IDENTIFICATION METHODS FOR ADAPTIVE MANIPULATOR CONTROL

F. Jones and J.V. Ringwood

School of Electronic Engineering,
Dublin City University,
Glasnevin, Dublin 9.

ABSTRACT

Models for industrial robots are characterized by highly nonlinear equations with nonlinear coupling between the variables of motion. This paper examines possible model structures suitable for identification of manipulator dynamics. Attention is focussed on single input/single output models with a view to the implementation of decentralized control. Both linear and simple nonlinear models are considered. The nonlinear models are based on the Hammerstein and Volterra structures for which modified linear control methods have already been developed. Implementation results from these methods are presented for a PUMA 560 robot to demonstrate the applicability of each method. Some implementation details are also discussed.

1. Introduction

Robotic manipulators are highly coupled, nonlinear mechanical systems designed to perform specific tasks. The control problem centres around the computation of the control voltages and torques required to execute these tasks. One of the major difficulties in tackling this control problem is the lack of design methodologies for complex nonlinear systems.

Adaptive control has been suggested as a solution, in which a linear model is updated at frequent intervals, providing a linear representation for the system at different operating points. The resulting linearized plant models yield time varying plant parameters which may be transformed to controller parameters using linear systems design methodologies. In addition, the varying linear models may be parameterized to take account of coupling between different manipulator joints, alleviating to some degree the need to implement a true multivariable controller.

The main objective of this paper is to examine various robot models, linear and nonlinear, to assess their suitability for implementing adaptive controllers. This is achieved by using the least squares (LS) identification technique to identify the parameters of these models for a PUMA 560 industrial robot.

The paper is organized as follows: First, the linear and nonlinear model structures are presented. This is followed in each case by a description of how the LS identification technique can be applied to each model. The application of these LS models to the PUMA 560 robot is then detailed. This is followed by a comparison of the properties of these identification models. Finally, the paper draws some conclusions based on the comparison of models.

2. Identification Models for Manipulator Robots

The dynamic control of an industrial manipulator involves the determination of the inputs (torques or voltages) for the actuators which operate at the joints so that a set of desired values for the positions and velocities for the manipulator is achieved. Virtually all forms of dynamic control involve the use of a system model for the design of control algorithms. In the case of adaptive/self tuning control, the model used is generally a discretized one which takes the form of a time series model containing any linear and nonlinear terms which are present in the system. A general time series model can be assumed for each joint as follows:

$$y(kT) = A_0 + A_1 y[(k-1)T] + A_2 y[(k-2)T] \ldots\ldots + B_1 u[(k-1)T] + B_2 u[(k-2)T] \ldots\ldots + f[kT] + M(kT) \qquad (1)$$

where u(kT) is the model input, or joint voltage, and y(kT) is the output or joint position at time kT. A_i and B_i are coefficients of the linear portion of the model, $f(.)$ is the discretized joint nonlinearities contained in the torque terms of the robot model and M(.) represents modelling errors.

2.1 Linear Models for Manipulator Identification

An ARMA Identification Model

By assuming the coupling terms are small and that the robot's system parameters are slowly time-varying [1] with negligible measurement noise, it is possible to assume an ARMA model representation of the robot's dynamics. This model can be written as:

$$y(k) = A(q^{-1})y(k) + B(q^{-1})u(k) + e(k) \tag{2}$$

If the parameter vector Θ and the regressor information vector Φ are defined as

$$\theta^T = (a_1, \ldots, a_n; b_1, \ldots, b_n) \tag{3}$$

$$\Phi^T = [\ y(k-1), \ldots, y(k-n);\ u(k), \ldots, u(k-n+1)] \tag{4}$$

the model can then be written as:

$$y(k) = \theta^T . \Phi(k-1) + e(k) \tag{5}$$

The parameter estimation problem is to find the estimates of the unknown parameters which minimize the cumulative loss function:

$$E(\theta_i) = \frac{1}{m+1} \sum_{i=1}^{m} [e_i(k)]^2 \tag{6}$$

where $e_i(t)$ is the prediction error in the parameters of joint i and m is the number of parameters being estimated. The solution to the Least Squares problem is furnished by the following recursive equations [2]:

$$\theta(k) = \theta(k-1) + P(k)\Phi(k-1).[\ y(k) - \theta^T(k-1)\Phi(k-1)] \tag{7}$$

$$P(k) = \frac{1}{\mu} \left[P(k-1) - \frac{P(k-1)\Phi(k-1)\Phi^T(k-1)P(k-1)}{\mu + \Phi^T(k-1)P(k-1)\Phi(k-1)} \right] \tag{8}$$

where P is the covarience matrix (2nx2n) of the estimation errors and μ is known as the forgetting factor which discounts old data.

