

Adaptive lattice estimation and control of a manipulator with one flexible forearm

Y.-P. Yang
K.-M. Huang

Indexing terms: Adaptive control, Lattice filter, Flexible arm, Vector-channel, Finite element modelling

Abstract: An application of recursive covariance lattice algorithms to the adaptive estimation and control of a manipulator with one flexible link is presented. These algorithms are a set of pure order recursive lattice equations, which can in principle identify the effective order and the corresponding parameters of an ARMA prediction model of the manipulator. The reduced-order prediction model represents the significant dynamics of the plant, and is used to generate control sequences by minimising a weighted performance index. In the simulations, the manipulator is modelled by the finite element method and Lagrange's equations. The performance and robustness of the lattice filter and the adaptive controller is demonstrated by numerical results.

1 Introduction

The design and control of lightweight robotic manipulators is currently attracting the attention of many researchers. A substantial amount of research effort has been devoted to the adaptive control of manipulators with flexible links [1-7]. One of the advantages of an adaptive control scheme is its capability of real time estimation of plant uncertainties, while classical control schemes are designed with *a priori* information on plant dynamics. The design of a robust adaptive controller for a manipulator presents problems of system uncertainties, unmodelled dynamics, nonlinearities and time-varying disturbances.

In the past decade, Nelson, Miltra and Boie [2, 3] described a method for controlling a lightweight flexible link, using an on-line estimator to provide periodic updates of load mass estimates to a parametric optimal controller whose gains were adjusted to follow a desired response. Chung and Leininger [4] applied the pole placement self-tuning method to a six-degree-of-freedom JPL Stanford arm with the third link flexible. Yuh [5] used a discrete-time model reference adaptive controller to a single flexible link. Rovner and Cannon [7] applied an adaptive control algorithm based on the self-tuning regulator concept for a very flexible one-link robotic

manipulator. In Reference 1 we developed a model reference adaptive control scheme on a two-link manipulator with a flexible forearm. Simulations showed that the adaptive controller produced fast, accurate manipulator response for a wide range of payloads and stiffness of the flexible link.

Recently, very little literature has dealt with the stochastic adaptive control of robotic manipulators with flexible links. Oakley and Cannon [10] proposed a non-linear controller incorporating state feedback and a constant gain extended Kalman filter driven by end-point measurements of a two-link manipulator with a very flexible forearm. We also advanced a self-tuning stochastic adaptive controller for a two-link manipulator with one flexible link [11]. The process and measurement noises with unknown statistics were modelled as innovative processes with unknown parameters in a reduced-order ARMAX prediction model, on which a one-step-ahead minimum variance controller was based.

Most of the literature mentioned the order updating of the prediction model in the adaptive control of a flexible structure. Sundararajan and Montgomery [9, 12] were first to apply lattice filters to identification and adaptive modal control of flexible structures. Lattice algorithms were originally proposed in the signal processing area for parameters estimation and system identification; later, the exact recursive least squares method was developed to compute the lattice coefficients in real time; see Reference 13 for a survey. This class of exact fast algorithms has many advantages over conventional least-squares algorithms. Besides its recursive property in both time and order, it has an orthogonalising property, good tracking performance, rapid start-up capability, fast convergence and easy implementation owing to a high repetitive structure. The extension of the original AR lattice form to the ARMA models by the embedding technique gives the lattice filter a very promising perspective in the control framework. Friedlander [14], Ölçer and Morf [15] advanced new types of algorithms for adaptive control based on recursive lattice forms, yet considering minimum phase plants only.

This paper presents an application of the recursive covariance lattice filter [19] to the adaptive control of a robotic manipulator with one rigid link and one flexible link in the gravitational field. The adaptive control scheme is indirect; i.e. the control signal at each sampling instant is based on a prediction model of the plant. This prediction model is in a linear ARMA form with time-varying parameters. The recursive covariance lattice filter is applied to estimate the effective order of the prediction

Paper 8478D (C8, C13, C15), first received 5th October 1990 and in revised form 2nd September 1991

The authors are with the Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan 10764, ROC

model to reflect most significant dynamics (or modes) of the plant, as well as to estimate the corresponding parameters. The most attractive feature of this filter is that the algorithm structure is a pure order recursive lattice of a set of generalised residual energy matrices. Therefore, the round-off errors will not propagate in the time direction as the conventional lattice filter does through time update of a cross-residual energy matrix. Thus the numerical accuracy and stability problems [16] caused by round-off errors are absent. The other important aspect of the proposed adaptive estimation process is the vector-channel embedding technique [17] to treat the moving average part of the ARMA model as an additional output which we include in a vector with AR terms. As opposed to conventional lattices, which treat each scalar output as a single channel, the vector-channel lattice treats several outputs as components of one vector measurement channel. Since the plant modelled in this paper is very nonlinear, sensors at various links may not observe the same dynamics. These motivate a separation of two main estimation channels, each for one robot arm. Each main estimation channel consists of several sub-channels, and each subchannel contains a number of scalar measurements. Our embedding method is elaborated in Section 3.2.

