This is a repository copy of *Identification of Robot Dynamics: A Parallel Processing Approach*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/78201/

Monograph:

Zomaya, A.Y. and Morris, A.S. (1989) Identification of Robot Dynamics: A Parallel Processing Approach. Research Report. Acse Report 361 . Dept of Automatic Control and System Engineering. University of Sheffield

Reuse

Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

PAM ● 629.8(S)
PAM BOX

Identification of Robot Dynamics: A Parallel Processing Approach

by

A. Y. ZOMAYA

A. S. MORRIS

Department of Control Engineering
University of Sheffield
Mappin Street
Sheffield S1 3JD
U.K.

Research Report No. 361

May 1989

Abstract:

Knowledge of the exact dynamic characteristics of robot manipulators is one of the most significant factors in designing motion control systems, since the control performance is directly dependent upon the accuracy of the dynamic model.

Dynamic models normally have complicated behaviour including varying inertia depending upon the arm configuration, uncertain load effects, non-linear effects such as the Coriolis and Centripetal forces, and interactions among joints. Unless these characteristics are included in the manipulator dynamics exactly, the performance of the controller is not expected to meet the given requirements. This necessitates the development of an efficient method to identify the dynamic parameters of robot arms.

This paper will describe the on-line estimation of the link inertial parameters using a semi-customised symbolic representation of the dynamic equations based on the Lagrangian formulation. The identification is carried out using Recursive Least Squares techniques. The whole of the algorithm is distributed over a network of parallel processors (**INMOS TRANSPUTERS**) with its associated programming language **OCCAM**. An example is given for the case of the first three links of the Stanford Arm and good real-time results are demonstrated.

1. Introduction

Present day robot manipulators are controlled by a conventional servomechanism at each joint. This approach neglects system dynamics which might lead to an overall degrading performance.[37] Several schemes had been proposed that incorporate the full dynamic model of the arm in the controller design.[20,22,17,16]

The mathematical formulation of the equations of motion of a robot manipualtor is divided into two distinct areas:

a. the inverse dynamics, which are concerned with finding the forces required to drive the arm through some specified trajectory.

b. the forward dynamics, which deal with the calculation of the position, velocity and acceleration of each link for a given set of applied forces/torques.

The dynamics consist of a set of differential, coupled and highly non-linear equations which determine the arm's dynamic behaviour. Many researchers have reported simplified forms of the dynamics based on the Lagrangian and Newtonian mechanics principles.[2,31,21,30] The Lagrangian posesses a highly structured and systematic formulation which allows for the application of a wide range of modern control techniques.[25,36] The recursive Newton-Euler(NE) is simpler computationally but with the

200091869

loss of compact structure. Both approaches have great potential for modelling and simulation of robot manipulators. The equivalence and interaction of the (NE) and (LE) has been addressed by Silver.[33] Other methods have been proposed to solve for the dynamics but the (LE) and (NE) remain the most popular.[10,32] In this paper a simplifed form of the (LE) is used.

2. Statement of the Problem

Model-based control algorithms rely on the arm dynamics being accurately known. However, this assumption is barely practical in actual situations due to the nonuniform mass distribution of the different links of the arm. Therefore, a good model is an important component of any accurate and robust robot control scheme.

The dynamic model of a robot arm is defined using two kinds of parameters. The first is the *Kinematic* (Denavit-Hartenberg) parameters, which define homogeneous transformations between successive links. These are rather easy formulated.[35,6] The second kind is the *Dynamic* parameters consisting of the mass, centre of mass and the inertia parameters of each link. Measurement of the mass properties of the links is a time consuming operation. Hence, several techniques for the identification and estimation of these properties have been introduced.[1,23,24,3,28,29,14,15,26,11,13]

The problems to be resolved prior to the actual implementation of these techniques are as follows:

1. Greater computing power is needed to enable the real-time solution of model-based problems. The use of both the (NE) and customisation techniques[27] managed to reduce the execution time of dynamic models. However, these techniques suffer from computational deficiency arising from the large amount of multiplications, additions and trignometric evaluations which hamper their real-time applicability.
2. The model must be simplified in a way that makes the extraction and regrouping of dynamic parameters easy and straightforward. On the other hand this should not lead to loss of generality.
3. The robot model must be linear in the dynamic parameters so that it can be utilised by using linear identification techniques.[19] The (LE) is linear in the dynamic parameters and the (NE) with some modification can be linearised.[15]

In this paper we propose to solve these problems as following:

a. To reduce the processing-time by distributing the algorithm over a parallel processing system. The system consists of a number of processors, in particular, the

INMOS TRANSPUTER with its programming language OCCAM. This implies the division of the whole task into smaller sub-tasks each running on a processor.

b. To reduce the computational cost of the dynamic model by using a semi-customised symbolic form of the dynamics based on the (LE). It is easier to employ this method instead of the full customisation of the dynamics which is very complicated. In addition, it is very difficult to derive these models manually for more than 2 dof (degree of freedom). The method used is linear in the dynamic parameters.

3. Robot Dynamical Model

The importance of the (LE) evolves from its simple, algorithmic and highly structured formulation. In general, the (LE) equations of motion can be written in a compact form which is the final outcome of solving the dynamics :

$$\tau(t) = \mathbf{D}(\boldsymbol{\Theta})\, \ddot{\boldsymbol{\Theta}}\,(t) + \mathbf{C}(\boldsymbol{\Theta},\dot{\boldsymbol{\Theta}}) + \mathbf{h}(\boldsymbol{\Theta}) \tag{1}$$

where $\tau(t)$ is an $n \times 1$ applied force/torque vector for joint actuators; $\Theta(t)$, $\dot{\Theta}(t)$, and $\ddot{\Theta}(t)$ are $n \times 1$ vectors representing position, velocity and acceleration respectively; $\mathbf{D}(\Theta)$ is an $n \times n$ effective and coupling inertia matrix; $\mathbf{C}(\Theta,\dot{\Theta})$ is an $n \times 1$ Corioilis and Centripetal effects vector; and $\mathbf{h}(\Theta)$ is an $n \times 1$ gravitational force vector, where (n) is the degree of freedom (dof).

The very general form of eq.(1) is important in state space and modern control applications. However, it can't be utilised unless simplified.[2, 18, 38]

In this work a semi-customised symbolic form is used. The formulation was first introduced by Bejczy[2] and later by Paul [31] , then refined and further simplified by Zomaya and Morris.[38] The conventions used are the same as those of Bejczy[2] and Paul[31] , being based on the Denavit-Hartenberg (DH) representation.[4]

The dynamic formulation is divided into two main parts :

(I) The vectors δ_l, and $\mathbf{d}_l$ which describe the differential rotation and differential transformation of link (l) respectively. These vectors are customised and symbolically expressed for a certain type of manipulator. The general description of δ^l_i, and $\mathbf{d}^l_i$ is given in Paul[31] ;

$$\mathbf{d}_i^l = \begin{cases} (-n_{lx}^{i-1}\, p_{ly}^{i-1} + n_{lx}^{i-1}\, p_{lx}^{i-1})\; i \\ (-o_{lx}^{i-1}\, p_{ly}^{i-1} + o_{lx}^{i-1}\, p_{lx}^{i-1})\; j \\ (-a_{lx}^{i-1}\, p_{ly}^{i-1} + a_{lx}^{i-1}\, p_{lx}^{i-1})\; k & \textit{revolute joint} \\ (n_{lz}^{i-1}\; i + o_{lz}^{i-1}\; j + a_{lz}^{i-1}\; k) & \textit{prismatic joint} \end{cases} \tag{2}$$

$$\boldsymbol{\delta}_i^l = \begin{cases} (n_{lz}^{i-1}\; i + o_{lz}^{i-1}\; j + a_{lz}^{i-1}\; k) & \textit{revolute joint} \\ 0 & \textit{prismatic joint} \end{cases} \tag{3}$$

(II) This part describes a general formulation of the inertial, coriolis, centripetal and gravitational effects. However, it can be further customised if a specific manipulator is used.