A Modified ARMA Identification Model

This method of is developed from the ARMA model just described. This more comprehensive autoregressive model can be written as:

$$y(k) = A(q^{-1})y(k) + B(q^{-1})u(k) + h + e(k) \tag{9}$$

where h is a forcing term intended to include the nonlinearities in the robot. In this case, the parameter estimates and the regressors can be written in the following vector format:

$$\theta^T = (a_1, \ldots, a_n; b_1, \ldots, b_n; h_1) \tag{10}$$

$$\Phi^T = [\ y(k-1), \ldots, y(k-n);\ u(k), \ldots, u(k-n+1); 1] \tag{11}$$

The autoregressive model can be again written as in equation (5). This is the format required to apply the loss function equation for the minimization of the prediction error.

An ARMAX Identification Model

This method attempts to estimate a model for the noise present in the system, as well as the system model itself. This model can be written in time series form as follows:

$$y(k) = A(q^{-1})y(k) + B(q^{-1})u(k) + C(q^{-1})e(k) + d(k) \tag{12}$$

where $C(q^{-1})$ is the polynomial containing the parameters of the noise model and $d(k)$ is called the loaded disturbance variable. In this case, the parameter estimates and the regressors can be written in the following vector format:

$$\theta^T = (a_1,\ldots,a_n;b_1,\ldots,b_n;c_1,\ldots,c_n) \tag{13}$$

$$\phi^T = [\, y(k-1),\ldots,y(k-n);\ u(k),\ldots,u(k-n+1);e(k),\ldots,e(k-n)] \tag{14}$$

The autoregressive model can be written as in equation (5).

2.2 Nonlinear Methods For Manipulator Identification

An ARMAX Model with a Hammerstein Nonlinearity (HARMAX)

This method [3] attempts to estimate a model for the residual as a combination of linear and nonlinear functions. This model can be written as follows:

$$y(k) = A(q^{-1})y(k) + B(q^{-1})u(k) + C(q^{-1})e(k) + N(k) \tag{15}$$

where $C(q^{-1})$is the polynomial containing the parameters of the noise model and $N(k)$ is a nonlinear polynomial defined by:

$$N(k) = n_1u^2(k) + n_2u^3(k)\ldots\ldots\ldots n_m u^{m+1}(k) \tag{16}$$

The parameter estimates and the regressors can be written in the following vector format:

$$\theta^T = (a_1,\ldots,a_n;b_1,\ldots,b_n;c_1,\ldots,c_n;n_1,n_2) \tag{17}$$

$$\begin{aligned}\phi^T = [\, & y(k-1),\ldots,y(k-n);\ u(k),\ldots,u(k-n+1); \\ & e(k),\ldots,e(k-n);u^2(k),u^3(k)]\end{aligned} \tag{18}$$

The autoregressive model can be again written as in equation (5).

An ARMAX Model with a Volterra Nonlinearity (VARMAX)

This method attempts to estimate a model for the residual as a combination of linear and nonlinear functions. This model can be written as follows:

$$y(k) = A(q^{-1})y(k) - B(q^{-1})x(k) + C(q^{-1})e(k) \tag{19}$$

where $C(q^{-1})$is the polynomial containing the parameters of the noise model and $x(k)$ is a nonlinear element defined by:

$$x(k) = u^T B_n u \tag{20}$$

where $u^T = (u(t), u(t-1),\ldots\ldots,u(t-m))$; (21)

and
$$B_n = \begin{vmatrix} B_{00} & B_{01} & \ldots\ldots & B_{0m} \\ 0 & B_{11} & \ldots\ldots & B_{1m} \\ . & . & . & . \\ 0 & & \ldots\ldots\ldots\ldots & B_{mm} \end{vmatrix} \tag{22}$$

This model can then be fitted into the the autoregressive model of equation (5) in a similar way to the HARMAX method.