The adaptive controller is designed to minimise a weighted measure of control effort. The error between the predicted output of the system and the output of a control model is obtained from an optimal linear regulator calculation. This error is made to decay according to an error dynamics model. Owing to the noncollocated sensors and actuators on the manipulator, the plant is near non-minimum phase [1], therefore unrobust and sensitive to the perturbation of control parameters; for example the weighting matrices of the cost function. We found that a PD controller in an inner loop can augment the plant to produce stable discrete-time transmission and channel zeros, thereby improving robustness. The most robust controller consists of an indirect adaptive controller and an inner PD loop. This is verified by simulation results which are shown in Section 5.

2 Manipulator-plus-actuator model

For simulating the response of a robotic manipulator in a digital computer, system equations of motion are derived. The two-link manipulator to be controlled is displayed schematically in Fig. 1. Two actuators centred at o_1 and o_2 are modelled as rigid discs with the same mass M_1 . The first link is uniform, rigid and clamped to the first disc; the second link is a uniform Euler-Bernoulli beam, clamped to the second disc. The end-effector and the payload are modelled as a point mass M_2 at the other end of the second link. A control torque u_1 acts on the first disc, and a control torque u_2 acts between the second link and the second disc; both torques are generated by DC servomotors whose dynamics are coupled with those of robot structures.

The manipulator is constrained to move in a vertical plane of the gravity field. The cross-section of the flexible link is rectangular, long enough in the out-of-plane direction that the flexible link is relatively rigid in the out-of-plane motion, which is then negligible. We do not model the axial vibration of the flexible link, but we include the effect of the inertia axial load on bending stiffness. The flexible link is modelled by the finite element method, being approximated with n -elements of equal length and

cubic B-splines as interpolation functions. Therefore we have n elastic degrees of freedom, which we take to be the transverse elastic displacements of node 1, 2, ... and n . Node 0 is the end of the end of link 2 attached to the second disc, node n is the end of link 2 to which the

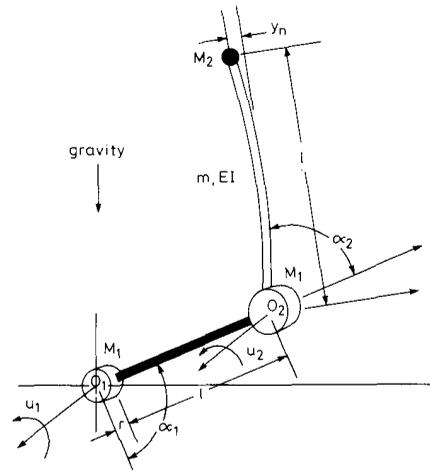


Fig. 1 Two-link manipulator

payload is attached. With the two rigid-body degrees of freedom — angular displacements measured by digital encoders on both actuators, there are $n_s (= n + 2)$ degrees of freedom in the simulation model. The generalised displacement vector in the finite element model is $q = [\alpha_1 \alpha_2 y_1 y_2 \dots y_n]^T \in \mathcal{R}^{n_s}$ where α_1 and α_2 are rigid angles and y_1, y_2, \dots, y_n are the transverse elastic displacements of node 1, 2, ..., n on the second link.

In the finite element model, we include all nonlinearities that would be present if both links were rigid. Therefore, small transverse vibrations of the flexible link are coupled with those nonlinearities. Lagrange's equations for the finite element model without actuator dynamics have been derived simply in Reference 8 as follows:

$$[M(q) + M_a(q)]\ddot{q} + D\dot{q} + [K + K_a(q, \dot{q})]q + N(q, \dot{q}) = T \quad (1)$$

where $M(q)$ is a positive definite symmetric mass matrix, K is the non-negative symmetric stiffness matrix due to bending stiffness of the flexible link, $N(q, \dot{q})$ is a vector containing various gravity and inertial torques and Coulomb frictions, and T is the applied torque from actuators. The matrices $M_a(q)$ and $K_a(q, \dot{q})$ represent the effect of the inertial axial load on the transverse vibration of the second link; $K_a(q, \dot{q})$ is symmetric but $M_a(q)$ is not, the asymmetry results because we do not model axial vibrations. The damping matrix D is 10^{-6} times the part of the mass matrix that corresponds to the flexible link if α_1 and α_2 are held constant; this means that we model small proportional damping for the flexible link.

The electric actuators are modelled as typical armature-controlled DC servomotors, whose inductance of the armature winding is usually small, in the order of tenths of millihenries, and thus negligible. The motor dynamics can then be modelled as a second-order vector

differential equation

$$G\ddot{\alpha} + C\dot{\alpha} + E\tau = u \quad (2)$$

where $\alpha^T = [\alpha_1, \alpha_2]$, G , C and E are 2×2 diagonal matrices whose elements are $G_{ii} = R_i J_i / K_{ai} N_i$, $C_{ii} = (R_i f_i + K_{bi} K_{ai}) / K_{ai} N_i$ and $E_{ii} = R_i N_i / K_{ai}$, respectively; T is the load torque, and $u = [u_1, u_2]^T$ is the torque delivered by the motor. For the i th actuator ($i = 1, 2$), R_i is the armature-winding resistance, J_i and f_i are respectively the equivalent moment of inertia and equivalent viscous friction referred to the motor shaft, K_{ai} is the motor torque constant, K_{bi} is the back emf constant, and N_i is the gear ratio.

