Chapter 3

An LQG approach to self-tuning control with applications to robotics

S. A. Carr, G. Anderson, M. J. Grimble and J. Ringwood

1. INTRODUCTION

Self-tuning control has been recognised as an effective approach for mechanical manipulator control design due to its ability to cope with the presence of nonlinearities and uncertainties in robot dynamic models. The vast majority of existing self-tuning controllers are based on a linear plant description, the fact that most industrial processes are nonlinear is taken into account by regarding the plant as a sequence of pseudolinear descriptions. Therefore, as the plant operating point changes, the nonlinear plant dynamics are reflected as time varying parameters in the linear plant description. The applicability of this approach has been demonstrated by Koivo and Guo [1] where manipulator joint angular position is the controlled variable. This work is extended in Koivo et al [2] to the case in which the manipulator is controlled directly in the cartesian coordinate system. It is found that convergence of the parameter estimates may not be achieved during the finite time over which the motion takes place. Therefore this approach is particularly suited to repetitive tasks where the last estimates from the previous run can be used as the initial estimates. Lelic and Wellstead [3] have successfully applied generalised pole placement to the control of a 5 axis electrically actuated robot manipulator.

This paper considers control of the joint angular position of a Puma 500 robot arm using a Linear Quadratic Gaussian self-tuning controller. LQG based controllers are widely used since they offer a guarantee of stability (when the plant is known) for open loop unstable and nonminimum phase plants for all values of the cost function parameters. In addition LQG controllers are extremely flexible in the control objectives which may be achieved by appropriate choice of these parameters (for example, integral action for offset removal may easily be introduced via the control weighting function). An ARMAX system model (Section 3) is assumed in the optimal controller design. It can be shown that if the continuous time, nonlinear robot model is linearised and discretised at various operating points, the resulting simplified models are indeed of ARMAX form where the $C(z^{-1})$ polynomial (coloured disturbance) models the constant term resulting from gravity loading.

The aim of this paper is to introduce the reader to the field of self-tuning control and to demonstrate its applicability to robot control by the use of some illustrative examples. Section 2 describes the nonlinear, continuous time model of the PUMA 500 which is used for simulation and analysis. For those unfamiliar with the area, Section 3 introduces the concept of self-tuning control, placing particular emphasis on the process model and parameter estimation scheme employed in this paper. In section 4 the optimal LQG controller is presented in both its explicit and implicit forms. Only the single input/single output solution is presented here but the results may be extended to the multivariable case, Grimble [4]. Section 5 provides examples demonstrating the performance of the self-tuning controller for step changes in the reference angle, Section 6 considers the tracking situation where the reference angle varies sinusoidally. In both cases issues such as closed loop stability, control weighting and conditions for good parameter convergence are considered and these are discussed in greater detail in Section 7. Section 8 concludes the paper with a summary of the ideas which have been presented.

2. ROBOT MODEL

The PUMA 500 industrial robot has six degrees of freedom as shown in Figure 1. The waist, shoulder and elbow joints dictate the end effector position while the wrist determines orientation. The following second order dynamic model of the PUMA 500 (Paul [5]) describes joint angular position in terms of joint torque:

$$T_i = \sum_{j=1}^{3} D_{ij}\ddot{q}_j + Ia_i\ddot{q}_i + \sum_{j=1}^{3}\sum_{k=1}^{3} D_{ijk}\dot{q}_i\dot{q}_k + D_i \qquad (1)$$

where

(i) T_i is the torque at joint i.

(ii) q_i is the angular position of joint i. Likewise $\dot{q}_i$ and $\ddot{q}_i$ represent the joint angular velocity and acceleration respectively.

(iii) The D_{ii} terms are multiplied by the angular acceleration of the ith joint and as such they are a measure of its effective inertia, the D_{ij} terms represent coupling inertias between the joints. The terms D_{ii} are often overshadowed by the effect of the reflected motor inertia Ia_i, which is frequently large in comparison.

(iv) Terms of the form D_{iij} and D_{ijk} represent the centripetal and coriolis torques respectively acting on joint i. A centripetal torque is a torque which acts inwards on any body which rotates or moves along a curved path and D_{iij} is the centripetal force at joint i due to velocity at joint j. D_{ijk} represents coriolis forces at joint i due to velocities at joints j and k. Coriolis forces arise in cases of motion relative to a moving axis where the motion of the axis produces a change in the direction of the velocity of the mass.

(v) Finally, D_i represents the gravity loading at joint i.

The coriolis and centripetal torques are important only when the

manipulator is moving at high speed. Both the inertial and gravity terms are important in manipulator control as they affect the servo stability and positioning accuracy.

The robot actuator dynamics are now incorporated into the model. Each joint is driven by a permanent magnet D.C. motor, the dynamics of which are described by:

$$V_i = R_i i_i + L_i \frac{di_i}{dt} + k_i^e \frac{d\omega_i}{dt} \tag{2}$$

$$T_i = k_i^T i_i \tag{3}$$

$$\omega_i = N_i q_i \tag{4}$$

where

R_i = Armature resistance

L_i = Armature inductance

N_i = Gear ratio

i_i = Armature current

k_i^e = Electrical time constant

k_i^T = Torque time constant

ω_i = Armature position.

