LEARNING FEEDFORWARD CONTROL OF MIMO NONLINEAR SYSTEMS USING U-MODEL

Ali, S. Saad Azhar†, Fouad M. Al-Sunni‡, Muhammad Shafiq‡ and Bakhashwain, Jamil M.†
†Electrical Engineering Department, and ‡Systems Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia.
email: {saadali, alsunni, mshafiq, jamilb}@kfupm.edu.sa

ABSTRACT

In this paper, a learning feedforward controller (LFFC) using the U-model is proposed for a better tracking control of multivariable nonlinear systems over a finite time interval. The multivariable system is modelled using the U-model and the LFFC is established using Newton-Raphson method. U-model significantly simplifies the online synthesis of the feedforward control law. The proposed technique is verified on 2-link robot manipulator in real-time. The performance of the proposed U-model based LFFC is compared with a number of schemes under varying load conditions.

KEY WORDS

MIMO Nonlinear systems, LFFC, U-model, RBFNN, robotics.

1 Introduction

Feedback control is often applied to improve the dynamical behavior of electromechanical systems. Feedback controllers are usually designed on the basis of a process model. When only an approximate process model is available, a sub-optimal performance may result. Several methods can be used to improve the performance of feedback controllers. Commonly, a more detailed process model is derived. However, this method has two disadvantages.

• For complex processes, obtaining a quantitative model that is suited for the controller design might be difficult and time consuming. In some cases, e.g. in case of process uncertainties, proper identification is even impossible.

• The feedback controller has to provide for both a high performance and robust stability. This usually results in a trade off between the two properties. A feedback controller that has a high performance often does not feature a robust stability. Small variations in process parameters may destabilize the system.

The latter problem can be overcome by creating separate means for obtaining high performance and robust stability using an additional learning feedforward controller, Fig. 1 [1]. By adding a (learning) feedforward component to the feedback controller, an extra degree of freedom in the design of the controller is created. The feedforward part is intended to generate steering signals that make the output of the process $y(t)$ follow the reference. This control concept is known as Learning feedforward Control (LFFC) [1],[2].

LFFC can be regarded as a variant of Iterative Learning Control (ILC) [3], [4]. A learning feedforward controller (LFFC) using the Bartlet window function is proposed in [5] for a better tracking control of linear system over a finite time interval. LFFC is applied as a feedforward controller to the existing feedback controller. In [1], the robustness and easy design of a PD-feedback controller was combined with a learning control strategy for improved tracking performance. The learning feedforward controller was designed using the B-spline networks (BSN). A parsimonious (reduced dimensionality) LFFC has been applied to a linear motor motion system in [6]. The BSN based LFFC is applied to linear motion control in [7]. [8] presents a frequency-domain analysis and design approach for LFFC using a dilated B-spline network. In [9], a support vector machine is proposed as the learning mechanism.

In order to design an accurate feedforward controller, the process dynamics need to be known. The feedforward controller has to compensate for the process dynamics and disturbances. These requirements call for plant inverse. For a complex nonlinear or non-minimum phase plant inverse finding poses theoret-
ical and implementation difficulties. Several model based inversion techniques such as based on NARMAX model [10] or Hammerstein model [11] lead to complex nonlinear algebraic equations.

However the U-model proposed by [12] 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 and the Hammerstein model. U-model is 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 \( Y(t−1), \ldots Y(t−n) \), 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. U-model has been used in the internal model control scheme for single and multivariable nonlinear systems [13, 14].

In this paper, we propose a novel LFFC design approach. MIMO nonlinear dynamic plant is identified online using the radial basis functions neural networks (RBFNN) in U-model structure. Then the inverse of the model is obtained by Newton-Raphson root solving technique. This inverse is embedded in the nonlinear IMC scheme. This approach gives same design procedure for both invertible and non-invertible nonlinear systems.

The rest of the 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 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. Finally the contributions are concluded in section 6.

1.1 Problem Statement

Consider the NARMAX representation of the MIMO nonlinear plants as,

\[
Y(t) = F(Y(t−1), \ldots, Y(t−n), U(t−1), \ldots, U(t−m), E(t−1), \ldots, E(t−l)), \tag{1}
\]

where \( Y(t) \) and \( U(t) \) are the output and input signals of the plant respectively at a discrete time instant \( t \), \( E(t) \) represents the error due to measurement noise, model mismatch, uncertain dynamics and plant variation. \( F(\cdot) \) 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 an arbitrary piece-wise continuous trajectory \( R(t) \). Further it is considered that the plant parameters are unknown and slowly time varying.

2 The U-model Structure

The feedforward controller in the proposed scheme is established based on the U-model methodology. The MIMO nonlinear plant is identified online in U-model format. Then the tracking inverse controller is obtained using the Newton-Raphson root solving techniques. The SISO U-model used for internal model control of a SISO plant in [13] models a plant of NARMAX representation given by,

\[
y(t) = f(y(t−1), \ldots, y(t−n), u(t−1), \ldots, u(t−m), e(t−1), \ldots, e(t−l)), \tag{2}
\]

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

\[
y_m(t) = \sum_{j=0}^{M} \alpha_j(t)u^j(t−1) + e(t), \tag{3}
\]

where \( M \) is the degree of model input \( u(t−1) \), \( \alpha_j(t) \) is a function of past inputs and outputs \( u(t−2), \ldots, u(t−m) \), \( y(t−1), \ldots, y(t−n) \) and errors \( e(t), \ldots, e(t−l) \).

2.1 Advantages of U-Model

There are several advantages of using the U-model over other many other approaches, such as [12]:

- The control-oriented U-Model is more general than other parameterizing approaches, such as the polynomial NARMAX model, the Hammerstein model etc.
- The sampled data representation of many nonlinear continuous time systems has a structure similar to U-model.
- The U-model exhibits a polynomial structure in the current control signal \( U(t−1) \).
- Due to its polynomial structure, the nonlinear algebraic equations, which need to be solved to synthesize the control signal, are also polynomials in \( U(t−1) \), unlike other models which lead to complex nonlinear algebraic equations. The structural simplicity leads to reduction in the effort for numerical computations.
- Since only an approximate inverse is used in the control structure, several theoretical difficulties are taken care of, such as inverting non-minimum phase systems or even a non-invertible system.

3 U-model based LFFC

The proposed U-model based LFFC for multivariable nonlinear system is shown in Fig. 2. The multivariable
nonlinear plant is modelled in U-model format online and the inverse is established using the Newton-Raphson Method. \( R(t) \) is the arbitrary reference trajectory. \( U(t - 1) \) is the control input, \( Y(t) \) and \( Y_m(t) \) are plant and model outputs.

![Figure 2. U-model based Learning Feedforward Control Scheme](image)

### 3.1 Proposed MIMO U-model

The SISO U-model is extended for multivariable systems as,

\[
Y_m(t) = A_0(t) + A_1(t) \frac{1}{t} U(t - 1) + A_2(t) \frac{2}{(t - 1)} + \ldots ,
\]

or

\[
Y_m(t) = \sum_{j=0}^{M} A_j(t) \frac{j}{(t - 1)} = F(U(t - 1)).
\]

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. \( U(t) \) is the vector with \( j^{th} \) power of the control inputs \( u_i(t - 1) \) as,

\[
\hat{U}(t - 1) = [u_1^j(t - 1) \ u_2^j(t - 1) \ \ldots \ u_p^j(t - 1)]^T,
\]

and \( A_j(t) \) are matrices instead of scalars \( \alpha_j \). For a system with unknown parameters, the matrices \( A_j \) are estimated online using gradient descent adaptive algorithm. In this work, normalized least means squares (nLMS) principle [15] is adopted for the update of the parameters. However, the matrix \( A_0 \) is modelled using radial basis functions neural networks (RBFNN). The reason for incorporating the RBFNN is to assist the nonlinear modelling. Neural networks are capable of learning models of backlash [16], saturation [17] and deadzone [18]. These nonlinearities may be present in cascade with the actual nonlinear system, particularly in actuators and sensors. The effect of these unknown nonlinearities can be compensated using the neural networks such as multilayered feedforward neural networks (MFNN) and RBFNN.

#### 3.1.1 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 [19]. In these networks the learning involves only one layer with lesser computations. An \( m \) input \( p \) output RBFNN is shown in Fig. 3.

![Figure 3. A MIMO RBF neural network.](image)
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 $W = [W_1 \ldots W_p]^T$ and $\Phi = [\phi_1 \ldots \phi_p]$, the MIMO RBFNN is expressed in matrix notation neatly as,

$$Y(t) = W\Phi(t). \quad (7)$$

The weights of the RBFNN and the rest of the parameters $A_j$ are estimated online, and updated using the normalized least mean square (nLMS) principle [15]. The weight update equations for the weights $W$ and the $A_j$ are

$$W(t + 1) = W(t) + \mu(t)Err(t)\Phi(t)^T, \quad (8)$$
$$A_j(t + 1) = A_j(t) + \mu(t)Err(t)\frac{\partial f_j}{\partial u_j}(t - 1)^T, \quad (9)$$

where $\mu(t)$ is the nLMS learning rate.

### 3.2 The Newton-Raphson based Controller

The feedforward controller is established if the controller-model cascade results in unity gain. This implies that if the controller is excited by the signal $Y_m(t)$, the control signal $U(t - 1)$ will force the model and so the plant to track $Y_m(t)$. Therefore, reference tracking can be achieved by setting

$$R(t) = Y_m(t) = F(U(t - 1)) \quad (10)$$

$$F(U(t - 1)) = R(t). \quad (11)$$

Eq. 11 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 [20]. 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,

$$U_{k+1}(t - 1) = U_k(t - 1) + H, \quad (12)$$

which satisfies

$$F(U_{k+1}(t - 1)) = F(U_k(t - 1) + H) = R(t). \quad (13)$$

Now expanding the nonlinear function $F(U_k(t - 1) + H)$ using the Taylor series expansion and considering only the linear first order terms [20],

$$F(U_k(t - 1) + H) \approx F(U_k(t - 1)) + F'(U_k(t - 1))H. \quad (14)$$

The term $F'(U_k(t - 1))$ is the Jacobian matrix with elements $\partial f_j/\partial u_{jk}(t - 1)$, corresponding to the $j^{th}$ input and $k^{th}$ output.

Using Eq. 13 in Eq. 14, the value of the correction vector $H$ can be obtained as,

$$H = F'(U_k(t - 1))^{-1}(R(t) - F(U_k(t - 1))) \quad (15)$$

or

$$H = F'(U_k(t - 1))^{-1}(R(t) - Y_m(t)) \quad (16)$$

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

$$U_{k+1}(t - 1) = U_k(t - 1) + F'(U_k(t - 1))^{-1}(E(t) - Y_m(t)). \quad (17)$$

Considering the solution of the Newton-Raphson based controller, it is evident that existence of Jacobian for the $F(U(t - 1))$ is a necessary condition, which might not always be true. However, in case of U-model, where the $F(\cdot)$ itself is a polynomial function of the control signal $U(t - 1)$, the Jacobian will also be containing elements that are polynomial in $U(t - 1)$. Hence, the condition for existence is satisfied in addition to the computational ease.

### 3.3 Algorithm Summary

The proposed algorithm can be implemented as follows:

1. Measure plant output $Y(t)$ and compute model output $Y_m(t)$ using Eq. 4.
2. Calculate mismatch $Err(t)$,
3. Update the weights $W(t)$ and $A_1(t)$ using Eq. 8 and Eq. 9,
4. Synthesize control move $U(t - 1)$ using the updated values of $W(t)$ and $A_1(t)$ in Eq. 17,
5. Go back to step 1.

### 4 Real-Time Implementation

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

#### 4.1 The Real-Time Setup

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. The primary link is made of Aluminium and the secondary link is made of Plastic. The primary link is actuated by a geared motor HN-GH 27, with a gear ratio of 50:1 and an allowable maximum torque of 2.3 KG-cm. The secondary link is actuated by another geared motor
HN-GH 12, with a gear ratio of 188:1 and an allowable maximum torque of 1.4 KG-cm. The geometry of the 2 link robot is shown in Fig. 5.

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 \( \pm 60^\circ \) from the central position. However, the secondary link can maneuver the whole \( \pm 180^\circ \) rotation. The workspace for the developed 2 link robot is shown in Fig. 6.

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.

For the 2 link robot having lengths \( L_1 \) and \( L_2 \), the inverse kinematics problem is defined by, 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.

The function \( \text{Atan2}(Y/X) \) 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,

\[
x_{eff} = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2),
\]
\[
y_{eff} = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2).
\]
position of the link. Tension is higher with a higher rotation angle. The proposed U-model based LFFC 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 $A_0$. 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 reference signal is set to be a random piece-wise continuous signal with a step time of 3 seconds. The tracking is shown in Fig. 7. The proposed scheme that is able to perform even with load variation which is a common practice in robotics. The perfor-

![Tracking using MIMO U−model based LFFC at no load](image1)

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

![Tracking using MIMO U−model based LFFC at varying load](image2)

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

mance of the proposed U-model based scheme is also compared with a number of schemes; such as nonlinear PID controller, adaptive PD controller, adaptive inverse control scheme. Fig. 9 shows a comparison in the mean squared errors sense. The comparison shows that the proposed scheme has performed better under varying load conditions. The U-model based LFFC scheme is also compared with the U-model based IMC. An initial faster convergence was observed using the U-model based IMC scheme, however, the LFFC scheme converged to smaller MSE, despite a little slower initial convergence.

The successful implementation of the proposed U-model based LFFC on real system shows the usefulness of the scheme. The proposed scheme can be a substitute to the already existing classical controllers implemented in the industry.

![Comparison in the Mean Squared Error Sense](image3)

Figure 9. Tracking MSE Comparison

5 Conclusion

A new technique is introduced for the learning feed-forward 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 several schemes. The comparison depicted better tracking performance using the U-model based LFFC.
Acknowledgments

The authors acknowledge the support of King Fahd University of Petroleum & Minerals and SABIC for funding this work under project SABIC 2006-11.

References