3. Identification Results for the PUMA 560

To obtain comprehensive information about the parameters of the PUMA 560 it was decided to use similar joint trajectories to those used in [4]. These test trajectories (see figures 1 and 2) were used because according to [4] they provide a good insight into the dynamic characteristics of the PUMA 560. The tests used can be broken down into two blocks:

1) slow trajectory (i.e. 50% of max. joint speed) unloaded, and
2) fast trajectory (i.e. max. joint speed) unloaded.

The parameters of the models detailed in Section 2 were estimated from input/output (ie. joint voltage/joint position) data pairs gathered on-line from the PUMA 560. A new control structure [5] consisting of an Intel 80386-based host computer and three NEC μP77230 floating-point DSP cards was used to gather the the data pairs.
The implementation results for these tests are presented as follows: Figures 3 to 7 show a representative sample of the cumulative loss functions (see equation (6)) obtained for the linear and nonlinear identifications methods. Figures 8 to 10 show a sample parameter convergence observed from the identification experiments preformed.

The parameters and loss functions in these figures were obtained using a forgetting factor of 0.95. The sampling interval chosen for the input/output pairs was 5ms. A second order model structure (n=2, m=2 see Table 1) was chosen for the identification models. This was dictated by the fact that increasing the model order showed little improvement in the the accuracy of the models identified.

TABLE 1 Identification Implementation Details

METHOD	No. of PARAMETERS	n=2 & m=2 mult	n=2 & m=2 add	NEC μPD77230 execution times
ARMA	n*2	42	45	24μs
MARMA	n*2+1	65	62	34.85μs
ARMAX	n*3	83	81	45.05μs
HARMAX	n*3+m	163	125	78.5μs
VARMAX	$n*3 + \sum_{1}^{m+1} i$	429	269	108.05μs

4. Comparative Properties of the Identification Methods

The results of section 3 show that the ARMA model produces the largest cmulative loss function values and therefore the least accurate model. The MARMA model identification produces identical a_i and b_i model parameters to the ARMA model. The addition of the the h_1 parameter (see Figure 8 for example) to model the ARMA residual has the effect of reducing the model errors threefold. The ARMAX method models this residual using a noise model. This proves successful at reducing the estimation loss function by a factor of 10 over the ARMA model. The HARMAX model has the effect of reducing the model errors even further. This indicates some dependency of the robot model on past and present inputs and their squares. The VARMAX model can be seen to have the lowest values of loss functions. This indicates the robot model is also dependent on the the products of past and present inputs.

By obesrving the loss function curves it is also possible draw some conclusions about the convergence of the model parameters. In the ARMA and MARMA cases the loss functions show a rapid initial increases in their loss function values. This indicates that the initial model errors are high and so parameter convergence is will be slow. This can be seen more clearly by comparing the convergence rates of the ARMA and ARMAX parameters in Figures 9 and 10. The introduction of good initial parameter estimates was found to reduce the initial increase in the loss function and decrease parameter convergence times. In the cases of ARMAX and nonlinear models this initial rise in the loss function was much less pronounced which indicates that convergence occurs rapidly even in the absence of good initial estimates.

The ability of an identification model to track parameter variations can be seen by examining the rate of change its loss function. The VARMAX method shows the lowest rate of change of the loss functions over the test trajectories. From this it can be inferred that the the modeling errors for this method tend to zero. This implies that this method has the ability to track any time varing parameters in the system.

The advantages gained by using the VARMAX model are somewhat negated by the large amount of computation required (see Table 1) to implement even a second order VARMAX identification algorithm. If, however, the control hardware described briefly in Section 3 is used then from Table 1 the implementation time for this method is 108µs. Since the existing PUMA 560 sampling period is approximately 1ms, this leaves almost 90% of the sampling interval for the implementation of the desired control law. One control law which has been developed for this identification method is NLQG contoller detailed by Grimble[3]. Preliminary calculations indicate that implementation of this algorithm could take 0.6ms. This when combined with implementation time for the identification gives a total implementation time of 0.7 ms. This is within the sampling period necessary to control the PUMA robot.

5. Conclusions

This paper has outlined several linear and nonlinear time series models which can be used to design adaptive robot controllers. It shows in each case how the parameters of these models can be estimated using a recursive least squares identification technique. Through the use of implementation results on a PUMA 560 robot arm, the paper examined the ability of these methods to accurately model a robotic system.