The following third order model may be derived by substituting for T_i in equation 2 using equations 1, 3 and 4.

$$\begin{aligned} V_i = {} & \frac{L_i}{k_i^T} \sum_j^3 D_{ij}\ddot{q}_j + \frac{L_i}{k_i^T} Ia_i \dddot{q}_i + \frac{L_i}{k_i^T} \sum_j^3 \sum_k^3 D_{ijk}\ddot{q}_j\dot{q}_k \\ & + \frac{L_i}{k_i^T} \sum_j^3 \sum_k^3 D_{ijk}\dot{q}_j\ddot{q}_k + \frac{R_i}{k_i^T} \sum_j^3 D_{ij}\ddot{q}_j + \frac{R_i}{k_i^T} Ia_i\ddot{q}_i + \frac{L_i}{k_i^T} \sum_j^3 \dot{D}_{ij}\ddot{q}_j \\ & + \frac{L_i}{k_i^T} \sum_j^3 \sum_k^3 \dot{D}_{ijk}\dot{q}_k\dot{q}_j + \frac{R_i}{k_i^T} \sum_j^3 \sum_k^3 D_{ijk}\dot{q}_j\dot{q}_k \\ & + k_i^e N_i \dot{q}_i + \frac{L_i}{k_i^T} \dot{D}_i + \frac{R_i}{k_i^T} D_i \end{aligned} \tag{5}$$

This equation may be written in the general nonlinear state space form as

$$\dot{x} = f(x) + g(x).u \tag{6}$$

having defined the state vector for the first three joints as

$$x = (x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9)^T$$

where

$$x_1 = q_1 \;;\; x_2 = q_2 \;;\; x_3 = q_3$$

$$x_4 = \dot{q}_1 \;;\; x_5 = \dot{q}_2 \;;\; x_6 = \dot{q}_3$$

$$x_7 = \ddot{q}_1 \;;\; x_8 = \ddot{q}_2 \;;\; x_9 = \ddot{q}_3$$

and u is the system input.

Using a classical fourth order Runge Kutta numerical integration of equation 6 a solution for the state vector may be found once the actuator input voltage is specified. For simulation of the control scheme the motor voltage is calculated from joint position error using an LQG criterion.

3. SELF-TUNING

An adaptive control system is one in which the controller is automatically adjusted to compensate for unanticipated changes in the process or environment. Adaptive control systems therefore provide a systematic approach for dealing with nonlinearities such as those encountered in robotic systems.

Self-tuners, which estimate model parameters on-line and adjust controller settings accordingly fall into this category.

A typical self-tuning scheme is shown in Fig. 2. It consists of four main blocks.

(i) The Process/Process Model

(ii) The Parameter Estimator

(iii) Controller Design

(iv) The Controller

Each block will now be considered in greater detail.

3.1 The Process Model

The ARMAX (AutoRegressive Moving Average with eXogeneous inputs) linear process model is used in this paper to approximate the robot dynamics at the operating point for controller design purposes. This model describes the plant output in terms of a linear combination of previous plant outputs and delayed control inputs and a coloured noise disturbance (Eqn. 7)

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - \ldots - a_n y(t-n)$$
$$+b_o u(t-k) + b_1 u(t-k-1) + \ldots + b_m u(t-k-m)$$
$$+\zeta(t) + c_1 \zeta(t-1) + \ldots + c_\ell \zeta(t-\ell) \qquad (7)$$

where

$y(t)$ = process output at time t

$u(t)$ = control signal at time t

$z(t)$ = white Gaussian noise disturbance

k = process delay.

This can be written more compactly as

$$A(z^{-1})y(t) = B(z^{-1})u(t) + C(z^{-1})\zeta(t)$$

where

$$A(z^{-1}) = 1 + a_1 z^{-1} + \ldots + a_n z^{-n}$$
$$B(z^{-1}) = z^{-k}(b_o + b_1 z^{-1} + \ldots + b_m z^{-m})$$
$$C(z^{-1}) = 1 + c_1 z^{-1} + \ldots + c_\ell z^{-\ell} \qquad (8)$$

and z^{-1} is the backward shift operator.

3.2 The Parameter Estimator

Identification is the process of constructing a mathematical model (in this case an ARMAX structure is used) of a system from observations and prior knowledge. Identification and parameter estimation have found applications in areas as diverse as engineering, science, economics, medicine, ecology and agriculture.

For the purposes of parameter estimation the ARMAX system description is rewritten in the form

$$y(t) = \Phi^T(t-1).\theta + \varepsilon(t) \qquad (9)$$

where

$$\Phi^T(t-1) = (-y(t-1) - \ldots - y(t-n) : u(t-k) \ldots u(t-k-m)$$
$$\zeta(t-1) \ldots \zeta(t-\ell))$$
$$\theta^T = (-a_1 \ldots -a_n ; b_o \ldots b_m : c_1 \ldots c_\ell)$$

and

$$\varepsilon(t) = \xi(t)$$

$\Phi^T(t-1)$ is the data vector which contains information about the process up to and including sample time (t-1). θ is the system parameter vector which is to be estimated and $\varepsilon(t)$ represents the estimation error which is assumed to be statistically independant of the inputs and outputs.

A recursive parameter estimation algorithm will, for a specified system output $y(t)$ and data vector $\Phi(t-1)$, find the estimates, θ, of the unknown parameters which minimise a specified loss function $V(\theta)$.

For a quadratic criterion (where the objective is to minimise the squared difference between actual and estimated plant outputs):

$$V(\theta) = 1/2.\ E\left[y(t) - \Phi^T(t-1).\theta\right]^2$$

where $E[.]$ denotes the expectation operator.

Minimisation of the loss function, $V(\theta)$, in conjunction with an ARMAX plant description yields the following Extended Least Squares algorithm.