As given by Paul[31]

$$\mathbf{D}_{ij} = \sum_{l=max(i,j)}^{n} tr\ (\ \Delta_j^l\ \mathbf{J}^l\ \Delta_i^{l\,T}\ ) \tag{4}$$

More simplifications can be achieved by expanding eq.(4) to remove the multiplication by zero/one and redundant operations. Assume a matrix (E) such that :

$$\mathbf{E} = \begin{bmatrix} \mathbf{e} & 0 \\ 0 & 0 \end{bmatrix} \tag{5}$$

where (e) is a 3×3 matrix. Using the trace operator,

$$\mathbf{D}_{ij} = \sum_{l=max(i,j)}^{n} \ \sum_{m=1}^{3} e_{mm} \tag{6}$$

where $\sum_{m=1}^{3} e_{mm}$ is given as,

$$= J^l_{11} \begin{bmatrix}\delta_{iy}\\ \delta_{iz}\end{bmatrix}_l \begin{bmatrix}\delta_{jy}\\ \delta_{jz}\end{bmatrix}_l + J^l_{22} \begin{bmatrix}\delta_{ix}\\ \delta_{iz}\end{bmatrix}_l \begin{bmatrix}\delta_{jx}\\ \delta_{jz}\end{bmatrix}_l + J^l_{33} \begin{bmatrix}\delta_{ix}\\ \delta_{iy}\end{bmatrix}_l \begin{bmatrix}\delta_{jx}\\ \delta_{jy}\end{bmatrix}_l + J^l_{44} \begin{bmatrix}d_{ix}\\ d_{iy}\\ d_{iz}\end{bmatrix}_l \begin{bmatrix}d_{jx}\\ d_{jy}\\ d_{jz}\end{bmatrix}_l$$

$$+ J^l_{14} \left(\begin{bmatrix}\delta_{iz}\\ \delta_{jy}\end{bmatrix} \begin{bmatrix}d_{jy}\\ -d_{iz}\end{bmatrix} + \begin{bmatrix}\delta_{jz}\\ \delta_{iy}\end{bmatrix} \begin{bmatrix}d_{iy}\\ -d_{jz}\end{bmatrix} \right)_l + J^l_{24} \left(\begin{bmatrix}\delta_{jx}\\ \delta_{iz}\end{bmatrix} \begin{bmatrix}d_{iz}\\ -d_{jx}\end{bmatrix} + \begin{bmatrix}\delta_{ix}\\ \delta_{jz}\end{bmatrix} \begin{bmatrix}d_{jz}\\ -\delta_{jz}\end{bmatrix} \right)_l$$

$$+ J^l_{34} \left(\begin{bmatrix}\delta_{iy}\\ \delta_{jx}\end{bmatrix} \begin{bmatrix}d_{jx}\\ -d_{iy}\end{bmatrix} + \begin{bmatrix}\delta_{jy}\\ \delta_{ix}\end{bmatrix} \begin{bmatrix}d_{ix}\\ -d_{jy}\end{bmatrix} \right)_l \tag{7}$$

Zomaya and Morris [38] used a similar approach to describe the coriolis and centripetal effects ;

$$\mathbf{C}_{ijk} = \sum_{l=max(i,j,k)}^{n} tr\ (\Delta^l_j\, \Delta^l_k\, \mathbf{J}^l\, \Delta^{l\,T}_i) \tag{8}$$

assuming a matrix (**U**) such that :

$$\mathbf{U} = \begin{bmatrix}\mathbf{u} & 0\\ 0 & 0\end{bmatrix} \tag{9}$$

where (**u**) is a 3×3 matrix. Using the trace operator will yield,

$$\mathbf{C}_{ijk} = \sum_{l=max(i,j,k)}^{n} \sum_{m=1}^{3} u_{mm} \tag{10}$$

where $\sum_{m=1}^{3} u_{mm}$ is given as,

$$= J^l_{11}\, \delta^l_{jx} \left(\begin{bmatrix}\delta_{ky}\\ \delta_{iy}\end{bmatrix} \begin{bmatrix}\delta_{iz}\\ -\delta_{kz}\end{bmatrix} \right)_l + J^l_{22}\, \delta^l_{ix}\delta^l_{jy} \left(\begin{bmatrix}\delta_{kz}\\ 1\end{bmatrix} \begin{bmatrix}1\\ -\delta_{kx}\end{bmatrix} \right)_l + J^l_{33}\, \delta^l_{jz} \left(\begin{bmatrix}\delta_{kx}\\ \delta_{ix}\end{bmatrix} \begin{bmatrix}\delta_{iy}\\ -\delta_{ky}\end{bmatrix} \right)_l$$

$$+ J_{44}^{l}\left(d_{ix} \begin{bmatrix} \delta_{jy} \\ \delta_{jz} \end{bmatrix} \begin{bmatrix} d_{kz} \\ -d_{ky} \end{bmatrix} + d_{iy} \begin{bmatrix} \delta_{jz} \\ \delta_{jx} \end{bmatrix} \begin{bmatrix} d_{kx} \\ -d_{kz} \end{bmatrix} + d_{iz} \begin{bmatrix} \delta_{jx} \\ \delta_{jy} \end{bmatrix} \begin{bmatrix} d_{ky} \\ d_{kx} \end{bmatrix} \right)_{l}$$

$$+ J_{14}^{l}\left(\delta_{iz} \begin{bmatrix} \delta_{jz} \\ \delta_{jx} \end{bmatrix} \begin{bmatrix} d_{kx} \\ -d_{kz} \end{bmatrix} + \delta_{jx} \begin{bmatrix} \delta_{ky} \\ \delta_{kz} \end{bmatrix} \begin{bmatrix} d_{iy} \\ d_{iz} \end{bmatrix} + \delta_{iy} \begin{bmatrix} \delta_{jy} \\ \delta_{jx} \end{bmatrix} \begin{bmatrix} d_{ky} \\ -d_{ky} \end{bmatrix} - d_{ix} \begin{bmatrix} \delta_{jy} \\ \delta_{jz} \end{bmatrix} \begin{bmatrix} \delta_{ky} \\ \delta_{kz} \end{bmatrix} \right)_{l}$$