The method of VARMAX was found to model the robot most accurately with rapid parameter convergence. This method involves the estimation of a large number of parameters making real-time implementation difficult. A solution involving the use DSPs is shown to be capable of implementing a control law which uses the VARMAX parameters in real time.

References

[1] Koivo A.J. & Guo T-H., "Adaptive Linear Controllers for Robotic Manipulators", IEEE Trans. on Automatic Control, Vol. AC-28, No 2, Feb 1983.
[2] Ljung L. & Soderstrom T., "Theory and Practice of Recursive Identification", MIT Press, 1983.
[3] Carr S. & Grimble M.J. "Nonlinear Quadratic Gaussian Self-Tuning Control", ICU Report 228, Industrial Control Unit , University of Strathclyde, Glasgow, Nov. 1988.
[4] Leahy M.B. Jnr., "Performance Characterization of a PUMA 600 Robot", RAL Technical Report, No. 56, RPI, Sept 1985.
[5] Jones F., " Modelling, Simulation and Identification of an Industrial Manipulator", M. Eng. Thesis, Electronic Eng. Dept., DCU, Dublin, Sept 1990.

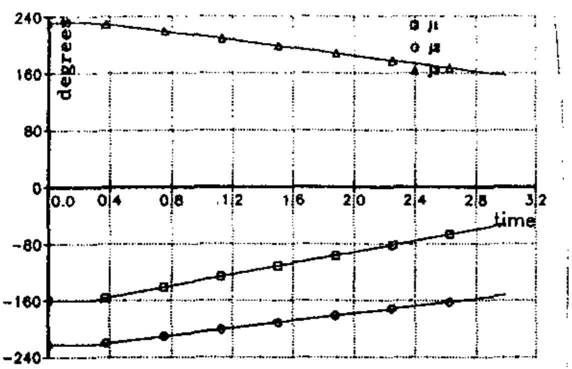

Fig 1: Slow trajectories (unloaded)

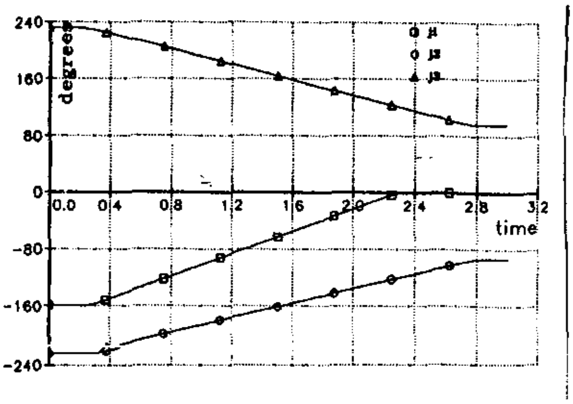

Fig 2: Fast trajectories (unloaded)

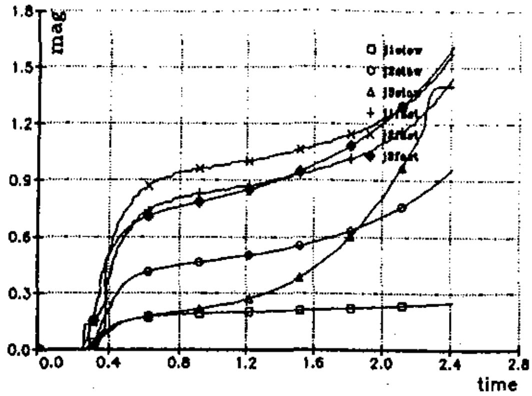

Fig 3: Arma Cumulative Loss Functions

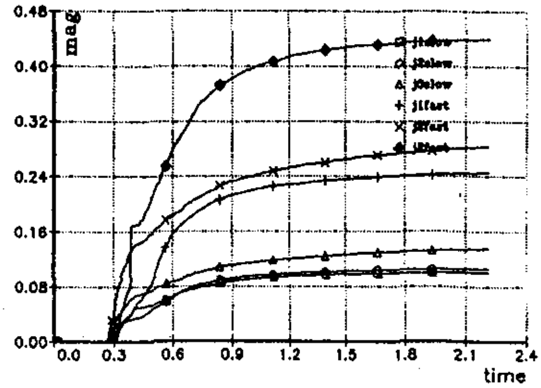

Fig 4: MARMA Cumulative Loss Functions

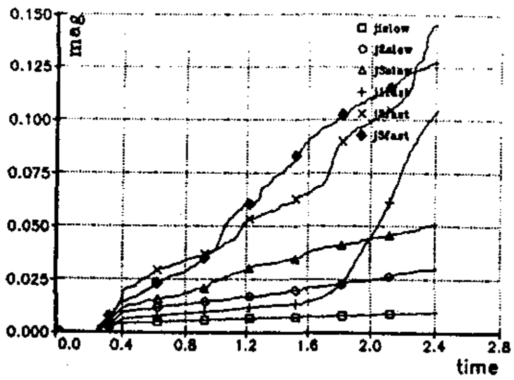

Fig 5: ARMAX Cumulative Loss Functions

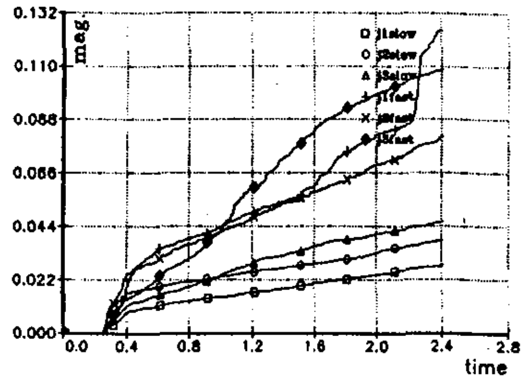

Fig 6: HARMAX Cumulative Loss Functions

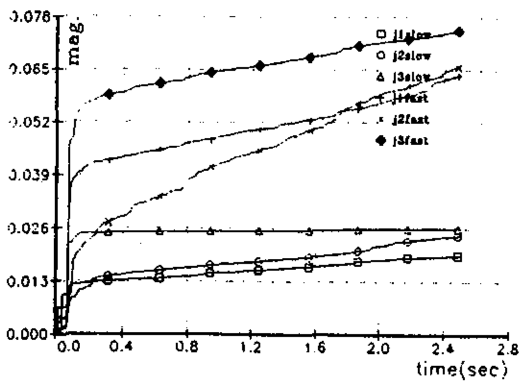

Fig 7: VARMAX Cumulative Loss Functions

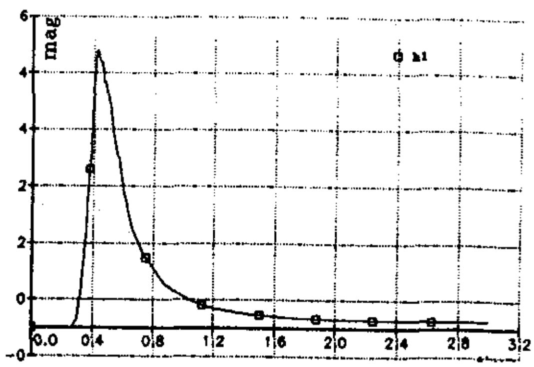

Fig 8: MARMA Sample hi Parameter

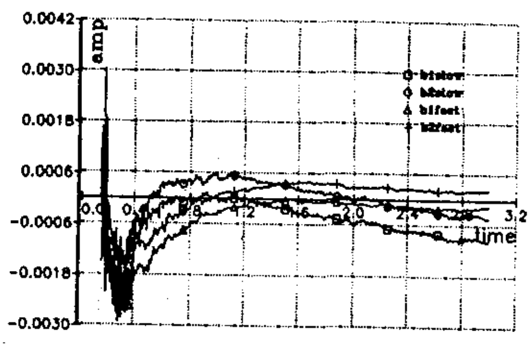

Fig 9: Sample ARMA convergence

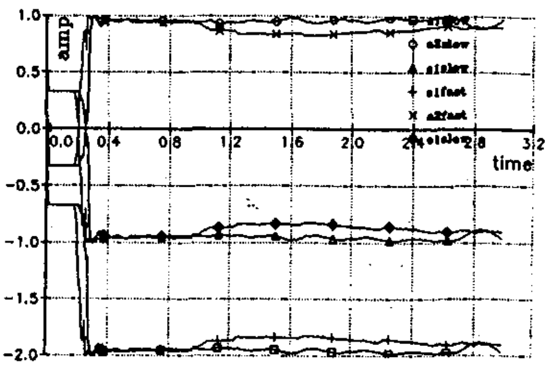

Fig 10: Sample VARMAX convergence