3.2.1 Extended Least Squares Algorithm (ELS).

$$\theta(t) = \theta(t-1) + L(t)\left[y(t) - \theta(t-1).\Phi(t-1)\right] \qquad (10)$$

$$L(t) = \frac{P(t-1)\phi(t-1)}{1 + \phi^T(t-1).P(t-1).\phi(t-1)} \qquad (11)$$

$$P(t) = P(t-1) - \frac{P(t-1).\phi(t1)\phi^T(t-1)P(t-1)}{1 + \phi^T(t-1).P(t-1)\phi(t-1)} \qquad (12)$$

where

$P(t)$ = Covariance matrix

$L(t)$ = Gain vector

$\phi(t)$ = The gradient vector

$$= \frac{dy(t,\theta)^T}{d\theta} = \frac{d\Phi^T\theta}{d\theta}$$

$= \Phi^T(t-1)$ under the assumption that $\Phi^T(t-1)$ is independant of θ.

3.3 Controller Design

Once the model parameters have been estimated they are used to design the controller. This can be done in two ways.

(i) **Explicit Algorithms**
The estimated process parameters are manipulated mathematically to produce the updated set of controller parameters.

(ii) **Implicit Algorithms**
The process model is parameterised in terms of the controller parameters in such a way as to update them directly at the identification stage.

The implicit approach gives some advantages with respect to computational speed while the explicit algorithms are more flexible in the different control objectives which can be achieved.

3.4 The Controller

The controller used is a fixed structure controller, the parameters of which are varied by the design stage. The choice of control law depends upon many factors, including,

(i) Is the process open loop unstable?

(ii) Is it non-minimum phase?

(ii) Is there a significant dead time?

(iv) Is the measured output corrupted by noise?

(v) Are there constraints on available computational power?

(vi) Is excessive actuator movement undesirable?

Therefore the choice of control law is highly application dependent. Ultimately a trade-off must be made between improved quality of control and controller simplicity.

For this paper, the control law under consideration is based on an LQG criterion and is discussed in the next section.

4. LINEAR QUADRATIC GAUSSIAN CONTROL

The fundamental difference between LQG control laws and those which are based on a minimum variance type solution is that LQG controllers are based on the minimisation of a cost function containing an unconditional expectation operator while the expectation is conditional for minimum variance solutions. Therefore minimum variance control laws are suboptimal with respect to LQG control laws.

The closed loop discrete time system decription used (Fig. 3) is given by the following set of equations:

(i) **System output equations**

$$y(t) = A^{-1}(z^{-1})(B(z^{-1})u(t) + C(z^{-1})\xi(t))$$

This is the ARMAX plant model which is described in Section 3.1.

(ii) **Observation process**

$$z_o(t) = y(t) + v(t)$$

where v(t) is an output disturbance of variance R.

(iii) **Controller input**

$$e_1(t) = r(t) - z_o(t)$$

The controller is fed by the difference between the desired output and observed output.

(iv) Reference Generation Process

$$r(t) = A^{-1}(z^{-1})E(z^{-1})\omega(t)$$

Provision is made for a stochastic reference which is generated by the above subsystem. A deterministic set point may also be included.

(v) Tracking error

$$e(t) = r(t) - y(t).$$

Tracking error is the difference between the desired output and the uncorrupted plant output.

In future discussions the arguments of the polynomials and variables will be omitted for notational simpliciity.

It is assumed that none of these subsystems includes unstable hidden modes (thus unstable and uncontrollable modes are not present).

The following unconditional cost function is to be minimised.

$$J = E[Q_1 e^2(t) + R_1 u^2(t)] \qquad (13)$$

where

e(t) = Error signal (difference between reference and output)

u(t) = Control signal

Q_1 = Error weighting

R_1 = Control weighting

The signals r, z and ω are assumed to be stationary, zero mean sequences of uncorrelated random variables which have variances given by R, Q_2 and Q_3 respectively. The generalised spectral factors satisfy:

$$Y\bar{Y} = (EQ_3\bar{E} + CQ_2\bar{C} + AR\bar{A})/(A\bar{A}) \qquad (14)$$

$$\bar{Y}_1 Y_1 = (\bar{B}Q_1 B + \bar{A}R_1 A)/(\bar{A}A) \qquad (15)$$

where

$$\bar{X}(z^{-1}) = X(z)$$

is called the adjoint of $X(z^{-1})$.

The strictly Hurwitz (stable inverse) polynomials in z^{-1}, D and D_1 may be defined as

$$Y\bar{Y} = D\bar{D}/(A\bar{A}) \Rightarrow Y = D/A \tag{16}$$

$$\bar{Y}_1 Y_1 = \bar{D}_1 D_1/(\bar{A}A) \Rightarrow Y_1 = D_1/A \tag{17}$$

4.1 Explicit LQG

The solution for the optimal explicit LQG controller may now be presented, a more detailed derivation may be found in Grimble [6]. The optimal controller transfer function is:

$$C_o = G_o/H_o \tag{18}$$

where G_o and H_o are polynomials in z^{-1}. The following coupled diophantine equations in terms of the unknown polynomials G, H and F, provide the unique particular solution G_o, H_o with minimal degree with respect to F:

$$\bar{D}_1 z^{-g} G + FA = \bar{B} z^{-g} Q_1 D \tag{19}$$

$$\bar{D}_1 z^{-g} H - FB = \bar{A} z^{-g} R_1 D \tag{20}$$

where $g \triangleq \max(n_{d1}, n_b, n_a)$. These equations can be combined to obtain the implied equation:

$$AH + BG = D_1 D \tag{21}$$

The solution of the implied diophantine equation (21) with $n_h < n_b$ and $n_g < n_a$ and A,B coprime is unique, as may be verified from Theorem 4, Jezek [7].

