABIC under project SABIC 2006-11 and Higher Education Commission, Pakistan

the proposed MIMO U-model scheme and the Newton-Raphson based controller. The real-time experiment details and the results are presented in section 5.

### *1.1 Problem Statement*

In this research, the problem of tracking an input reference signal for MIMO nonlinear dynamic plants is considered. The NARMAX representation of the MIMO nonlinear plants is [12],

$$Y(t) = F\big(Y(t-1), \ldots, Y(t-n), U(t-1), \ldots, U(t-n), E(t-1), \ldots, E(t-n)\big), \quad (1)$$

where $Y(t)$ and $U(t)$ are the output and input signals of the plant respectively at a discrete time instant $t$, $n$ is the order of the plant, $E(t)$ represents the error due to measurement noise, model mismatch, uncertain dynamics and plant variation. $F(\boldsymbol{.})$ is a non-linear function of the inputs, outputs and errors. The objective is to synthesize the control input $U(t)$ such that $Y(t)$ tracks the desired input reference signal $R(t)$, while the plant parameters are unknown or time varying.

## 2. THE U-MODEL STRUCTURE

The SISO U-model used for internal model control of a SISO plant in [8] and [9], based on the basic U-model developed by Zhu *et.al* [13], models a plant of NARMAX representation given by,

$$y(t) = f\big(y(t-1), \ldots, y(t-n), u(t-1), \ldots, u(t-n), e(t-1), \ldots, e(t-n)\big), \quad (2)$$

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:

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

where $M$ is the degree of model input $u(t-1)$, $\alpha_j$ 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.

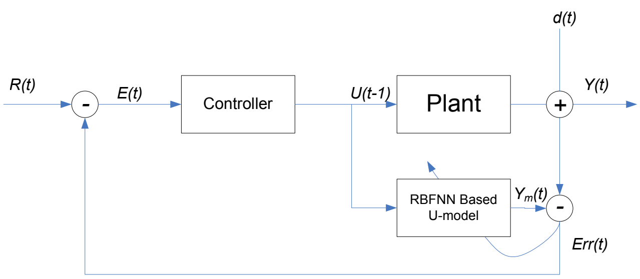


Fig. 1. U-model based IMC scheme

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

$$y(t) = E(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1) + e(t). \quad (4)$$

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 [9] using the Newton-Raphson method is given by,

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

## 3. 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,

$$Y_m(t) = A_0 + A_1 \overset{1}{U}(t-1) + A_2 \overset{2}{U}(t-1) + \ldots, \quad (5)$$

or

$$Y_m(t) = \sum_{j=0}^{M} A_j \overset{j}{U}(t-1) = F\big(U(t-1)\big). \quad (6)$$

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,

$$\overset{j}{U}(t-1) = [u_1^j(t-1) \; u_2^j(t-1) \; \ldots \; u_p^j(t-1)]^T. \quad (7)$$

$A_j$ 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 $Y_m(t)$ [11]. 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,

$$E(t) = Y_m(t) \quad (8)$$

Eq. 8 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 [15]. 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 $H = [h_1 \ldots h_n]$ such that,

$$U_{k+1}(t-1) = U_k(t-1) + H, \quad (9)$$

and

$$F\big(U_{k+1}(t-1)\big) = F\big(U_k(t-1) + H\big) = E(t) \quad (10)$$

is satisfied.

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

$$F\big(U_k(t-1)+H\big) \approx F\big(U_k(t-1)\big) + F'\big(U_k(t-1)\big)H. \quad (11)$$

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. 10 in Eq. 11, the value of the correction vector $H$ can be obtained as,

$$H = F'\big(U_k(t-1)\big)^{-1}\big(E(t) - F\big(U_k(t-1)\big)\big) \qquad (12)$$

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

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

$$U_{k+1}(t-1) = U_k(t-1) - F'\big(U_k(t-1)\big)^{-1}\Big(E(t) - \sum_{j=0}^{M} A_j U_j(t-1)\Big). \qquad (13)$$

***Remarks:*** The Newton-Raphson solution is conditioned with the existence of the inverse of the *Jacobian* in Eq. 13. 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.