The combination of eqns. 1 and 2 yields an integrated dynamic equation of the manipulator-plus-actuator system:

$$\begin{aligned} & \left\{ M(q) + M_d(q, \dot{q}) + \begin{bmatrix} E^{-1}G & 0 \\ 0 & 0 \end{bmatrix} \right\} \ddot{q} \\ & + \left\{ D + \begin{bmatrix} E^{-1}C & 0 \\ 0 & 0 \end{bmatrix} \right\} \dot{q} \\ & + [K + K_d(q, \dot{q})]q + N(q, \dot{q}) \\ & = \begin{bmatrix} E^{-1}u \\ 0 \end{bmatrix} \end{aligned} \quad (3)$$

The system parameters are listed in Table 1.

3. Prediction model and embedding technique

Of importance here is the fact that the simulation model (eqn. 3) can be modified further, thereby leading to an effective prediction model. First, each element of $N(q, \dot{q})$ contains at least one power of $\sin \alpha_1$, $\sin \alpha_2$, $\sin(\alpha_1 + \alpha_2)$, an element q , or an element \dot{q} . Hence, q and \dot{q} can be factored out of $N(q, \dot{q})$ and the corresponding terms are combined with damping and stiffness matrices, so that eqn. 3 can be rewritten as

$$M(\tau)\ddot{q} + D(\tau)\dot{q} + K(\tau)q = Bu(\tau) \quad (4)$$

where the matrices $M(\tau)$, $D(\tau)$ and $K(\tau)$ are dominant polynomials in q , \dot{q} , $\cos \alpha_i$, $\sin \alpha_i$, $(\sin \alpha_i)/\alpha_i$ and so on. Second, dominant terms in matrices $M(\tau)$, $D(\tau)$ and $K(\tau)$ do not involve y_1 , y_2 , ... and y_n for a sufficiently small elastic vibration of the flexible link. Hence the dominant terms in the coefficient matrices $M(\tau)$, $D(\tau)$ and $K(\tau)$ in eqn. 4. vary no faster than the rigid-body angles and angular velocities. Thus the system (eqn. 4) can be regarded as a linear time-varying system with slow-varying parameters. This conclusion is justified in [8] by more detailed examination of the equations of motion.

3.1 ARMA model

Consider a digital control of eqns. 3 and 4 with zero-order sample and hold devices. The output vector $y(t) \in \mathcal{R}^m$ ($m = 3$) is measured at the beginning of the i th sampling interval ($t = 1, 2, \dots$), and the constant control vector $u(t) \in \mathcal{R}^r$ ($r = 2$) is applied for the duration of the i th sampling interval. We take

$$y(t) = [\alpha_1(t)\alpha_2(t)y_n(t)]^T \quad (5)$$

where α_1 and α_2 are the rigid-body angles and y_n is the transverse elastic tip deflection, measured from the undeformed position, of the flexible link that holds the payload. According to the standard linear system theory, the corresponding input/output model for eqns. 4 and 5 with digital input and digital linear output has the form of the ARMA model

$$y(t) = \sum_{i=1}^N A_i(t)y(t-i) + \sum_{i=1}^N B_i(t)u(t-i) \quad (6)$$

where $A_i \in \mathcal{R}^{m \times m}$ and $B_i \in \mathcal{R}^{m \times r}$ are matrices of appropriate dimensions, N is an integer as order of the ARMA model.

If the sampling rate is fast compared to the rigid-body angular velocities and accelerations, or in other words, fast compared to the time rates of change of the dominant terms in the coefficient matrices of eqn. 4, the coefficient matrices in eqn. 6 can be considered to vary slowly. In this case, an adaptive parameter estimator should be able to track the coefficients in eqn. 6 and predict $y(t)$ from data taken through time $t-1$. Such a prediction is the basis for the subsequent adaptive control algorithm. However, system dynamics would change owing to various operating conditions, and the order N should vary to accommodate the present dynamics. This motivates the use of an intelligent lattice filter to determine the effective order N of eqn. 6 and its corresponding parameters.

3.2 The vector-channel embedding procedure

The lattice filter was originally designed for AR models. Its extension to the ARMA model is achieved by the embedding technique to embed the AR parts as additional measurement channels. For any completely observable linear system, properly located sensors observe the same dynamics of the system. Therefore, the matrices A_i 's in eqn. 6 can be replaced by scalars a_i 's which are coefficients of the system characteristic polynomial. In this paper, however, the two-link robotic manipulator with a flexible arm is very nonlinear. This suggests the use of two vector-channel auto-regressive models [17] for two independent estimation channels, each for one link.

The first estimation channel for the rigid link consists of the first rigid angle α_1 and two inputs u_1 and u_2 ; the

Table 1: System parameters

Manipulator		
l = length of each link	= 1.5 m	r = radius of each joint = 0.05 m
m = mass of each link	= 1.2465 kg	M_1 = mass of each disc = 62.325 g
M_2 = mass of payload		EI of the flexible link = 399 nm ²
DC motor for each joint (75 V, 1.1 A)		
f = viscous friction	= 10^{-5} n/m/s	$N = 1/128$
R = armature-winding resistance	= 7.4 Ω	
J = moment of inertia	= 15.4 kg · cm ²	
K_b = back emf constant	= 2.73 V/rpm	
K_a = motor torque constant	= 265 kg · cm/A	

second estimation channel for the flexible link consists of the second rigid angle α_2 , tip displacement y_n and the two inputs. Both estimation channels have the same AR model form, i.e.,

$$\begin{aligned} z_l(t) &= \sum_{j=1}^N \begin{bmatrix} a_{lj}(t) & B_{lj}(t) \\ C_{lj}(t) & D_{lj}(t) \end{bmatrix} z_l(t-j) \\ &= \sum_{j=1}^N \mathcal{A}_{l,N,j}(t) z_l(t-j) \quad l = 1, 2 \end{aligned} \quad (7)$$

where

$$z_1 = \begin{bmatrix} \alpha_1 \\ u_1 \\ u_2 \end{bmatrix}, \quad z_2 = \begin{bmatrix} \alpha_2 & y_n \\ u_1 & 0 \\ u_2 & 0 \\ 0 & u_1 \\ 0 & u_2 \end{bmatrix} \quad (8)$$

It is apparent that the first rows of $\mathcal{A}_{l,N,j}$ contain all the original ARMA coefficients.

4 The recursive covariance lattice filter

4.1 General description of lattice filters

Let the Hilbert space $H_l(t)$ be spanned by the history process $\{z_l(t) \in \mathcal{R}^{n_l \times m_l}\}$, where the $p_l \times m_l$ matrix z_l amounts to p_l subchannels, each of which contains m_l scalar measurements. The subscript l denotes the l th main estimation channel, for the l th link. Without loss of generality, we omit subscript l for simplicity. Thus, for each estimation channel of the robot arm, we further let $H_n^k(t)$ be a closed subspace of $H(t)$ spanned by $Z_{t-k}^T = \{z(t-n), \dots, z(t-k-1), z(t-k)\}$, $1 \leq k \leq n$. In this subspace we define $P_n^k(t)$ as the orthogonal projection operator onto $H_n^k(t)$, see Lemma 2.1 in Reference 17. The basic idea of the lattice algorithm is to construct a set of orthogonal basis vectors for $H_n^k(t)$ to replace the original history process.

Then, we define the forward and backward residual error vectors for each main estimation channel (actually, the 'vector' is a $p \times m$ matrix or vector array):

$$\begin{aligned} f_0(t-n) &\triangleq z(t-n) \\ f(t-n+i) &\triangleq [I - P_n^{n-i+1}(t)]z(t-n+i) \\ & \quad i = 1, \dots, n \end{aligned} \quad (9)$$

and

$$\begin{aligned} b_0(t-1) &\triangleq z(t-1) \\ b_i(t-1) &\triangleq [I - P_i^1(t)]z(t-i-1) \\ & \quad i = 1, \dots, n-1 \end{aligned} \quad (10)$$

These can be readily achieved by using the Gram-Schmidt orthogonalisation of the history spaces. It is easy to show that the lattice equations for residual errors are [19]

$$f_{n+1}(t) = f_n(t) + K_{n+1}^f(t)b_n(t-1) \quad f_0(t) = z(t) \quad (11)$$

$$b_{n+1}(t) = b_n(t-1) + K_{n+1}^b(t)f_n(t) \quad b_0(t) = z(t) \quad (12)$$

where K_n^f and K_n^b are respectively called the forward and backward reflection coefficient matrices of dimension $p \times p$. These reflection matrices are so obtained that they minimise the residual energy matrices

$$R_n^f(t) \triangleq f_n(t)f_n^T(t) \quad (13)$$

and

$$R_n^b(t) \triangleq b_n(t)b_n^T(t) \quad (14)$$

Likewise, the cross residual energy matrix is defined by

$$C_n(t) \triangleq f_n(t)b_n^T(t-1) \quad (15)$$

Thus the reflection matrices can be adjusted through some algebra:

$$K_n^f(t) = -C_{n-1}(t)R_{n-1}^{-b}(t-1) \quad (16)$$

$$K_n^b(t) = -C_{n-1}^T(t)R_{n-1}^{-f}(t) \quad (17)$$

Substituting (11), (12), (16) and (17) into (13) and (14) yields order update equations for R_n^f and R_n^b as follows:

$$R_{n+1}^f(t) = R_n^f(t) + K_n^f(t)C_n^T(t) \quad (18)$$

$$R_{n+1}^b(t) = R_n^b(t-1) + K_n^b(t)C_n(t) \quad (19)$$

However, there exists no order update for the cross residual energy matrix $C_n(t)$. To close the lattice filter loop, we need to derive the time-update equation for $C_n(t)$, see Reference 17 for details. This causes the accumulation of round-off errors from calculation of the forward and backward error vectors in the time direction. To avoid this defect, Strobach [19] proposed a generalised covariance lattice technique called the pure order recursive lattice algorithm method.

4.2 Pure order recursive covariance lattice algorithm

The pure order recursive lattice algorithm method is developed for AR system identification calculations by expanding the residual matrix (eqns. 13-15) to the generalised residual energy matrices in block matrix forms as follows:

$$R_{n,i,j}^f(t) = f_n(t-i)f_n^T(t-j) \quad (20)$$

$$R_{n,i,j}^b(t) = b_n(t-i-1)b_n^T(t-j-1) \quad (21)$$

$$C_{n,i,j}(t) = f_n(t-i)b_n^T(t-j-1) \quad (22)$$

Substituting eqns. 11 and 12 into eqns. 20-22 leads to order update equations for the block generalised residual energy matrices:

$$\begin{aligned} R_{n,i,j}^f(t) &= R_{n-1,i,j}^f(t) + K_n^f(t-i)C_{n-1,i,j}^T(t) \\ & \quad + C_{n-1,i,j}(t)K_n^{fT}(t-j) \\ & \quad + K_n^f(t-i)R_{n-1,i,j}^b(t)K_n^{fT}(t-j) \end{aligned} \quad (23)$$

$$\begin{aligned} R_{n,i,j}^b(t) &= R_{n-1,i+1,j+1}^b(t) + K_n^b(t-i-1) \\ & \quad \times C_{n-1,i+1,j+1}(t) + C_{n-1,i+1,j+1}^T(t) \\ & \quad \times K_n^{bT}(t-j-1) + K_n^b(t-i-1) \\ & \quad \times R_{n-1,i+1,j+1}^f(t)K_n^{bT}(t-j-1) \end{aligned} \quad (24)$$

$$\begin{aligned} C_{n,i,j}(t) &= C_{n-1,i,j+1}(t) + K_n^f(t-i)R_{n-1,i,j+1}^b(t) \\ & \quad + R_{n-1,i,j+1}^f(t)K_n^{bT}(t-j-1) + K_n^f(t-i) \\ & \quad \times C_{n-1,i+1,j+1}^T(t)K_n^{bT}(t-j-1) \end{aligned} \quad (25)$$

The conventional residual energy matrices can be observed as the upper-left elements of the block generalised residual energy matrices. The reflection coefficients in eqns. 16 and 17 can thus be redefined by

$$K_n^f(t) = -C_{n-1,0,0}(t)R_{n-1,0,0}^{-b}(t) \quad (26)$$

$$K_n^b(t) = -C_{n-1,0,0}^T(t)R_{n-1,0,0}^{-f}(t) \quad (27)$$

These equations, together with eqns. 23–25, constitute a closed form for the order recursion of the covariance lattice algorithm. Two important properties hold for the block generalised residual energy matrices from their definitions. These are symmetric properties

$$R_{n,i,j}^{fT}(t) = R_{n,j,i}^f(t) \quad (28)$$

$$R_{n,i,j}^{bT}(t) = R_{n,j,i}^b(t) \quad (29)$$

but

$$C_{n,i,j}^T(t) \neq C_{n,j,i}(t) \quad (30)$$

and shift invariance properties

$$R_{n,i+1,j+1}^f(t) = R_{n,i,j}^f(t-1) \quad (31)$$

$$R_{n,i+1,j+1}^b(t) = R_{n,i,j}^b(t-1) \quad (32)$$

$$C_{n,i+1,j+1}(t) = C_{n,i,j}(t-1) \quad (33)$$

Substituting these symmetric and shift invariance properties results in a simplification of eqns. 23–25, and only four recursive equations, together with eqns. 26 and 27, are needed to complete the order-recursive covariance lattice loop.

At the sampling instant t ,

for $n = 1, N-1$

for $j = 0, N-n-1$

N : order of the algorithm or order of the AR model (7)

$$\begin{aligned} R_{n,0,j}^f(t) &= R_{n-1,0,j}^f(t) + K_n^f(t)C_{n-1,j,0}^T(t) \\ &\quad + C_{n-1,0,j}(t)K_n^{fT}(t-j) \\ &\quad + K_n^f(t)R_{n-1,0,j}^b(t)K_n^{fT}(t-j) \end{aligned} \quad (34)$$

$$\begin{aligned} R_{n,0,j}^b(t) &= R_{n-1,0,j}^b(t-1) + K_n^b(t-1) \\ &\quad \times C_{n-1,0,j}(t-1) + C_{n-1,j,0}^T(t-1) \\ &\quad \times K_n^{bT}(t-j-1) + K_n^b(t-1) \\ &\quad \times R_{n-1,0,j}^f(t-1)K_n^{bT}(t-j-1) \end{aligned} \quad (35)$$

$$\begin{aligned} C_{n,0,j}(t) &= C_{n-1,0,j+1}(t) + K_n^f(t)R_{n-1,0,j+1}^b(t) \\ &\quad + R_{n-1,0,j+1}^f(t)K_n^{bT}(t-j-1) \\ &\quad + K_n^f(t)C_{n-1,j+1,0}^T(t)K_n^{bT}(t-j-1) \end{aligned} \quad (36)$$

$$\begin{aligned} C_{n,j+1,0}(t) &= C_{n-1,j,0}(t-1) + K_n^f(t-j-1) \\ &\quad \times R_{n-1,j,0}^b(t-1) + R_{n-1,j,0}^f(t-1) \\ &\quad \times K_n^{bT}(t-1) + K_n^f(t-j-1) \\ &\quad \times C_{n-1,0,j}^T(t-1)K_n^{bT}(t-1) \end{aligned} \quad (37)$$

4.3 Initial conditions for order updates

At each sampling instant t , in the adaptive estimation process, each main estimation channel receives measured output and input information to construct the history vector $z_l(t)$, $l = 1, 2$. Therefore, the initial values of the block generalised residual energy matrices are

$$R_{0,0,j}^f(t) = z_l(t)z_l^T(t-j) = W_{0,j}(t) \quad (38)$$

$$R_{0,0,j}^b(t) = z_l(t-1)z_l^T(t-j-1) = W_{1,j+1}(t) \quad (39)$$

$$C_{0,0,j}(t) = z_l(t)z_l^T(t-j-1) = W_{0,j+1}(t) \quad (40)$$

$$C_{0,j,0}(t) = z_l(t-j)z_l^T(t-1) = W_{j,1}(t) \quad (41)$$

To accommodate the change of system parameters, we use a sliding exponential window of function $F(t) = \mu^t$, $0 < \mu \leq 1$, so that newer measurement information plays a more important role than do older measurements on determining the AR order and the corresponding parameters. In this case, the weighted initial covariance block matrices are obtained by the following recursive algorithm

$$W_{i,j}(t) = \mu W_{i,j}(t-1) + z_l(t-i)z_l^T(t-j) \quad i = 0, 1 \quad (42)$$

Several other windows are introduced in Reference 19, although we get better simulation results by using the exponential window.

4.4 AR coefficients and order determination

The coefficients $a_l(t)$ and $B_l(t)$ (subscript l is omitted) in the AR model (eqn. 7) of an effective order are required to generate optimal adaptive control signals. Recall that $f_n(t)$ is the error remaining after orthogonal projection of the data taken through time t onto the history space $H_n^1(t)$, i.e.,

$$f_n(t) = z(t) - \sum_{j=1}^n \mathcal{A}_{n,j}(t)z(t-j) \quad (43)$$

and the coefficient matrices $\mathcal{A}_{n,j}(t)$ minimise the H_n^1 norm of $f_n(t)$ over all AR models of order n . Similarly,

$$b_n(t) = z(t-n) - \sum_{j=0}^{n-1} \mathcal{B}_{n,j}(t)z(t-j) \quad (44)$$

Manipulating (eqns. 11–12) with (eqns. 43–44), we obtain the following equations

$$\mathcal{A}_{n,j}(t) = \mathcal{A}_{n-1,j}(t) + K_n^f(t)\mathcal{B}_{n-1,j-1}(t-1) \quad (45)$$

$$\mathcal{B}_{n,j}(t) = \mathcal{B}_{n-1,j-1}(t-1) + K_n^b(t)\mathcal{A}_{n-1,j}(t) \quad (46)$$

$$n = 2, \dots, N, j = 1, \dots, n-1$$

given the initial and terminal conditions

$$\begin{aligned} \mathcal{A}_{n,0}(t) &= I \quad \mathcal{A}_{n,n}(t) = -K_n^f(t) \\ \mathcal{B}_{n,0}(t) &= -K_n^b(t) \quad \mathcal{B}_{n,n}(t) = I \end{aligned} \quad (47)$$

It is apparent that the estimates of a_i and B_i are in the first rows of $\mathcal{A}_{n,j}$, $j = 1, \dots, n$. From eqns. 6 and 7, the output prediction $\hat{y}(t+1)$ at one step ahead of t can then be expressed as

$$\begin{aligned} \hat{y}(t+1) &= \sum_{i=1}^N \begin{pmatrix} a_{1i} & 0 & 0 \\ 0 & a_{2i} & 0 \\ 0 & 0 & a_{2i} \end{pmatrix} y(t-i+1) \\ &\quad + \sum_{i=1}^N \begin{pmatrix} B_{1i}|_1^2 \\ B_{2i}|_1^2 \\ B_{2i}|_3^4 \end{pmatrix} u(t-i+1) \\ &= \sum_{i=1}^N \hat{A}_i(t)y(t-i+1) \\ &\quad + \sum_{i=1}^N \hat{B}_i(t)u(t-i+1) \end{aligned} \quad (48)$$

where $B_{il}|_j^k$ denotes the j th through k th elements of the row vector B_{il} .

The effective order of the ARMA model (eqn. 48), which is twice the number of significantly excited modes,

working phase is zero because the flexible link is vertical. For the final position at the end of the second phase, the non-zero static tip deflection can be estimated by the filter (eqn. 55), and is used to correct the error in the final absolute tip position by small increments in α_1 and α_2 .

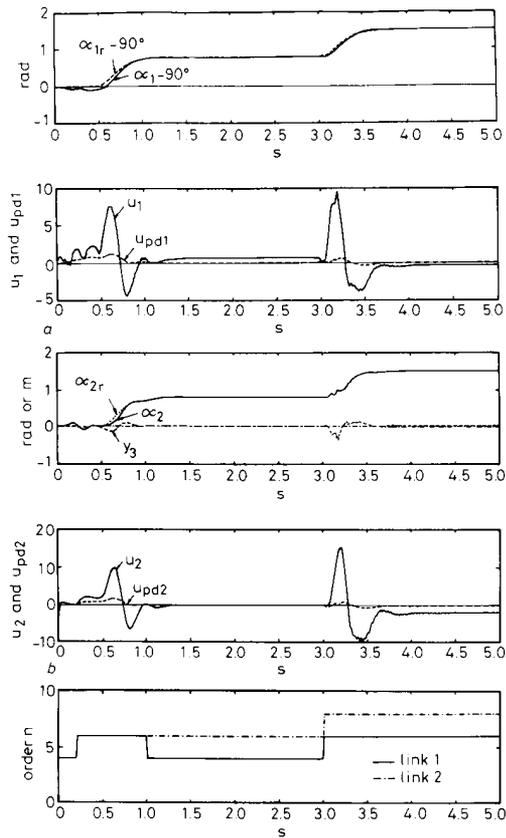


Fig. 3 PD loop
 $M_2 = 0$ for the first working phase
 $M_2 = 0.4$ m for the second working phase
 a rigid arm 1
 b flexible arm 2

The PD design is based on gravity torques and a rigid-body mass matrix that are 40% greater than their correct values for no payload. The proportional and derivative gain matrices, designed according to Reference 20, are

$$K_p = \begin{bmatrix} 266793 & 82267 \\ 82267 & 49114 \end{bmatrix} \quad K_d = \begin{bmatrix} 896862 & 27234 \\ 27234 & 13807 \end{bmatrix} \quad (56)$$

These provide at each joint a torque that is a linear function of rigid-body angular displacements and velocities at both joints.

On each sampling interval, the nonlinear response was simulated on an IBM 3090 supercomputer by solving the equations of motion in eqn. 3 with a fourth-order Runge-Kutta algorithm with variable step size. A learning period of about 45 samples is found essential before the robot motion for a converged set of parameters in the

prediction model, in which the control torques are constrained not to exceed 1.5 times the magnitudes of the initial torques holding the links at the horizontal position; but no further learning is necessary for the second

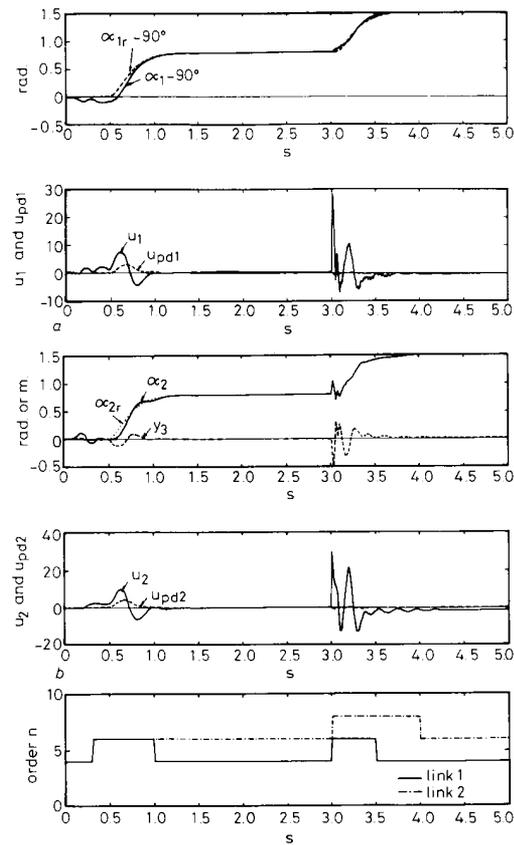


Fig. 4 PD loop
 $M_2 = 0$, impulse at $3 \text{ s} = 30 \text{ V}$
 a rigid arm 1
 b flexible arm 2

working phase. The control parameters in eqns. 50–54 are

$$Q = \text{diag} [30, 20, 0.02], \quad R_1 = \text{diag} [10^{-5}, 10^{-5}] \quad (57)$$

$$R_{20} = \text{diag} [3.5 \times 10^{-3}, 3.5 \times 10^{-2}], \quad \gamma = 1 \quad \left. \begin{array}{l} a_0 = 0.98, a_f = 0.7, \beta = 1 \\ \gamma = 1 \end{array} \right\}$$

during the learning period (58)

$$R_{20} = \text{diag} [2.5 \times 10^{-5}, 2.5 \times 10^{-4}], \quad \gamma = 0.1^{0.01} \quad \left. \begin{array}{l} a_0 = 0.98, a_f = 0.7, \beta = e^{-0.007} \\ \gamma = 0.1^{0.01} \end{array} \right\}$$

after the learning period (59)

In Fig. 3, there is no payload mass in the first working phase, but there is a payload weighing 40% of the flexible link mass in the second working phase. Initially, the orders of two main estimation channels are both set at 4, then settle respectively to 4 and 6 for the rigid link

channel and the flexible link channel in the first working phase. After the end effector grasps a payload, the orders rise to 6 and 8 in the two estimation channels. This shows that the payload mass makes the system less stiff,

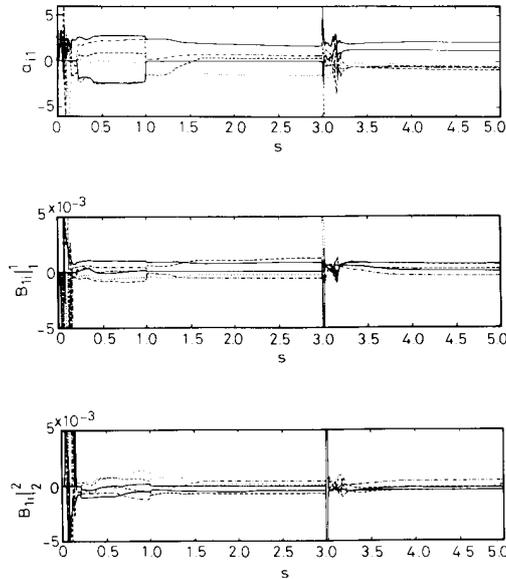


Fig. 5 Parameter variation in Fig. 3

therefore more modes are required to describe the system response. In Fig. 4, the manipulator moves without payload, although a 30 V input impulse is applied through each DC motor at the joints as the second working phase starts. The impulse excites high-frequency modes of the structure and effective orders change as shown in the figure. Fig. 5 shows the variation in the ARMA parameters corresponding to the rigid link estimation channel in the simulation of Fig. 3. These also demonstrate the fast convergence rate of the recursive covariance lattice filter.

7 Summary and conclusions

We have applied an adaptive lattice estimation and control scheme to a two-link robotic manipulator with one flexible forearm. The adaptive controller is digital and based on a reduced-order time-varying prediction model, although the simulation model of the manipulator is highly nonlinear. The combination of elastic vibrations and nonlinearities due to fast large-angle rigid-body motion necessitate certain special features of the predictor and controller presented in this paper. The recursive covariance lattice filter with the vector-channel embedding technique has a pure order recursive property, which enables the lattice to identify the effective order of the prediction model for the plant and the corresponding parameters. The error dynamics model, the time-varying weighting matrix on control variations, the continuous-time PD controller and the low-pass filter used to estimate the steady state static tip deflection were implemented inside the adaptive control loop. The simulations showed a fast convergence rate of the recursive covariance lattice

algorithm, and fast tracking ability of the proposed adaptive controller. The numerical results also shows that the resulting closed-loop system was robust during the payload change and disturbances.

8 Acknowledgment

This research was supported by the National Science Council under Contract No. NSC 79-0401-E002-10.

9 References

- 1 YANG, Y.P., and GIBSON, J.S.: 'Adaptive control of a robotic manipulator with a flexible arm', *J. Robotic Systems*, 1989, **6**, (3), pp. 217-232
- 2 NELSON, W.L., MITRA, D., and BOIE, R.A.: 'End point sensing and load-adaptive control of a flexible robot arm'. Proc. 1985 American Control Conference, pp. 1410-1415
- 3 NELSON, W.L., and MITRA, D.: 'Load estimation and load adaptive optimal control for a flexible robot arm'. 1986, IEEE International Conference on Robotics and Automation, pp. 206-211
- 4 CHUNG, C.H., and LEININGER, G.G.: 'Adaptive self-tuning control of manipulators in task coordinate system'. Proc. International Conference on Robotics, Atlanta, Georgia, March 13-15, 1984, pp. 608-612
- 5 YUH, J.: 'Application of discrete-time model reference adaptive control to a flexible single-link robot', *J. Robotic Systems*, 1987, **4**, (5), pp. 621-630
- 6 MENG, C.H., and CHEN, J.S.: 'Payload adaptation of a flexible one-link manipulator'. Proc. 1988 American Control Conference, pp. 69-73
- 7 ROVNER, D.M., and CANNON, R.C.: 'Experiments toward on-line identification and control of a very flexible one-link manipulator', *Int. J. Robotics Research*, 1987, **6**, (4), pp. 3-19
- 8 YANG, Y.P.: 'Adaptive control of robotic manipulators with flexible link'. 1988, PhD Dissertation, University of California, Los Angeles
- 9 SUNDARARAJAN, N., and MONTGOMERY, R.C.: 'Adaptive modal control of structural dynamics systems using least-squares lattice filters', *AIAA J. Guidance, Control, and Dynamics*, 1985, **8**, pp. 223-229
- 10 OAKLEY, C.M., and CANNON, R.H. Jr.: 'End-point control of a two-link manipulator with a very flexible forearm'. Proc. 1989 American Control Conference, Pittsburgh, PA, pp. 1381-1388
- 11 YANG, Y.P., and HUANG, K.M.: 'Stochastic adaptive control of a two-link manipulator with one flexible forearms'. 13th National Symposium on Automatic Control, National Taiwan University, Taipei, Taiwan, December 23, 1989, pp. 28-36
- 12 SUNDARARAJAN, N., and MONTGOMERY, R.C.: 'Identification of structural dynamics systems using least-squares lattice filters', *AIAA J. Guidance, Control, and Dynamics*, 1983, **6**, pp. 374-381
- 13 FRIEDLANDER, B.: 'Lattice filters for adaptive processing', *Proc. IEEE*, 1982, **70**, (8), pp. 829-867
- 14 FRIEDLANDER, B.: 'Recursive lattice forms for adaptive control'. Proc. Joint Automatic Control Conference, San Francisco, CA, 1980, Session WP2-E
- 15 ÖLÇER, S., and MORF, M.: 'Adaptive control by ladder forms'. Proc. 1984 American Control Conference, San Diego, CA, 1984, pp. 1265-1270
- 16 LING, F., and PROAKIS, J.G.: 'Numerical accuracy and stability: two problems caused by round-off error'. Proc. International Conference ASSP, San Diego, CA, 1984, pp. 30.3.1-30.3.4
- 17 JABBARI, F., and GIBSON, J.S.: 'Vector-channel lattice filters and identification of flexible structures', *IEEE Trans. Automatic Control*, 1988, **33**, (5), pp. 448-456
- 18 LEE, D.T.L., FRIEDLANDER, B., and MORF, M.: 'Recursive ladder algorithms for ARMA modeling', *IEEE Trans. Automatic Control*, 1982, **AC-27**, (4), pp. 753-764
- 19 STROBACH, P.: 'Recursive covariance ladder algorithms for ARMA system identification', *IEEE Trans. Acoustics, Speech, and Signal Processing*, 1988, **36**, (4), pp. 560-580
- 20 SERAJI, H.: 'Design of multivariable controllers for robot manipulators'. Proc. 1986 American Control Conference, 1986, **3**, pp. 1736-1741