It can be seen that equation (21) is in fact the closed loop characteristic polynomial and as DD_1 can be shown to be strictly Hurwitz the stability of the system is guaranteed, regardless of the control weighting even for open loop unstable or nonminimum phase plants.

4.2 Implicit LQG

Implicit self-tuning controllers sometimes have advantages for implementation due to their direct means of calculating the controller parameters. An implicit LQG control law may be derived assuming that a unique solution to (21) exists (A and B are coprime). The following innovations plant description is used:

$$A(z^{-1})e_1 = D(z^{-1})\varepsilon - B(z^{-1})u \tag{22}$$

where $A(z^{-1})$, $D(z^{-1})$, e and u are as previously defined and ε is a unit variance white noise signal. This closed loop model can be shown to be equivalent to that of Fig. 3. Combining equations (21) and (22) yields

$$Ae_1 = (AH_o + BG_o)\varepsilon/D_1 - Bu$$

After some manipulation this can be shown to be equivalent to

$$\phi(t) = H_o\varepsilon + B/D(G_o e_1 - H_o u) \tag{23}$$

where $\phi(t) = D_1 e_1$

This represents the desired implicit model from which G_o and H_o may be estimated. However if $n_{ho} > k-1$ then the residual $H_o\varepsilon(t)$ is correlated with the regressors $e_1(t-k)$, $e_1(t-k-1)$,..$u(t-k)$,$u(t-k-1)$ in (23). Thus write $H_o = H_{o1} + H_{o2}$ where H_{o1} includes all the terms with powers of z^{-1} up to $z^{-(k-1)}$ to obtain the least squares predictor:

$$\hat{\phi}(t|t-k) = H_{o2}\varepsilon(t) + B/D(G_o e_1(t) - H_o u(t)) \tag{24}$$

and the prediction error

$$\tilde{\phi}(t|t-k) = H_{o1}\varepsilon(t) \tag{25}$$

where $\hat{\phi}(t|t-k)$ denotes that the value of $\hat{\phi}$ at time t is based only on data up to time t-k. As the optimal control signal $u^o(t)$ is chosen to set the final term in (24) to zero, the prediction equation can be simplified as:

$$\hat{\phi}(t|t-k) = H_{o2}\varepsilon(t) + B(G_o e_1(t) - H_o u(t)) \tag{26}$$

using the argument employed by Clarke and Gawthrop [8].

If the polynomials A,B and D and the innovations signal ε are estimated using ELS parameter estimation and equation (22) and the stable spectral factor D_1 is calculated using (17) then by defining $e_b = Be_1$ and $u_b = Bu$ the controller polynomials G_o and H_o can be identified using

$$\hat{\phi}(t|t-k) = G_o e_b - H_o u_b + H_{o2}\varepsilon$$

and Extended Least Squares Identification

5. SET POINT CONTROL

This section evaluates the regulatory performance of the self-tuning controller. The response of the robot arm to the motor actuation voltage (controller output) is simulated using the third order nonlinear state space model of equation 6. The Runge-Kutta integration time used is 1 millisecond. The controller sampling time was chosen, based on open loop step response curves, to be 0.1 seconds. At each control sampling interval the data vector for parameter estimation (Section 3.2) is updated, the coefficients of the A,B and C polynomials are re-evaluated using the new data vector, the explicit LQG controller is designed based on the updated estimates and the new actuation signal is applied to the robot model. All simulations consider control of the angular position of joint three, with joints one and two locked at zero radians. Linearisation and discretisation of the robot model of equation 5 at various operating points shows that the linearised system is best described by an ARMAX model where the $A(z^{-1})$ and $B(z^{-1})$ polynomials are third order and the $C(z^{-1})$ polynomial which

characterises the gravity loading term is second order. The polynomials $G(z^{-1})$, $F(z^{-1})$ and $H(z^{-1})$ are chosen to be second order to balance the powers of z^{-1} on both sides of the diophantine equations (19) and (20).

Example 1

Figure 4a shows the angular position of joint three when the reference angle is zero radians (vertical joint). The initial $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ parameter estimates (Figures 4b - d) were chosen based on the final estimates from previous trials. The initial values of the diagonal elements of the covariance matrix, P(t), where chosen to be 1000 to promote rapid convergence since the magnitude of step increments/decrements in the parameter estimates at each iteration is directly dependent on the magnitude of the elements of P(t).

The open loop system transfer function, $B(z^{-1})/A(z^{-1})$, converges to

$$B(z)/A(z) = \frac{0.02(z + 1.025)(z-0.975)}{(z-.6298)(z-1.27)(z-1)}$$

which is both unstable and nonminimum phase. The closed loop transfer function, G(z), is:

$$G(z) = \frac{0.64z(z+1.025)(z-0.975)[(z-0.662)^2+0.1842^2]}{(z+0.97)[(z-0.69)^2+0.272^2](z-0.442)(z-0.974)} \qquad (27)$$

Therefore it can be seen that the LQG self-tuning controller has stabilised the system.

The joint angle is initially disturbed from the reference position while the parameters tune in, reaching a maximum angle of -1.65 radians.

Example 2

The initial output disturbance in the previous example is clearly undesirable. This example shows that a dramatic improvement in performance may be obtained by using a fixed LQG controller during the initial tuning-in period (Figure 5). The fixed controller is designed based on the estimates of the polynomials $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ of Figure 4. The maximum deviation of the output from the reference is reduced to approximately 0.1 radians, less than 1% of that in the previous example.