$$+ J_{24}^{l}\left(\delta_{ix} \begin{bmatrix} \delta_{jz} \\ \delta_{jy} \end{bmatrix} \begin{bmatrix} d_{ky} \\ d_{kz} \end{bmatrix} \begin{bmatrix} \delta_{jx} \\ \delta_{jy} \end{bmatrix} \begin{bmatrix} d_{ky} \\ -d_{kx} \end{bmatrix} + \delta_{jy} \begin{bmatrix} \delta_{kx} \\ \delta_{kz} \end{bmatrix} \begin{bmatrix} d_{ix} \\ d_{iz} \end{bmatrix} - d_{iy} \begin{bmatrix} \delta_{jx} \\ \delta_{jz} \end{bmatrix} \begin{bmatrix} \delta_{kx} \\ \delta_{kz} \end{bmatrix} \right)_{l}$$

$$+ J_{34}^{l}\left(\delta_{iy} \begin{bmatrix} \delta_{jy} \\ \delta_{jz} \end{bmatrix} \begin{bmatrix} d_{kz} \\ -d_{ky} \end{bmatrix} + \delta_{jz} \begin{bmatrix} \delta_{kx} \\ \delta_{ky} \end{bmatrix} \begin{bmatrix} d_{ix} \\ d_{iy} \end{bmatrix} + \delta_{ix} \begin{bmatrix} \delta_{jx} \\ \delta_{jz} \end{bmatrix} \begin{bmatrix} d_{kz} \\ -d_{kx} \end{bmatrix} - d_{iz} \begin{bmatrix} \delta_{jy} \\ \delta_{jx} \end{bmatrix} \begin{bmatrix} \delta_{ky} \\ \delta_{kx} \end{bmatrix} \right)_{l} \tag{11}$$

where

$$J_{14}^{l} = x_l \, m_l$$

$$J_{24}^{l} = y_l \, m_l$$

$$J_{34}^{l} = z_l \, m_l$$

$$J_{11}^{l} = (-I_{11}^{l} + I_{22}^{l} + I_{33}^{l})/\ 2$$

$$J_{22}^{l} = (I_{11}^{l} - I_{22}^{l} + I_{33}^{l})/\ 2$$

$$J_{33}^{l} = (I_{11}^{l} + I_{22}^{l} - I_{33}^{l})/\ 2$$

$$J_{44}^{l} = m_l$$

and the gravitational effects are given by,

$$\mathbf{h}_i = g \sum_{l=i}^{n} m_l \, \mathbf{\Psi}_l \, \mathbf{r}_l^{l} \tag{12}$$

where m_l and $\mathbf{r}_l^l$ are the mass and the centre of mass of link (l) respectively, and Ψ_l is a vector of the following form,

$$\mathbf{\Psi}_l = \begin{bmatrix} s\alpha\ \delta_{iz} - c\alpha\ \delta_{iy} \\ c\alpha\ \delta_{ix} \\ s\alpha - \delta_{ix} \\ s\alpha\ d_{iy} + c\alpha\ d_{iz} \end{bmatrix}_l \tag{13}$$

where sα and cα are sin(α) and cos(α) respectively.
The previous dynamic equations eq.(7,11,12) will be assumed throughout this work. Note the ease of extracting the inertial parameters from the previous equations.

3.1 The Inertial Parameters

The set of identifiable parameters for each link will be denoted by the vector:

$$\boldsymbol{\theta} = \left[m_l\ x_l\ y_l\ z_l\ J^l_{11}\ J^l_{22}\ J^l_{33}\ J^l_{12}\ J^l_{13}\ J^l_{23} \right] \tag{14}$$

Common assumptions are usually made to simplify the model. In this work the mass and the kinematic parameters are assumed to be known accurately. On the whole, estimation of all the link masses, dynamic parameters, and the kinematic (Denavit-Hartenberg) parameters is a problem of non-linear estimation. As a consequence, the kinematic parameters are assumed to be known and the problem is one of estimating the dynamic parameters only. In addition, the pseudo-inertia matrices of the different links are set to be diagonal. Hence, the maximum number to be identifed for each link will be 6 parameters, i.e.

$$\boldsymbol{\theta} = \left[x_l\ y_l\ z_l\ J^l_{11}\ J^l_{22}\ J^l_{33} \right] \tag{15}$$

4. Dynamic Identification

Our approach is similar to that of (Khosla et. al., An et. al.)[14,1] but the dynamic model is based on the (LE) formulaion which utilises the pseudo-inertia matrices and has been shown to be linear in the dynamic parameters. This work presents an algorithm to identify the dynamic parameters by applying a parallel processing approach.

4.1 The Transputer

The **INMOS TRANSPUTER** is a pioneering device which is considered to be the ideal component for fifth generation computers. The **T800** Transputer in (Fig. 1,2) which is used in this work is a 32 bit microcomputer with 4 Kbytes on chip RAM for high processing speed, a configurable memory interface, 4 bidirectional communication links, 64-bit floating point unit, and a timer. It achieves an instruction rate of 10 MIPS (millions of instructions per second) by running at a speed of 20 MHz. This makes the Transputer one of the first designs that incorporate several hardware features to support parallel processing. This allows for any number of Transputers to be arranged together to build a parallel processing system, and permits massive concurrency without further complexity. To provide maximum speed with minmial wiring, the Transputer uses point to point serial communication links for direct connection to other Transputers.

OCCAM is a high level language developed by **INMOS** to run on the Transputer[7,8,12] and optimise its operation. It is simple, block structured, and supports both sequential (**SEQ**) and parallel (**PAR**) features on one or more Transputers which can be used to facilitate simulation, modelling and control of complicated physical systems.[5,9]

4.2 Identification Procedure

In this paper a closed-loop type of identifier will be used. It is based on using a model subject to the same input as the system. Then the parameters of the model are adjusted so that the error between the output of the system and that of the model is minimised according to a certain criterion (Fig.3). In this regard a few problems have to be considered:

a. The parameter adaptation algorithm.

b. Initial condition setting.

c. Error criterion or performance index used.

For simulation purposes (Fig.4) will be employed instead of (Fig.3).

4.3 Parameter Adjustment Algorithm

Since ($\boldsymbol{\theta}$) in eq.(15) is an unknown parameter vector, it is replaced by an adjustable vector ($\hat{\boldsymbol{\theta}}$) identified in the parameter adaptation mechanism (Fig.4). The identification procedure is described as follows.
Let the output of the predictor in (Fig.4) be

$$\hat{f}(k) = \hat{\boldsymbol{\theta}}^T(k)\ \phi(k) \tag{16}$$

Based on the identification error

$$e(k) = \hat{f}(k) - f(k) \tag{17}$$

the equation for the recursive least-squares identification algorithm will be[34]

$$\hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{L}(k)\, e(k) \tag{18}$$

$$\mathbf{L}(K) = \frac{\mathbf{P}(k-1)\, \boldsymbol{\phi}(k)}{\lambda + \boldsymbol{\phi}^T(k)\, \mathbf{P}(k-1)\, \boldsymbol{\phi}(k)} \tag{19}$$

$$\mathbf{P}(k) = \frac{1}{\lambda}\, [\, \mathbf{I} - \mathbf{L}(k)\, \boldsymbol{\phi}^T(k)\,]\, \mathbf{P}(k-1) \tag{20}$$

$$0 < \lambda \leq 1 \ \textit{(forgetting factor)}$$

$$\mathbf{P}(0) > 0 \ \textit{(positive definite)}$$

4.4 Case Study

The first 3-dof of the Stanford arm (Table 1.) are used to demonstrate the proposed scheme.[31]