The term $A_0$ is modelled using a MIMO RBFNN as,

$$A_0 = \mathbf{W}\mathbf{\Phi}. \qquad (14)$$

The reason for incorporating the RBFNN is to assist the nonlinear modelling. A particular class of nonlinearities *e.g.* backlash, saturation and deadzone etc can be modelled using neural networks. These nonlinearities may or may not be present in cascade with the actual nonlinear system, particularly in actuators.

### *3.1 Radial Basis Functions Neural Networks*

RBFNN is a type of feedforward neural network. They are used in a wide variety of contexts such as function approximation, pattern recognition and time series prediction. Networks of this type have the universal approximation property [16]. In these networks the learning involves only one layer with lesser computations. These features make RBFNN attractive in many practical problems. An $M$ input $P$ output RBFNN is shown in Fig. 2. The RBFNN consists of an input node $u(t)$, a hidden layer with $n_o$ neurons and an output node $y(t)$. For MIMO systems, the input and output are vectors as $U(t) = [u_1(t)\ldots u_M(t)]$ and $Y(t) = [y_1(t)\ldots y_P(t)]$. Each of the input node is connected to all the nodes or neurons in the hidden layer through unity weights (direct connection). While each of the hidden layer nodes is connected to the output node through some weights, *e.g.* the $i_{th}$ output node is connected with all the hidden layer nodes by $W_i = [w_{i1}, \ldots, w_{in_o}]$. Each neuron finds the distance, normally applying Euclidean norm, between the input and its center and passes the resulting scalar through a nonlinearity. So the output of the $i_{th}$ hidden neuron is given by $\phi_i(\|U(t) - c_i\|)$, where $c_i$ is the center of $i^{th}$ hidden layer node, $i = 1, 2, \ldots, n_o$, and $\phi_i(\cdot)$ is the nonlinear basis

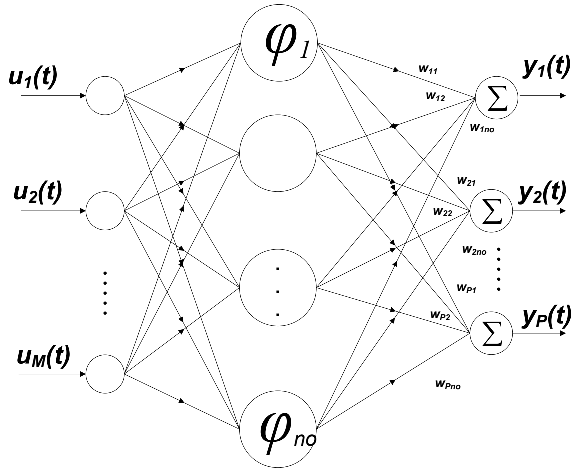


Fig. 2. A MIMO RBF neural network.

function. Normally this function is taken as a Gaussian function of width $\beta$. The output $y_j(t)$ is a weighted sum of the outputs of the hidden layer, given by

$$y_j(t) = W_j\Phi(t),$$
$$y_j(t) = \sum_{i=1}^{n_o} w_{ji}\phi_i(\|U(t) - c_i\|),$$

where the $j^{th}$ output has the effect of all the $M$ inputs and $w_{ij}$ is the weight connecting the $j^{th}$ neuron to the $i^{th}$ output. Defining $\mathbf{W} = [W_1 \ \ldots \ W_P]^T$, the MIMO RBFNN is expressed in matrix notation neatly as,

$$Y(t) = \mathbf{W}\Phi(t). \qquad (15)$$

The weights of the RBFNN and the rest of the parameters $A_j$ are estimated online, and updated using the LMS principle. The weight update equations for the weights $\mathbf{W}$ and the $A_j$ are

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \alpha Err(t)\mathbf{\Phi}^T, \qquad (16)$$

$$A_j(k+1) = A_j(k) + \alpha Err(t)U(t-1)^T. \qquad (17)$$

where the plant model mismatch $Err(t)$ is,

$$Err(t) = Y(t) - Y_m(t). \qquad (18)$$

$k$ is the iteration index and $\alpha$ is the LMS learning rate.

### *3.2 Algorithm Summary*

The proposed algorithm can be implemented as follows:

- generate plant output $Y(k)$ and compute model output $Y_m(k)$ using Eq. 5,
- calculate mismatch $Err(k)$,
- update the weights $\mathbf{W}$ and $A_1$ using Eq. 16 and Eq. 17,
- generate control move $U_{k-1}$ using the updated values of $\mathbf{W}$ and $A_1$ in Eq. 13,
- go back to first step.

### *3.3 Stability*

In IMC structure the closed-loop remains stable, if both the plant and controller are stable [2]. In adaptive IMC structure, closed-loop stability is assured if the parameter estimation law, the controller and the plant all are stable stable [17]

In the U-model based adaptive IMC structure the controller has two parts. a)the estimated U-model and b) inverse of U-model. To ensure the stability of the controller, it is necessary to show that the estimated U-model converge to the true stable (or stabilized) plant. The convergence of the U-model relies on the selection of a suitable learning rate that guarantees robust and faster convergence speeds in the presence of noise and load variations. Therefore, the adaptation of the parameters of (SISO) U-model is associated with lossless mapping in a feedback structure shown in Fig. 3.

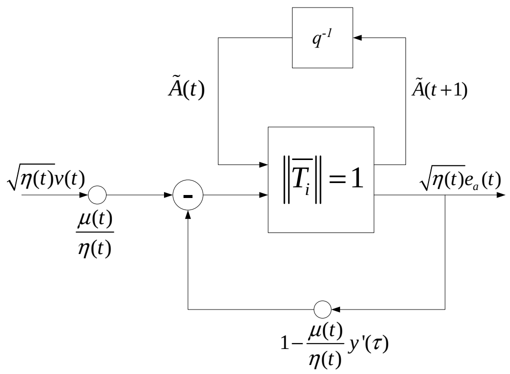


Fig. 3. A lossless mapping in a feedback structure for the learning algorithm of U-model

The feedback structure is similar to the feedback structure presented in [18] for the analysis of adaptive filters. The stability of the feedback structure is analyzed [19] using the small gain theorem that resulted in an optimal learning rate guaranteeing robust and faster convergence speeds for stable operation. The optimal learning rate is given by,

$$\mu(t) < \eta(t) \sum_{j=1}^{M} j\alpha_j(\tau) u^{j-1}(t-1), \qquad (19)$$

where, $\eta(t)$ is reciprocal of the squared norm of the U-model input vector $[1 \;\; u(t-1) \; u^2(t-1) \; \ldots \; u^M(t-1)]$. $\tilde{A}(t)$ is the parameter error matrix at instant $t$.

## 4. REAL-TIME IMPLEMENTATION

The proposed MIMO RBFNN based U-Model IMC scheme is tested on an experimental setup, developed for the verification of different algorithms.

### *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. 4.

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. 5


Fig. 4. The Real-Time setup for the 2 link Robot

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.

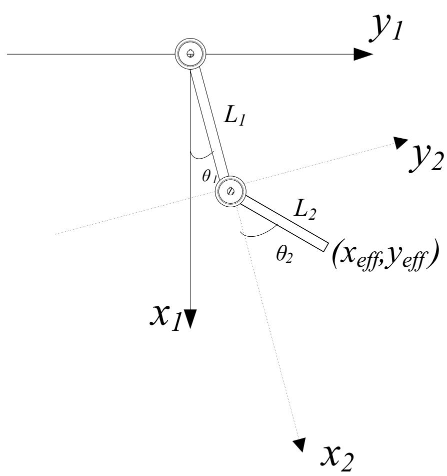


Fig. 5. The 2 link Robot Geometry

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.

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.

$$B = \sqrt{x^2 + y^2} \qquad (20)$$

The angles $\theta_1$ and $\theta_2$ are calculated by,

$$\theta_1 = Atan2(y_d/x_d) + cos^{-1}[(L_1^2 - L_2^2 + B^2)/2L_1B], \quad (21)$$

$$\theta_2 = cos^{-1}[(L_1^2 + L_2^2 - B^2)/2L_1L_2], \quad (22)$$

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 $\theta_1$ and $\theta_2$, the end effector coordinates $x_{eff}$ and $y_{eff}$ are,

$$x_{eff} = L1\cos(\theta_1) + L2\cos(\theta_1 + \theta_2), \quad (23)$$

$$y_{eff} = L1\sin(\theta_1) + L2\sin(\theta_1 + \theta_2). \quad (24)$$

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. 6. 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 based adaptive tracking scheme

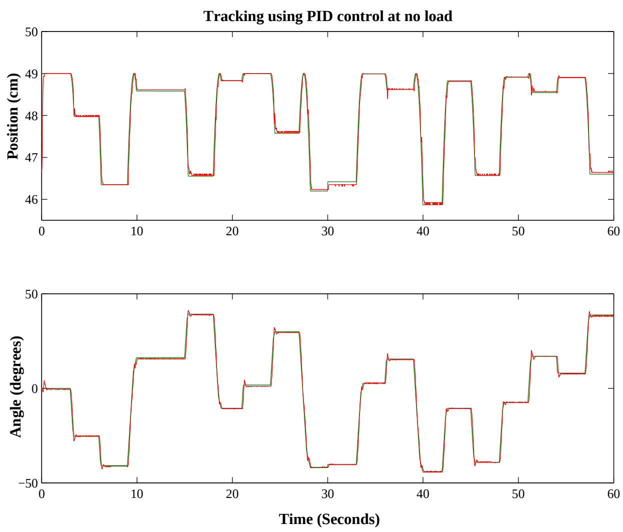


Fig. 6. Tracking using a PID controller at no load

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. 7. Fine tracking performance can be observed with no overshoot and mistracking as compared to the standard PID controller.

Using the same setup, with a varying load, the performance of the standard PID controller tuned at no load is shown in Fig. 8 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

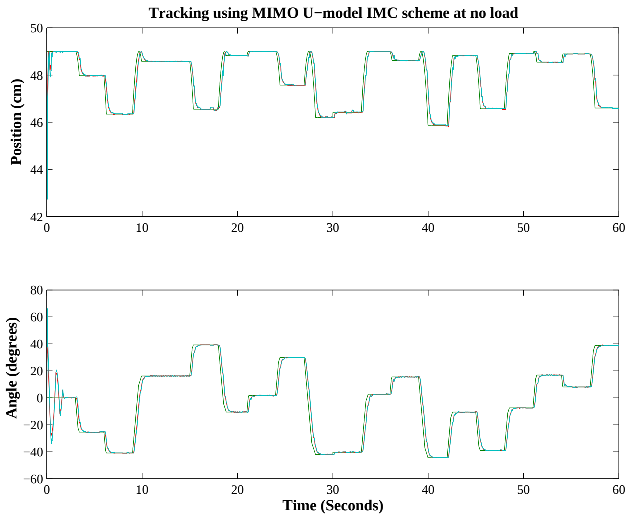


Fig. 7. Tracking using the proposed U-model scheme at no load

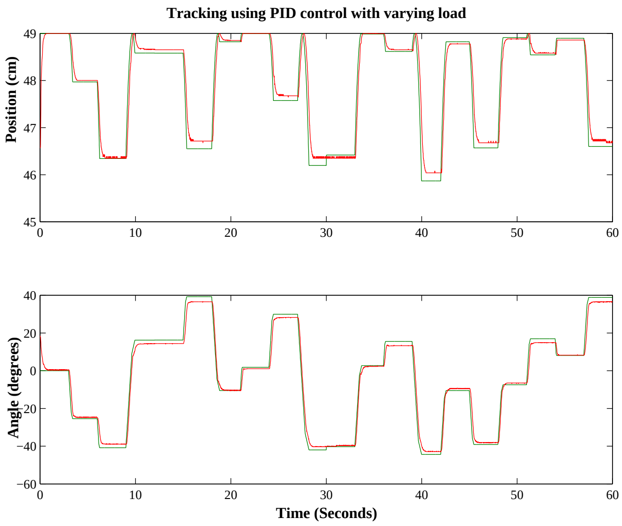


Fig. 8. Tracking with variable load using a PID controller tuned for no load

load setup, very similar tracking results were observed as depicted in Fig. 9. This shows the robustness of the adaptive scheme that is able to perform even with load variation.

## 5. CONCLUSION

A new technique is introduced for the internal model control of unknown MIMO nonlinear systems. MIMO U-model is proposed in the control scheme for the online identification of the unknown MIMO plant. The controller is developed based on the U-model methodology using the Newton-Raphson method. The proposed technique adequately simplifies the synthesis of control law that is directly derived from the model. The proposed scheme is tested on a case study of a 2 link robot manipulator and is compared with the standard PID controller. The experimental results show that the proposed scheme simply outperformed the standard PID controller under no load and varying load conditions.

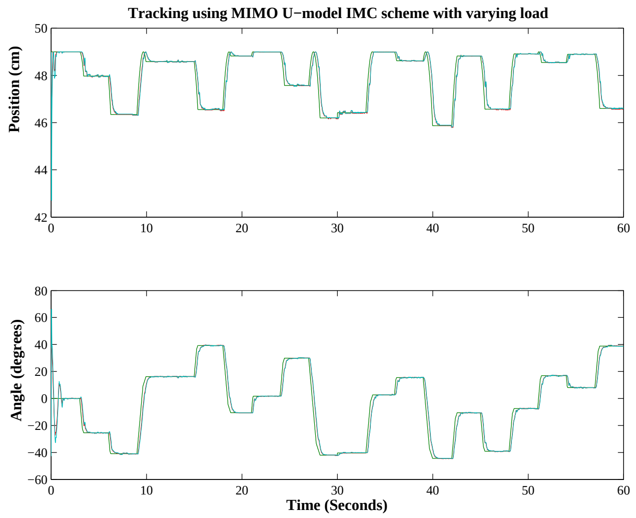


Fig. 9. Tracking with variable load using the proposed U-model scheme

REFERENCES


[1] Garcia , C. E. and Morari, M., "Internal Model Control. 1. A unifying review and some new results", Ind. Eng. Chem. Process Des. Dev., vol. 21, pp. 308-323, 1982.

[2] Morari, M. and Zafiriou, "Robust Process Control," *Prentice Hall*, March, 1997.

[3] Economou, C. G., Morari, and Palsson, B. O. , "Internal model control and extension to nonlinear systems," *Ind. Eng. Chem. Process Des. Dev.*,vol. 25, no. I, pp. 403-411, 1986.

[4] Rivals, I. and Personnaz, L., "Nonlinear internal model control using neural networks: application to processes with delay and design issues," *IEEE Transactions on Neural Networks* , vol. 11, no. 1, pp. 80 - 90, Jan 2000.

[5] Pu H., Bao-Hai H., Dong-Feng W., Yu H. and Yong-Ling L., "Internal Model Control of Superheated Steam Temperature System Using T-S Fuzzy Model," *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, Vol. 5, pp. 2603-2607, 2005.

[6] Shafiq, M. and Riyaz, S. H., "Internal model control structure using adaptive inverse control strategy," *in The 4th Int. Conf. on Control and Automation (ICCA)*, 2003, pp. 59-65.

[7] Wen, X., Zhang, J., Z. Zhao, Z. and Liu, L., "Multi-model neural network IMC," *in Proceedings of 2004 International Conference on Machine Learning and Cybernetics* , vol. 6, Aug 2004, pp. 3370 - 3374.

[8] 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.

[9] 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

[10] Ali, S. Saad Azhar, Fouad M. Al-Sunni and Shafiq M., "U-model based adaptive tracking scheme for unknown MIMO bilinear systems," $1^{st}$ *IEEE conference on Industrial Electronics & Applications*, May 2006.

[11] 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.

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

[13] 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.

[14] 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.

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

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

[17] Aniruddha, D., "Adaptive Internal Model Control", *Springer-Verlag*, June, 1998.

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

[19] Ali, S. Saad Azhar, M. Shafiq, F. M. AL-Sunni, and J. M. Bakhashwain, Feedback analysis of U-model via small gain theorem, $10^{th}$ *WSEAS International Conference on Automatic Control, Modeling and Simulation, Istanbul, Turkey*, May 2008.