Example 3

Figure 6 presents the closed loop response of the system for different step inputs. The converged values of the $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ polynomials at various operating points, including those of figure 6, are given in Table 1. Analysis of the $A(z^{-1})$ and $B(z^{-1})$ polynomials shows that at each operating point the open loop system is unstable and nonminimum phase, whilst the closed loop system is stable.

Each response is characterised by a time constant of approximately 1.5 seconds, an initial overshoot of between 10% and 20% of the set point reference and a settling time of approximately 3 seconds. In each case there is a small steady state offset (less than 0.01 radians) due to the fact that the nonlinear system is represented, for controller design, by a linear model.

6. PATH TRACKING

This example considers the more realistic situation where the required joint angular position is not stationary but varies with time.

Example 4

For this simulation a sinusoidal reference angle with a period of 10 seconds, a peak to peak amplitude of 1.6 radians and a mean value of 1.0 radians was applied to the system. Figure 7a shows that after an initial overshoot during the tuning in period (of approximately 3 seconds) the actual angle of joint three tracks the specified angle closely. As the link moves through its nonlinear region of operation the parameter estimator revises, at every sample instant, the linear plant description which is used for controller design. Figures 7b to 7d show the variation with time of the $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ parameters respectively. As expected from analysis of the robot model, the A parameters remain constant during the simulation. There is a small variation in the B parameters, particularly in the b_2 coefficient. However, as expected, the most noticeable variation occurs in the coefficients of the $C(z^{-1})$ polynomial, which models the gravitational disturbance term. This is because, under the operating conditions of this example the nonlinear gravitational torque predominates.

7. DISCUSSION

The following factors have been considered in the implementation of the self-tuning control scheme presented in this paper.

7.1 Parameter Tracking

The covariance matrix, P(t), is a positive definite measure of the estimation error - therefore the magnitude of it elements tends to decrease with time. The magnitude of the step change in θ, the parameter vector, at each iteration is directly dependent on the magnitude of the elements of P(t). To maintain the sensitivity of the algorithm and allow for parameter tracking some modifications must be made to the algorithm of equations (10) to (12) in order that the elements of P(t) are prevented from becoming too small. One technique which is commonly used is to include an exponential weighting factor in the performance index as follows:

$$V(\theta) = \sum_{i=1}^{t} \lambda^{t-i}[y(i)-\Phi^T(i-1).\theta(i)]^2$$

where

$$0.0 < \lambda \leq 1.0 \qquad (28)$$

for $\lambda = 1.0$ all data are weighted equally. For $0 < \lambda < 1.0$ more weight is placed on recent measurements than on older measurements. This revised performance index results in the following least squares algorithm.

$$\theta(t) = \theta(t-1) + L(t)[y(t) - \theta^T(t-1)\Phi(t)] \qquad (29)$$

$$L(t) = \frac{P(t-1).\phi(t)}{\lambda + \phi^T(t).P(t-1).\phi(t)} \qquad (30)$$

$$P(t) = 1/\lambda\left[P(t-1) - \frac{P(t-1).\phi(t).\phi^T(t).P(t-1)}{\lambda + \phi^T(t).P(t-1).\phi(t)}\right] \qquad (31)$$

7.2 Initial Parameter Estimates

Estimates of the data vector, θ, must be supplied to initiate the algorithm of equations (29) to (31). If no prior knowledge of the system is available then these are chosen arbitrarily. If the elements of the covariance matrix, P(t), are large (of the order of 10^5) and the forgetting factor, λ, is less than 1.0 (a value of 0.95 gives good results) then the elements of the gain vector, L(t), are large and rapid convergence is achieved. In order to ensure excitation of all of the modes of the system a stochastic, persistently exciting reference (Norton [9]) may be applied to the controller initially. It is advisable to use a fixed controller in parallel with process identification during the tuning in period, after which control is transferred to the self-tuner. After the first trial improved parameter estimates are available. These should be used as the initial estimates for the next task. For set point control the initial elements of the covariance matrix should be reduced (to of the order of 10) and the forgetting factor should be increased (a value of 0.98 was used in examples 1 to 3) to decrease the sensitivity of the algorithm.

However, if, as in the vast majority of robotics applications, parameter tracking is required (example 4), these values should be approximately 100 and 0.97 respectively to maintain sensitivity to parameter changes while at the same time rejecting measurement disturbances.

7.3 Offset Removal

From classical control theory it can be shown that offset removal may be achieved, for linear systems, by

(i) increasing controller gain

(ii) including integral action in the controller.

The open loop process transfer function of example 1 shows that the numerator is multiplied by a factor of 0.02 relative to the denominator. Therefore a fixed gain of 50 was included in the controller transfer function. This had the effect of signficantly decreasing the steady state offset, since it is known that steady state offset is inversely proportional to controller gain for a step reference input. However, if the gain is increased further, oscillations are introduced in the output as the closed loop poles move along the root locus towards the zeros near the unit circle of (27).

Integral action removes steady state offset for linear systems. However, the system under consideration is highly nonlinear and the inclusion of an integrator is equivalent to placing a controller pole on the unit circle. It was found that this results in an oscillatory response, even when the controller gain is reduced.

The steady state offset of examples 1 to 3 is a result of representing a highly nonlinear system by an approximate linear model. However the magnitude of this offset may be reduced, using a high gain controller, to within tolerable limits for many applications. The use of a virtual reference is also suggested.