LINK PARAMETERS				
Link	Θ	a	d	α
1	Θ_1	0	0	–90
2	Θ_2	0	d_2	90
3	d_3	0	d_3	0

Table 1. Link Parameters for the First 3 dof of a Stanford-Like arm

The vectors ($\boldsymbol{\theta}$) and ($\boldsymbol{\phi}$) of the estimated dynamic parameters and the kinematic parameters and output measurements respectively. These vectors are derived from the force/torque model of the first 3 dof of the Stanford arm:

$$\boldsymbol{\theta}_3^T = \left[x_3 \;\; y_3 \;\; z_3 \right] \tag{21.1}$$

$$\boldsymbol{\phi}_3^T = \left[-m_3(\ddot{q}_2 + s_2 c_2 \dot{q}_1^2) \;\; -m_3 s_2 \ddot{q}_1 \;\; -m_3(s_2^2 \dot{q}_1^2 + \dot{q}_2^2) \right] \tag{21.2}$$

$$\boldsymbol{\theta}_2^T = \left[x_2 \;\; z_2 \;\; (J_{2xx} + J_{3xx}) \;\; (J_{2zz} + J_{3zz}) \right] \tag{21.3}$$

$$\boldsymbol{\phi}_2^T = \left[m_2 d_2 s_2 \ddot{q}_1 \;\; -m_2 d_2 c_2 \ddot{q}_1 \;\; (\ddot{q}_2 + s_2 c_2 \dot{q}_1^2) \;\; (\ddot{q}_2 - s_2 c_2 \dot{q}_1^2) \right] \tag{21.4}$$

$$\boldsymbol{\theta}_1^T = \left[(J_{1zz} + J_{2xx} + J_{2yy} + J_{3yy}) \;\; y_2 \right] \tag{21.5}$$

$$\boldsymbol{\phi}_1^T = \left[\ddot{q}_1 \;\; 2 d_2 m_2 \ddot{q}_1 \right] \tag{21.6}$$

From the previous equations it can be said that not all of the six dynamic parameters of each link have to be identified at the same time, but only that part of them which affects the force/torque of each link. In addition, some of the parameters appear in linear combination of each other such as $(J_{1zz}+J_{2xx}+J_{2yy}+J_{3yy})$ in (eq.21.5). In this case only 9 parameters have to identified from the 18 parameters for the first 3 links of the stanford arm. This simplification occurs due to arm architecture.

4.5 Simulation Results

The dynamics of the manipulator and the identification algorithm are distributed over the network shown in (Fig.5). The different elements of (eq.1) are divided using the following task-allocation strategy:

PROCESSOR		
P_1	P_2	P_3
C_{13}^1	D_{11}	C_{12}^1
C_{23}^1	D_{12}	C_{22}^1
C_{33}^1	D_{13}	h_1
C_{13}^2	D_{22}	h_3
C_{23}^2	D_{23}	-
h_2	D_{33}	-

Table 2. Three-Processors Task Allocation for the First 3-dof of the Stanford Arm

By using this optimum task-allocation, the total processing-time for the dynamics was found to be (1.056 msec). In this case the dynamics of the model and the system are computed at the same time instant because of the parallel nature of the algorithm. The utilisation (the ratio of the total processing time of each processor to the total processing time of the network) of the three processors P_1, P_2, and P_3 is 100%, 97%, and 94% respectively. A deeper treatment of the application of parallel-processing and distributed computing to robot dynamics can be found in Zomaya and Morris[39]

The identification procedure is performed recursively, that is starting form link (3) and going backwardly to link (1). The estimation algorithm is executed by processor (P). The numerical values of the dynamic parameters are those reported by Paul.[31] The identification results and processing-time values are shown in (Table 3,4,5) and (Table 6) respectively.

LINK 3			
Parameter	Initial Value	Estimated Value	Actual Value
x_3 (m)	0.2	0.0000	0.0
y_3 (m)	0.2	0.0000	0.0
z_3 (m)	0.0	−0.6447	−0.6447

Table 3. Parameter Estimates of Link (3)

LINK 2			
Parameter	Initial Value	Estimated Value	Actual Value
x_2 (m)	0.2	0.0002	0.0
z_2 (m)	0.2	0.0000	0.0
$(J_{2xx} + J_{3xx})$ $(Kg.m^2)$	0.0	0.0079	0.008
$(J_{2zz} + J_{3zz})$ $(Kg.m^2)$	0.0	2.525	2.526

Table 4. Parameter Estimates of Link (2)

LINK 1			
Parameter	Initial Value	Estimated Value	Actual Value
y_2 (m)	0.0	–0.1054	–0.1054
$(J_{1xx} + J_{1zz} + J_{2yy} + J_{3yy})$ $(Kg.m^2)$	0.0	0.3525	0.353

Table 5. Parameter Estimates of Link (1)

Processing-Time (sec)		
Link 1	Link 2	Link 3
0.125	0.226	0.047

Table 6. Processing-Time for the 3 Links

Graphical illustrations of the identification error of the three links are given in (Fig. 6,7,8). In real-time situations the most important parameters to estimates are those of the third link (Fig. 9,10,11).

5. Conclusion and Summary

In this paper the problem of identifying the dynamic parameters of a robot manipulator have been addressed. The identification procedure was based on a simplified form of the Lagrangian dynamics. This formulation has three important properties. First, being based on the Lagrangian representation, the equations are linear in the dynamic parameters which enables the application of linear identification techniques. Second, the dynamic parameters are easily recognised, extracted and grouped. Third, and most importantly, the equations are amenable to the implementation of parallel processing schemes. For the identification a Recursive Least Squares algorithm was used.

The whole of the algorithm was distributed over a parallel processing system consisting of the INMOS TRANSPUTER. The programs were written in OCCAM. Real-time results have been produced to demonstrate how the recent advances in VLSI technology can be used together with parallel processing techniques to significantly speed up the dynamic modelling of robot manipulators.

References

[1]. AN, C. H., ATKESON, C. G., AND HOLLERBACH, J. M., (1988). *Model-Based Control of a Robot Manipulator*, MIT Press, Cambridge, Mass..

[2]. BEJCZY, A. K., (1974). "Robot Arm Dynamics and Control," *NASA-JPL Technical Memorandum, 33-669.*

[3]. COIFFET, P., (1983). *Interaction with the Environment*, Prentice-Hall, N.J..

[4]. DENAVIT, H. AND HARTENBERG, R., (1955). "A Kinematic Notation for Lower Pair Mechansims Based on Matrices," *J. Applied Mechanics*, no. 22, pp. 215-221.

[5]. HAMBLEN, J. O., (1987). "Parallel Continuous System Simulation Using the Transputer," *Simulation*, vol. 49, no. 6, pp. 249-253.

[6]. HAYATI, S. A., (1983). "Robot Arm Geometric Link Parameter Estimation," in *Proc. of the 22nd IEEE conf. on Decision and Control, San Antonio*, pp. 1477-1483.

7]. INMOS,(1984), *OCCAM Programming Manual, Prentice-Hall*, Englewood Cliffs, N.J.

8]. INMOS,(1988), *IMS T800 Architecture*, Technical Note 6.

[9]. JONES, D. I. AND ENTWISTLE, P. M., (1988). "Parallel Computation of An Algorithm in Robotic Control," in *Int. Conf. on Control 88, Oxford, U.K.*, pp. 438-443.

[10]. KANE, T. AND LEVINSON, D., (1983). "The Use of Kane's Dynamical Equations in Robotics," *Int J. Robotics Res.*, vol. 2, no. 3, pp. 3-21.

[11]. KAWASAKI, H. AND NISHIMURA, K., (1988). "Terminal-Link Parameter Estimation of Robotic manipulators," *IEEE J. Robotics and Automation*, vol. 4, no. 5, pp. 485-490.

[12]. KERRIDGE, J., (1987). *OCCAM Programming: A Practical Approach*, Blackwell.

[13]. KHALIL, W., GEUTIER, M., AND KLEINFINGER, J. F., (1986). "Automatic Generation of Identification Models of Robots," *Int. J. Robotics and Automation*, vol. 1, no. 1, pp. 2-6.

[14]. KHOSLA, P. K. AND KANADE, T., (1987). "An Algorithm to Estimate Manipulator Dynamics Parameters," *Int. J. Robotics and Automation*, vol. 2, no. 3, pp. 127-135.

[15]. KHOSLA, P. K., (1988). "Estimation of Robot Dynamics Parameters," *Int. J. Robotics and Automation*, vol. 3, no. 1, pp. 35-41.

[16]. LEE, C. S. G. AND CHUNG, M. J., (1984). "An Adaptive Control Strategy for Mechanical Manipulators," *IEEE Trans. Automatic Control*, vol. 29, no. 9, pp. 837-840.

[17]. LEE, C. S. G. AND CHUNG, M. J., (1985). "Adaptive Perturbation Control with Feed-Forward Compensation for Robot Manipulators," *Simulation*, vol. 44, no. 3, pp. 127-136.

[18]. LEWIS, R. A., (1974). "Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions," *Tech. Memo. 33-679, Jet Propulsion Laboratory, Pasadena, California.*

[19]. LJUNG, L. AND SODERSTROM, T., (1983). *Theory and Practice of Recursive Identification*, MIT Press, Mass..

[20]. LUH, J. Y. S., WALKER, M. W., AND PAUL, R. P., (1980). "Resolved Acceleration Control of Mechanical Manipulators," *IEEE Trans. Automatic Control*, vol. 25, no. 3, pp. 468-447.

[21]. LUH, J. Y. S., WALKER, M. W., AND PAUL, R. P., (1980). "On-Line Computational Scheme for Mechanical Manipulators," *Trans. ASME J. Dynamic Systems, Measurements, and Control*, vol. 102, pp. 69-76.

[22]. MARKIEWICZ, B. R., (1973). "Analysis of the Computed Torque Drive Method and Comparison with the Conventional Position Servo for a Computer Controlled Manipulator," *NASA-JPL Technical Memorandum, 33-601,Psadena,CA.*.

[23]. MAYEDA, H., OSUKA, K., AND KANGAWA, A., (1984). "A New Identification Method for Serial Manipulator Arms," in *the 9th IFAC Cong., Budapest, Hangry*, vol. 5, pp. 74-79.

[24]. MUKERJEE, A. AND BALLARD, D. H., (1985). "Self-Calibration in Robot Manipulators," in *Proc. of IEEE Int. Conf. on Robotics and Automation, St. Louis, Missouri*, ed. K. S. Fu, pp. 1050-1057.

[25]. NEUMAN, C. P. AND TOURASSIS, V. D., (1983). "Robot Control: Issues and Insight," in *Proc. 3rd Yale Workshop on Applications of Adaptive Systems Theory*, pp. 179-189.

[26]. NEUMAN, C. P. AND KHOSLA, P. K., (1986). "Identification of Robot Dynamics: An Application of Recursive Estimation," in *Proc. of the 4th Yale Workshop on Application of Adaptive Systems Theory*, ed. K. S. Narendra, pp. 175-194, Plenum Press, New York.

[27]. NEUMAN, C. P. AND MURRAY, J. J., (1987). "Customized Computational Robot Dynamics," *J. Robotic Systems*, vol. 4, no. 4, pp. 503-526.

[28]. OLSEN, H. B. AND BEKEY, G. A., (1985). "Identification of Parameters in Models of Robots with Rotary Joints," in *Proc. of the IEEE Conf. on Robotics and Automation,St. Louis, Missouri*, pp. 1045-1049.

[29]. OLSEN, H. B. AND BEKEY, G. A., (1986). "Identification of Robot Dynamics," in *Proc. of the IEEE Conf. on Robotics and Automation, San Francisco, CA*, pp. 1004-1011.

[30]. ORIN, D. E., MCGHEE, R. B., VUKOBRATOVIC, M., AND HARTOCH, G., (1979). "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods," *Math. Biosci.*, vol. 43, pp. 107-130.

[31]. PAUL, R. P., (1981). *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, Mass.

[32]. RAIBERT, M. H. AND HORN, B. K., (1978). "Manipulator Control Using the Configuration Space Method," *Indusrial Robot*, vol. 5, no. 2, pp. 69-73.

[33]. SILVER, W. M., (1982). "On the Equivalence of Langrangian and Newton-Euler Dynamics for Manipulators," *Int. J. Robotics Res.*, vol. 1, no. 2, pp. 60-70.

[34]. SODERSTROM, T., LJUNG, L., AND GUSTAVSSON, I., (1978). "A Theoritical Analysis of Recursive Identification Methods," *Automatica*, no. 3, p. 231.

[35]. STONE, H. W., SANDERSON, A., AND NEUMAN, C. P., (1986). "Arm Signature Identification," in *Proc. of the 1986 IEEE conf. on Robotics and Automation, San Francisco, CA.*

[36]. TOURASSIS, V. D. AND NEUMAN, C. P., (1985). "Properties and Structure of Dynamic Robot Models for Control Engineering Applications," *Mechansim and Machine Theory*, vol. 20, no. 1, pp. 27-40.

[37]. VUKOBRATOVIC, M. AND STOKIC, D., (1983). "Is Dynamic Control Needed in Robotic Systems, and, if so, to What Extent ?," *Int J. Robotics Res.*, vol. 2, no. 2.

[38]. ZOMAYA, A. Y. AND MORRIS, A. S., (1988). "The Dynamic Performance of Robot Manipulators Under Different Operating Conditions," *Research Report No. 345, Dept. of Control Engineering, University of Sheffield, Sheffield S1 3JD, U.K.*.

[39]. ZOMAYA, A. Y. AND MORRIS, A. S., (1989). "Robot Inverse Dynamics Computation Via VLSI Distributed Architectures," *Research Report No. 350, Dept. of Control Engineering, University of Sheffield, Sheffield S1 3JD, U.K.*.

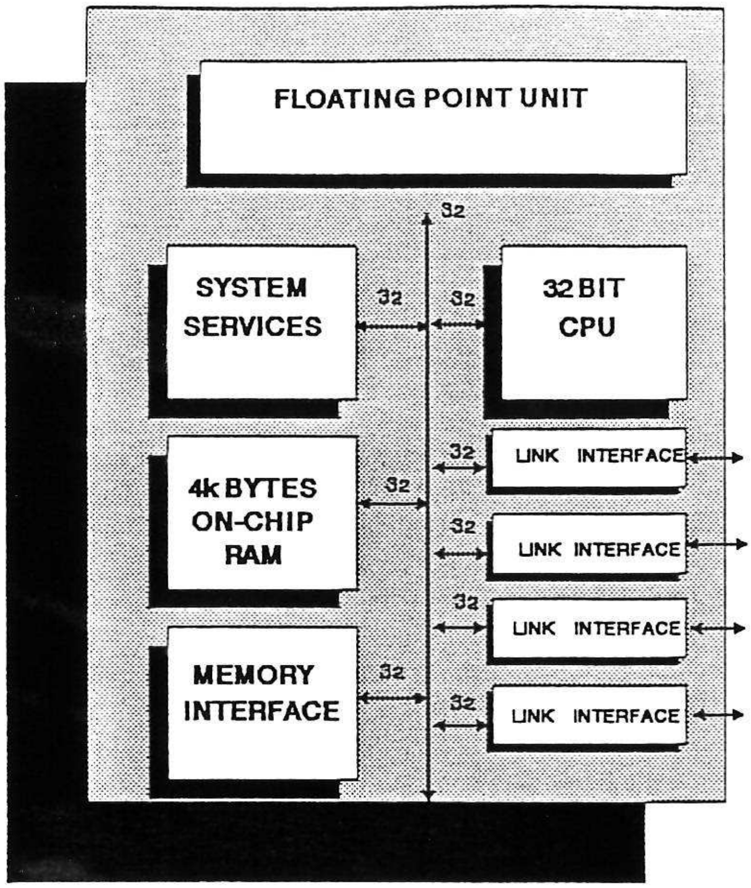


Figure 1. The well known INMOS T800 Transputer.

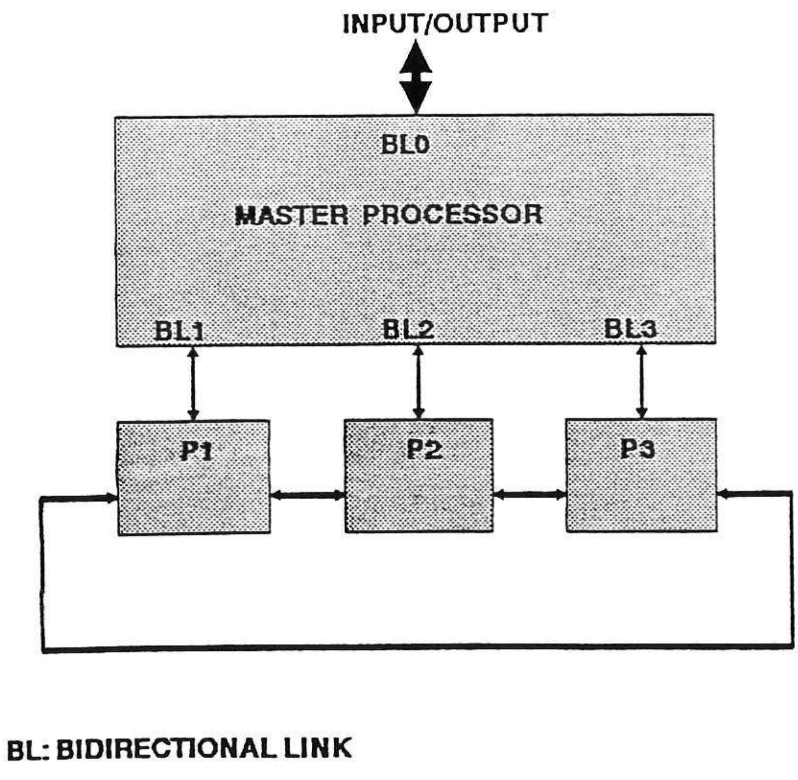


Figure 2. A Tree-Structured Transputer Network.

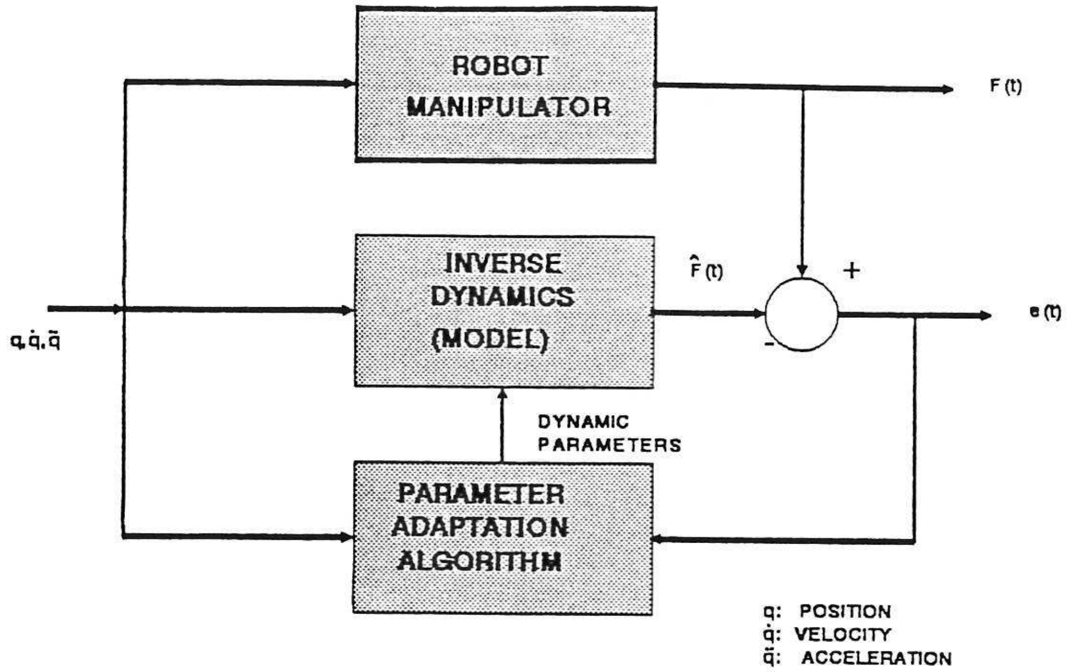


Figure 3. Closed-Loop (Learning Model) Identifier.

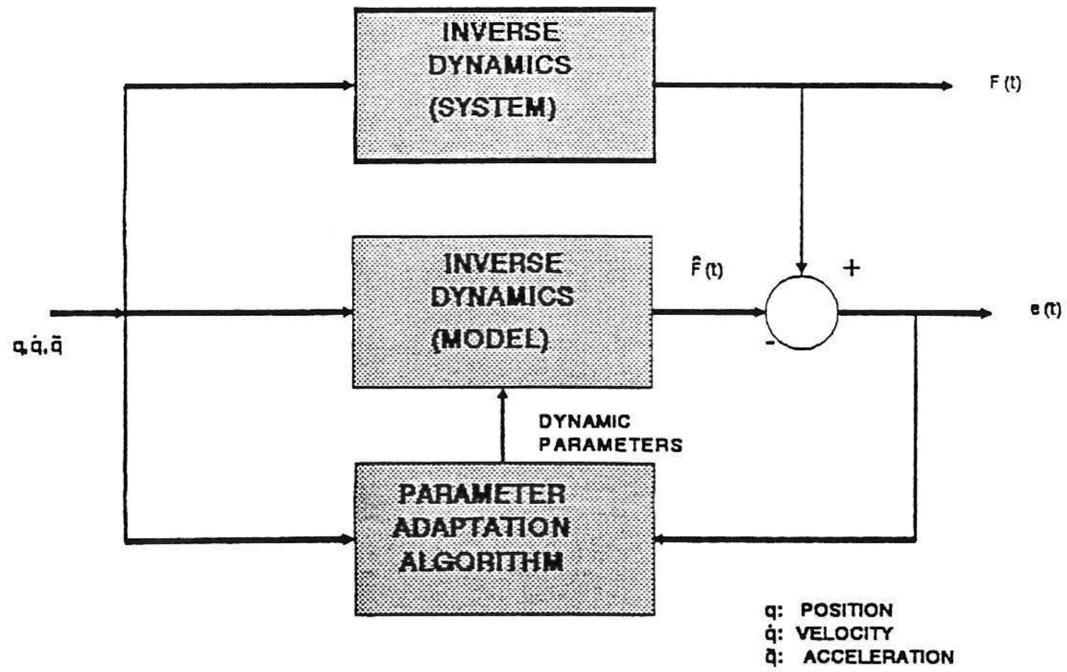


Figure 4. Alternate Closed-Loop Identifier.

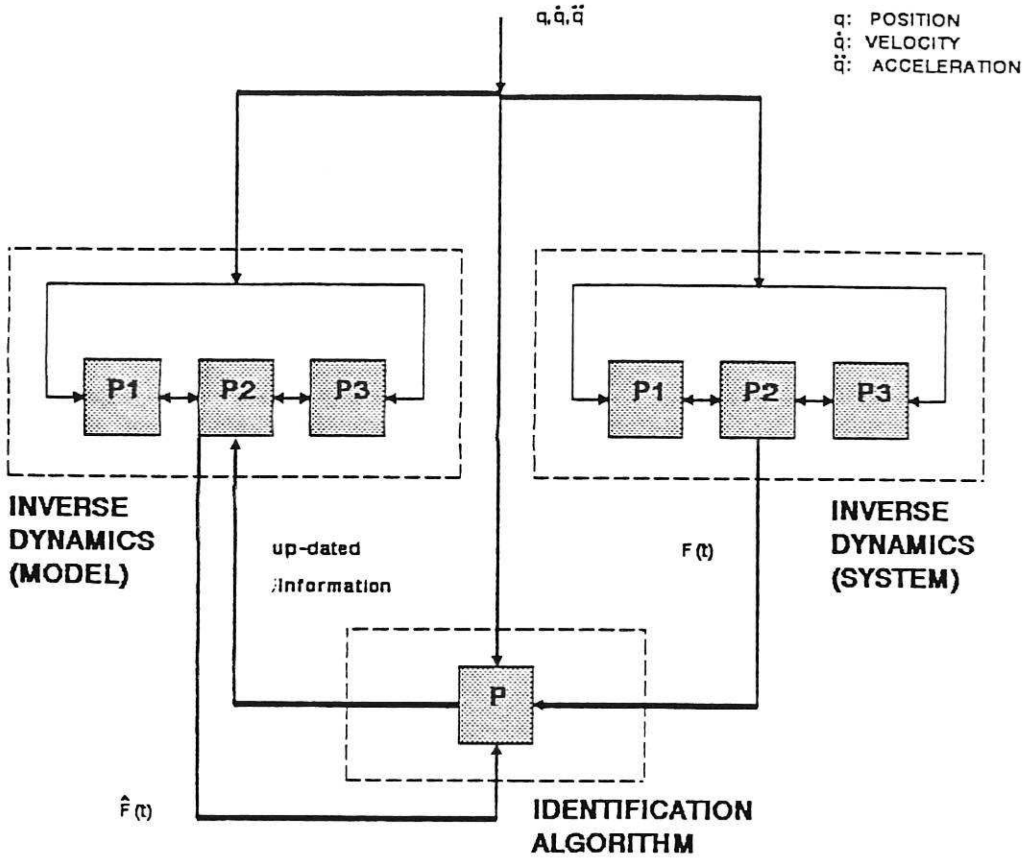
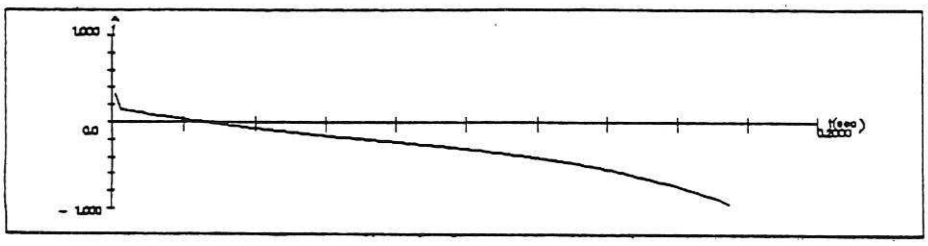
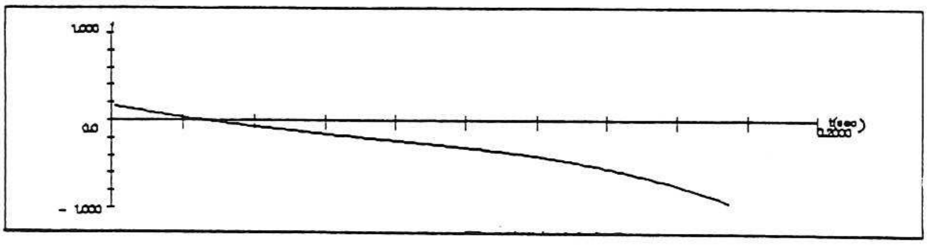


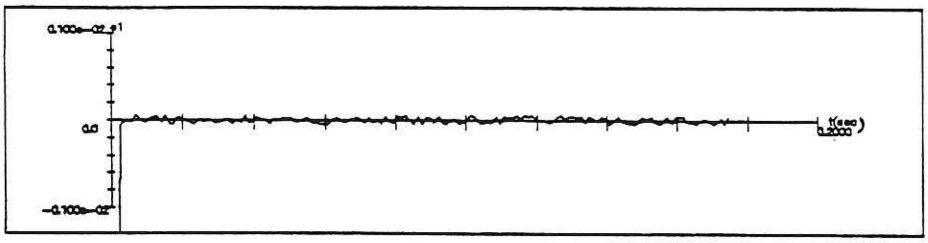
Figure 5. Block diagram describing
the Identification Procedure.



a. Model Output

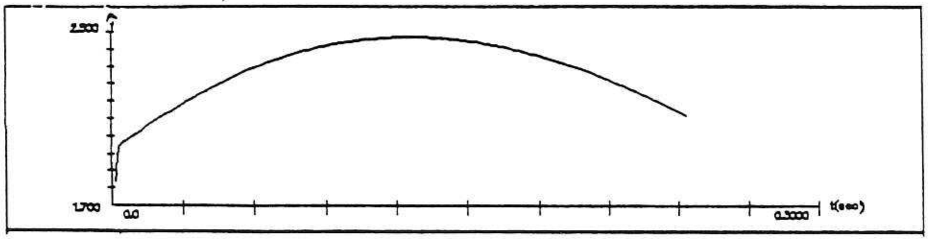


b. System Output

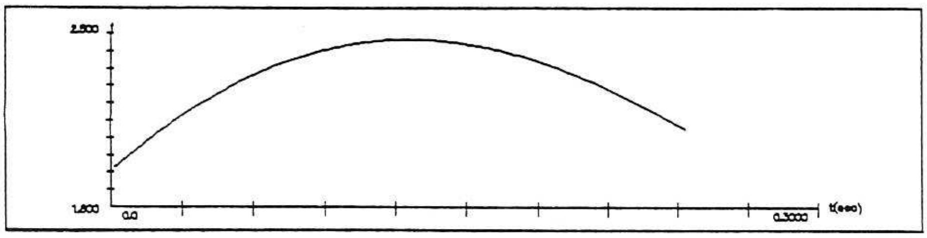


c. Error

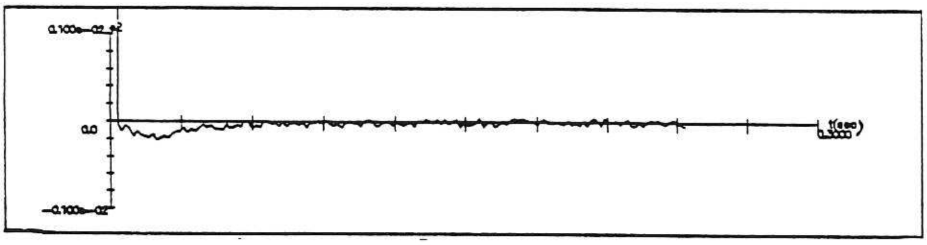
Figure 6. Identification Error of Link (1).



a. Model Output

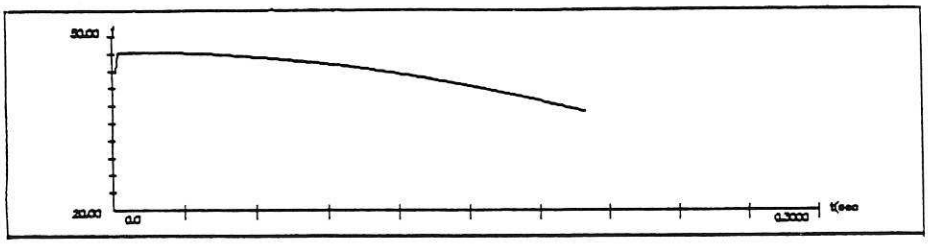


b. System Output

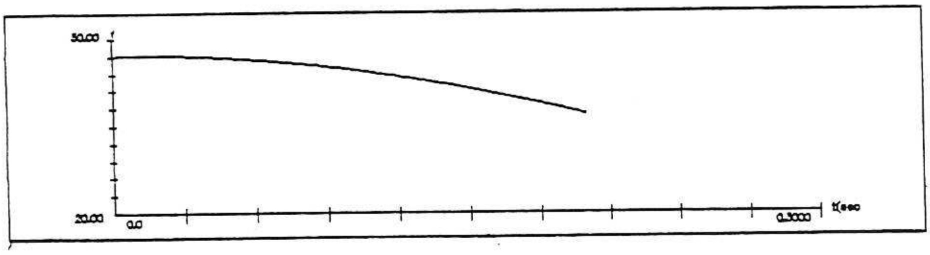


c. Error

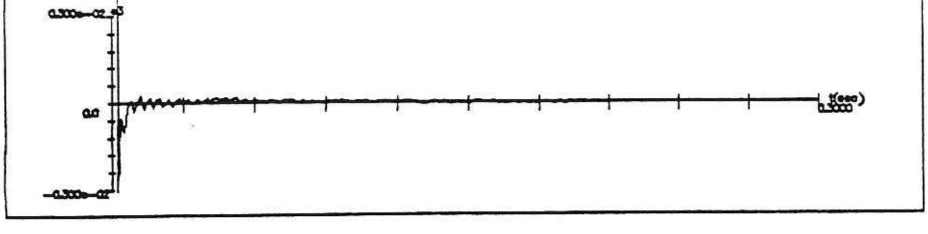
Figure 7. Identification Error of Link (2).



a. Model Output



b. System Output



c. Error

Figure 8. Identification Error of Link (3).

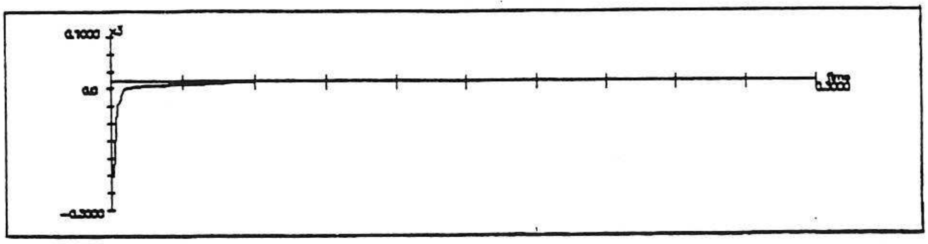


Figure 9. Identified x-component of the centre of mass of Link (3).

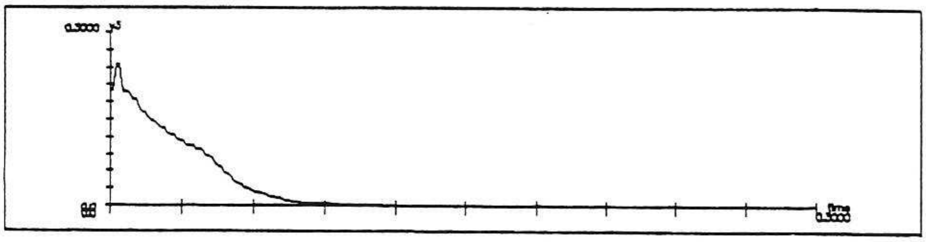


Figure 10. Identified y-component of the centre of mass of Link (3).

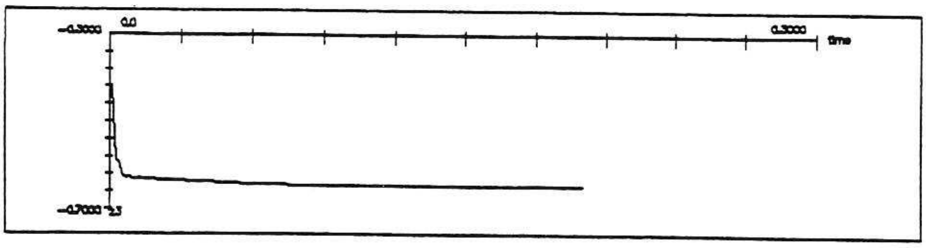


Figure 11. Identified z-component of the centre of mass of Link (3).