7.4 Control Weighting

The control weighting parameter, R_1, determines the relative importance which is to be placed on the penalisation of the control signal by the cost function. It follows that a high value of R_1 leads to less control variation and a more highly 'damped' output while a low relative value of R_1 leads to a lower variance of tracking error and smaller offset. For this reason the ratio of $Q_1 : R_1$ was chosen to be quite high at 1:0.01. It is also possible to use dynamic (frequency dependent) weights in the solution of the optimal controller to allow the error and control signals to be penalised differently in different frequency ranges.

8. CONCLUSIONS

It has been shown with the aid of some illustrative examples that LQG self-tuning control can successfully be used to stabilise and control a mechanical manipulator. For simulation purposes the robot dynamics are described by a set of third order, cross-coupled, nonlinear equations. The self-tuning controller represents these by a linear ARMAX model which is updated at every sample interval. The single joint control examples presented in this paper are simple but demonstrative. They may easily be extended to consider multiple joint control and end effector position control in cartesian coordinates.

9. REFERENCES

1. Koivo, A.J. and Guo, T., 1983, IEEE Trans. Aut. Control, Vol AC-28. 162-171.

2. Koivo, A.J., Kunkel, R., and Guo, T., 1983, 'Adaptive Manipulator Control in Cartesian Coordinate System', Proceeding of the 7th International Computer Software and Applications Conference of the IEEE Computer Society, Chicago, Illinois.

3. Lelic, M.A., and Wellstead, P.E., 1986, 'A Generalised Pole Placement Self-Tuning Controller - An Application to Robot Manipulator Control', Report No. 658, Control Systems Centre, UMIST.

4. Grimble, M.J., 1986, Int. J. Systems Sci., Vol. 17, No. 4, 543-557.

5. Paul, R.P., 1981, 'Robot Manipulators : Mathematics, Programming and Control', The MIT Press, Cambridge, Massachusetts.

6. Grimble, M.J., 1984, Automatica, Vol. 20, 5, 661-669.

7. Jezek, J., 1982, Kybernetika, Vol. 18, 505-516.

8. Clarke, D.W., and Gawthrop, P.J., 1975, Proc. IEE, Vol. 122, 929-934.

9. Norton, J.P., 1986, 'An Introduction to Identification', Academic Press, London, England.

Table 1 The estimated plant parameters at various operating points

Angle (Radians)	a_1	a_2	a_3	b_o	b_1	b_2	c_1	c_2
-0.50	-2.8	2.6	-0.8	0.0205	0.003	-0.0185	0.95	0.15
0.00	-2.9	2.7	-0.9	0.0200	0.001	-0.0200	0.94	0.12
0.30	-3.0	2.6	-1.0	0.0200	0.002	-0.0190	0.61	0.39
1.57	-2.9	2.9	-0.9	0.0200	0.001	-0.0190	0.40	0.47
2.00	-3.0	2.9	-1.0	0.0235	0.002	-0.0210	0.50	0.56
3.00	-2.8	2.4	-0.8	0.0280	0.001	-0.0180	0.20	0.40

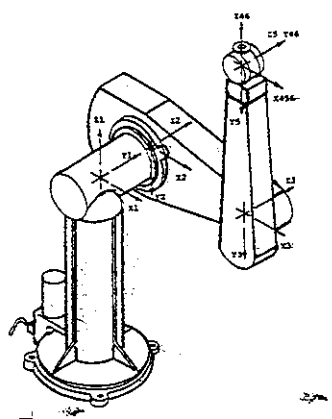


Fig. 1 The PUMA 500 manipulator.

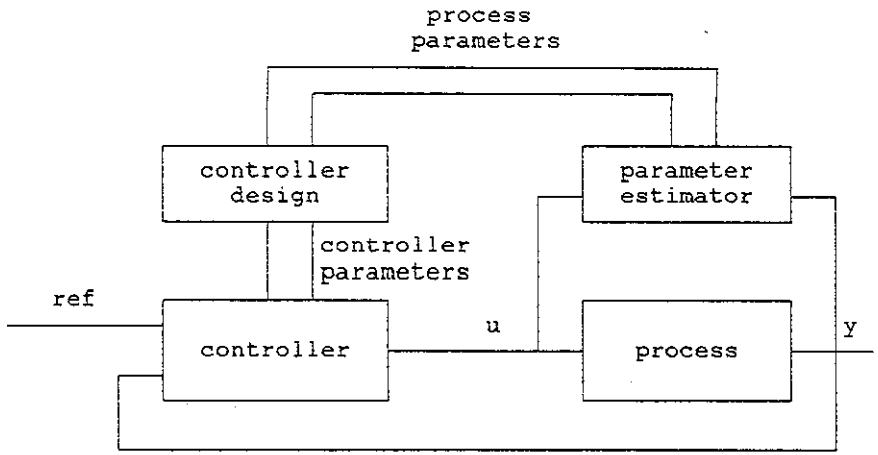


Fig.2 A typical self-tuning control scheme

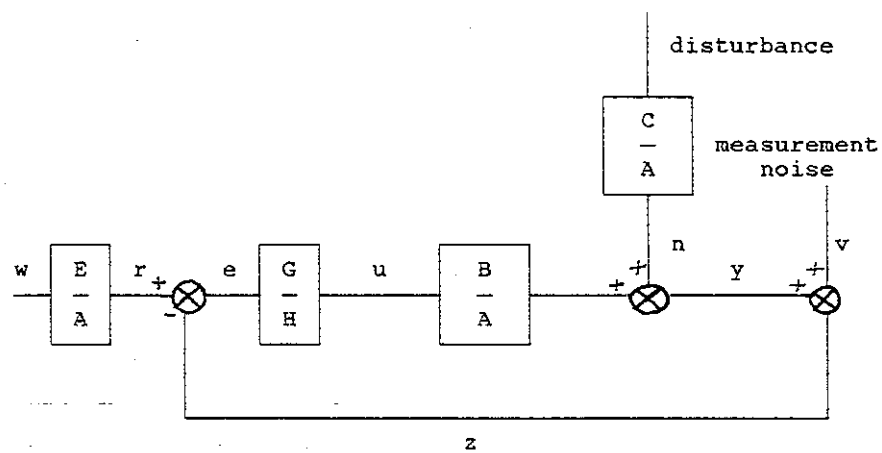
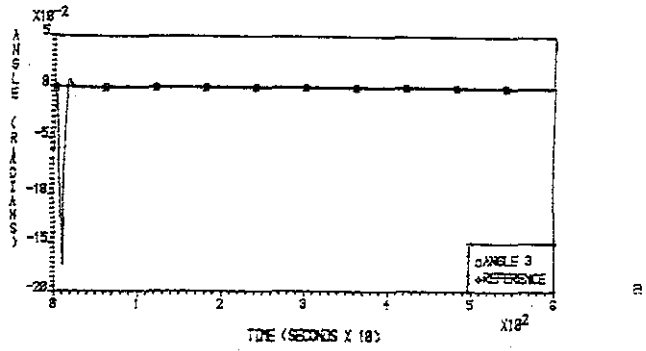
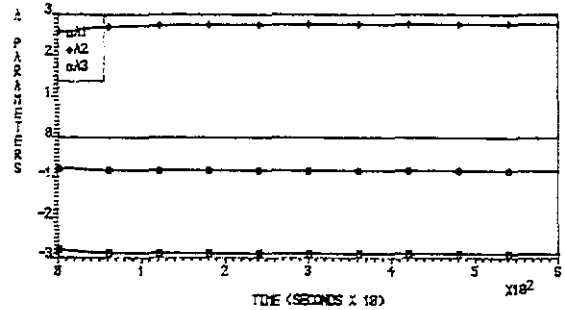


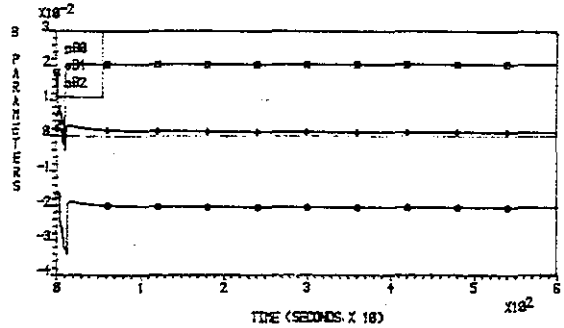
Fig.3 The closed loop system



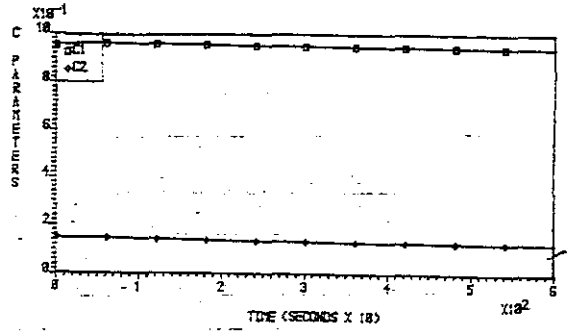
a



b



c



d

Fig. 4 a) The process output and reference b)The A polynomial parameters c)The B polynomial parameters d)The C polynomial parameters.

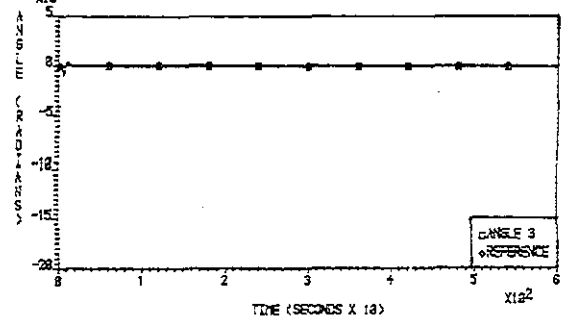
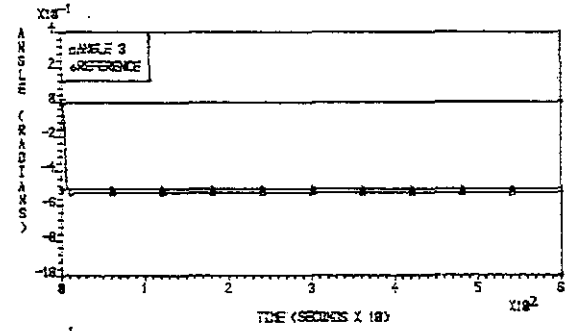
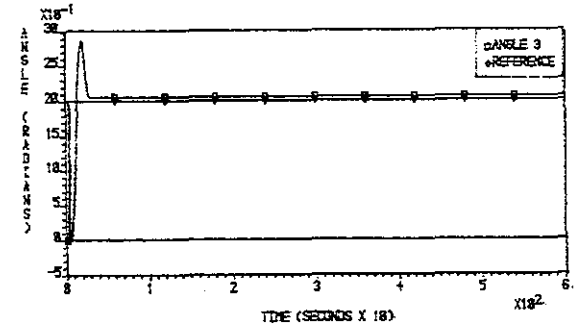


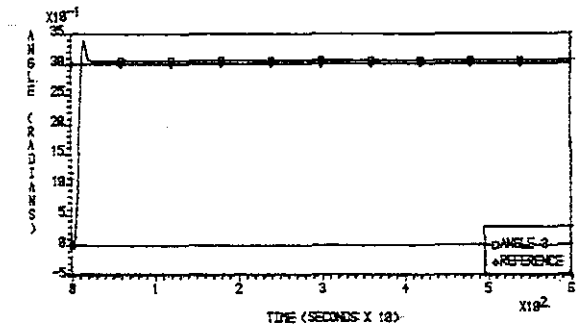
Fig. 5 The process output and reference using a fixed controller for the first 10 samples.



a

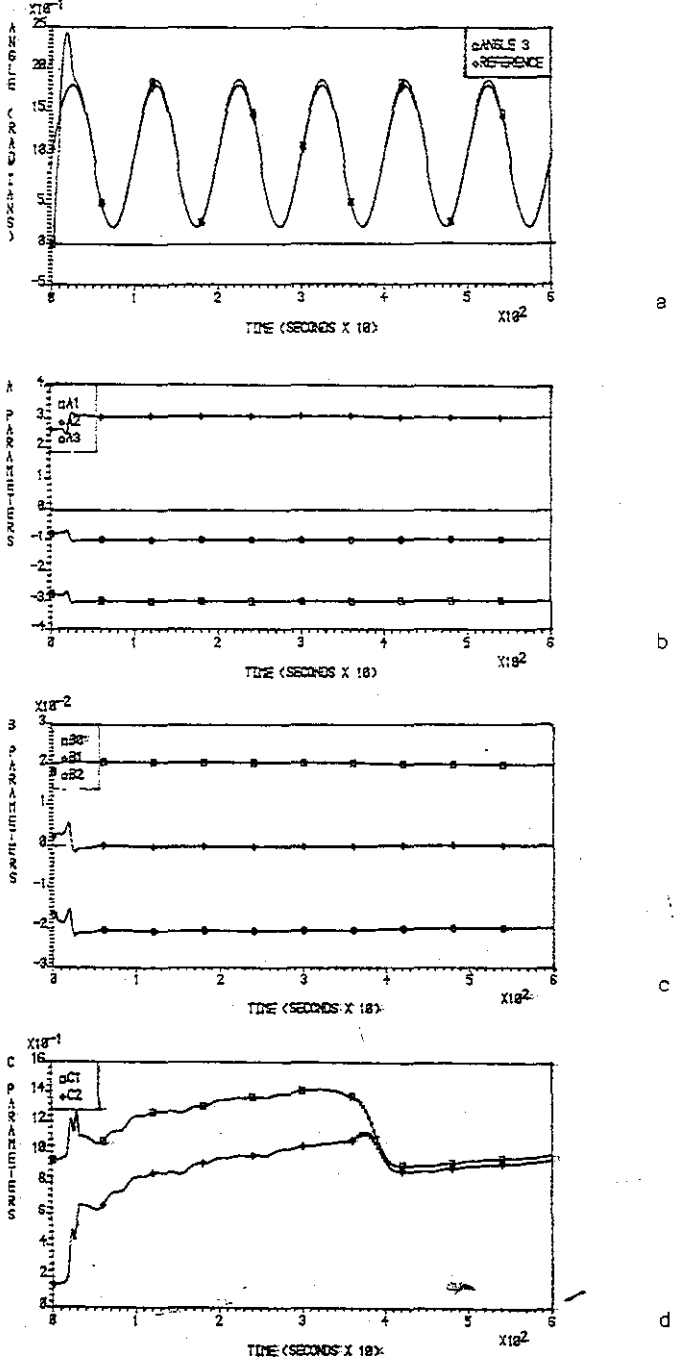


b



c

Fig. 6 The process output for a) -0.5 radian set point b) 2.0 radian set point c) 3.0 radian set point.



a

b

c

d

Fig. 7 a)The process output and reference b) The A polynomial parameters c)The B polynomial parameters d) The C polynomial parameters.

Chapter 4

Adaptive control algorithms for intelligent robot manipulators

M. Farsi, K. J. Zachariah and K. Warwick

1. INTRODUCTION

The aim of this paper is to describe adaptive controllers suitable for use in the control of robotic manipulators. Upper Diagonal Factorization (UDF) or a simplified parameter estimator is employed within the self-tuning algorithms to estimate the parameters contained in CARMA models of the joints. The simplified estimator used reduces the computational effort considerably.

The robot under investigation is made by Kuka (Fig. 1) and consists of six 'links' attached serially to each other at revolute joints. The coordinate frames attached to the manipulator are also indicated in the figure. Joint information is obtained via an optical encoder measuring relative angular displacement and a tachometer measuring relative angular speed between adjacent links.

The manipulator is intended to interact with objects in the three dimensional space surrounding it, therefore, it is necessary to obtain the position and orientation of the end effector in cartesian space. This information, in this case, in the form of binary data is made available to a transformation module of the robot via the vision sensor machine.

2. DYNAMIC MODEL OF THE KUKA ROBOT

In order to be able to test a variety of control schemes and also the behaviour of the controlled robot under different load and speed conditions it is necessary to operate on a comprehensive dynamic model. The dynamic model can be derived in a variety of ways such as the Newton-Euler or the Lagrange-Euler methods, Ranky and Ho (1). The approach used here is based on the use of Lagrangian mechanics, where expressions for the kinetic and potential energy of the robot structure are used to obtain relationship between the input (torque) and output variables. The equations for the first three links of the Kuka robot are as follows: