N90-22993

Recursive Linearization of Multibody Dynamics Equations of Motion

Tsung-Chieh Lin and K. Harold Yae

Center for Simulation and Design Optimization of Mechanical System
Department of Mechanical Engineering
The University of Iowa
Iowa City, Iowa 52242

Abstract

The equations of motion of a multibody system are nonlinear in nature, and thus pose a difficult problem in linear control design. One approach is to have a first-order approximation through the numerical perturbations at a given configuration, and to design a control law based on the linearized model. In this paper, a linearized model is generated analytically by following the footsteps of the recursive derivation of the equations of motion.

The equations of motion are first written in a Newton-Euler form, which is systematic and easy to construct; then, they are transformed into a relative coordinate representation, which is more efficient in computation. A new computational method for linearization is obtained by applying a series of first-order analytical approximations to the recursive kinematic relationships. The method has proved to be computationally more efficient because of its recursive nature. It has also turned out to be more accurate because of the fact that analytical perturbation circumvents numerical differentiation and other associated numerical operations that may accumulate computational error, thus requiring only analytical operations of matrices and vectors.

The power of the proposed linearization algorithm is demonstrated, in comparison to a numerical perturbation method, with a two-link manipulator and a seven degrees of freedom robotic manipulator. Its application to control design is also demonstrated.

1. Introduction

The behavior of a nonlinear dynamic system can be approximated by a linearized model in the neighborhood of a reference configuration. Intuitively, linear models of dynamic systems can be obtained by simply omitting nonlinear effects of the nonlinear dynamic systems, such as Coriolis forces, centrifugal forces, and the interaction forces between bodies. Such a model, however, cannot satisfy the needs of computer-aided design of control systems for multibody systems because the intuitive simplification is usually case-dependent. Therefore, a better linearized dynamic model based on a general purpose dynamics model is necessary. The approach that fits this requirement most is first-order approximations of the nonlinear dynamic models, which yield valid results in the neighborhood of the reference configuration for dynamic and control analysis. A straightforward approach to obtain first-order approximations of multibody systems is first to generate the analytical closed-form, nonlinear equations of motion of the systems, and then to generate the linearized equations of motion using first-order Taylor expansions. Unfortunately, these analytical equations of motion are generally not available because they are too complex to be generated.

Due to the difficulty of analytically generating the closed-form, nonlinear equations of motion of a multibody system, a numerical perturbation method is usually applied to obtain a linearized model at a certain configuration. For instance, in DISCOS [1] the numerical perturbation method is employed to generate a linearized model for stability analysis of a multibody system at a selected configuration. Following a similar idea, Liang [2] implemented a numerical perturbation method with DADS [3] for multibody mechanical system control. Moreover, the numerical perturbation method has been widely implemented in dynamic and control analysis. For example, Sohoni and Whitesell [4] applied it in ADAMS, and Singh, Likins, and Vendervoortt applied [5] it to generate linear models of flexible body systems.

In the implementation of the numerical perturbation method, iterative computations are employed to ensure that the resulting linearized models are accurate. However, the iterative computation sometimes may not generate a satisfactory linearized model because of failure in convergence. Therefore, a trade-off between the accuracy and convergence must be made to generate a useful linearized model. An accurate linearized model is difficult to be generated with good computational efficiency when the numerical perturbation approaches are applied. In resolving this problem, the numerical perturbation method must be avoided during the linearization procedure.

On the other hand, symbolic programming languages can be used to devise efficient computational techniques to obtain the linearized manipulator models. Vukobratovic and Nenad [6] proposed the linearization technique that first generates the nonlinear dynamic models of the manipulator by means of symbolic programming languages, and then takes first-order approximation from the given nonlinear model. Following the same approach, Neuman and Murray [7] linearized symbolically the Lagrangian dynamic robot model about a nominal trajectory to generate the linearized and trajectory sensitivity models of a manipulator. Balafoutis, Misra, and Patel [8] further extended Neuman's approach to obtain more computational efficiency in generating linearized models by using the fact that the derivatives of trigonometric functions need not be computed explicitly, and that the partial derivatives of the homogeneous transformation matrices may be obtained merely by row and column manipulations. The same idea was applied to generate linearized models for flexible multibody systems by Jonker [9]. Thus, although this approach has the advantage of not using the numerical perturbation method, there is at the same time a disadvantage: it relies heavily on symbolic programming languages. Consequently, the approach is restricted to special case studies only until a general purpose symbolic manipulation package for the dynamic modelling becomes available.

In searching for a general purpose computer-aided dynamic analysis algorithm, Bae and Haug [10,11,12] developed a recursive formulation, which was later improved by Bae, Hwang and Haug [13,14]. In this approach, the equations of motion are first written in a Newton-Euler form, which is systematic and easy to construct. They are then transformed into a relative coordinate representation, which is efficient for computation. This approach is extended in this paper to efficiently generate a linearized model using the recursive computational structure and applying the analytical linear approximations of the recursive kinematic relationships, without applying numerical perturbations. The computational efficiency and opportunity for parallelism of the recursive algorithm would make it possible to linearize successively for adaptive dynamics control.

An analytical linearization algorithm is derived by using the recursive variational derivation, and by linearizing kinematic relationships analytically. In the recursive formulation, the equations of motion are obtained through a series of coordinate transformations. By analytically taking first-order approximations of kinematic relationships between Cartesian, state vector, and joint variables and then applying these linearized relationships in the recursive variational derivation, linearized equations of motion are generated in joint space. The proposed linearization algorithm is shown in Fig. 1 and is explained as follows:

(1) Variational equations of motion are obtained in Cartesian space and the generalized mass and force are approximated with first-order Taylor expansions.

(2) First-order approximate kinematic relationships are obtained between Cartesian variables and state vector variables [14] and are substituted into the variational equations obtained in (1). Linearized variational equations in state vector space are generated.

(3) First-order Taylor expansions for the kinematical relationships between state vector variables and joint variables are obtained, and then substituted into the approximate variational equations obtained in (2).

(4) Linearized equations of motion are obtained from the approximate variational equations in joint space. At this stage, open-chain mechanisms are expressed in terms of independent coordinates and closed-chain mechanisms are expressed in a mixed differential algebraic equation (DAE) form.

(5) Linearized equations of motion expressed only in terms of independent coordinates are written in state space form for control applications.

The rest of the paper is organized as following. In Section 2, linearized kinematic relationships are expressed in terms of the generalized state vector, which is used to simplify expressions and to obtain compact equations. In Section 3, linearized relative kinematics relations are derived for two contiguous bodies. The linearized equations of motion are developed in Section 4. In Section 5, numerical examples of the recursive linearization method are given. In addition, control designs based on the linearized models and linear control theory are demonstrated. Finally, conclusions are presented in Section 6.

2. Generalized State Vector Notation

In this section, a first-order Taylor expansion is applied to approximate the relationship between Cartesian variables and generalized velocity state variables. The generalized velocity state vector, called the velocity state, is used to simplify expressions in later derivations. It is defined as [13]

$$\hat{\mathbf{Y}}_P = \begin{bmatrix} \dot{\mathbf{r}}_P + \tilde{\mathbf{r}}_P \boldsymbol{\omega}_P \\ \boldsymbol{\omega}_P \end{bmatrix} \tag{1}$$

where the subscript P represents the origin of a body-fixed frame, as shown in Fig. 2. The Cartesian velocity of point P can be written as

$$\mathbf{Y}_P = \begin{bmatrix} \dot{\mathbf{r}}_P \\ \boldsymbol{\omega}_P \end{bmatrix} \tag{2}$$

where $\dot{\mathbf{r}}_P$ and $\boldsymbol{\omega}_P$ are the translational velocity of point P and the angular velocity of a body-fixed frame at point P, respectively.

From the velocity expressions in Eqs. 1 and 2, the Cartesian velocity $\mathbf{Y}_P$ is expressed as

$$\begin{aligned} \mathbf{Y}_P &= \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{r}}_P \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \hat{\mathbf{Y}}_P \\ &\equiv \mathbf{T}_P \hat{\mathbf{Y}}_P \end{aligned} \tag{3}$$

Replacing $\dot{\mathbf{r}}_P$ by the virtual displacement $\delta \mathbf{r}_P$ and $\boldsymbol{\omega}_P$ by the virtual rotation $\delta \boldsymbol{\pi}_P$, yield the variation of the position state vector.

$$\delta \mathbf{Z}_P \equiv \mathbf{T}_P \, \delta \hat{\mathbf{Z}}_P \tag{4}$$

The Cartesian acceleration of the body-fixed frame shown in Fig. 2 is defined as the time derivative of Eq. 3,

$$\dot{\mathbf{Y}}_P = \mathbf{T}_P \, \dot{\hat{\mathbf{Y}}}_P - \mathbf{V}_P \tag{5}$$

where $V_P = \begin{bmatrix} \dot{\tilde{r}}_P \omega_P \\ 0 \end{bmatrix}$ is a velocity coupling vector.

Since the right sides of Eqs. 1 to 5 are explicitly expressed in terms of Cartesian variables, their first-order expansions can be obtained analytically with respect to perturbations in the Cartesian variables.

First, by expanding T_P at a reference configuration, Eq. 4 can be represented as

$$\delta Z_P = (T_P^o + d\, T_P^o + O(\Delta)^2)\, \delta \hat{Z}_P \qquad (6)$$

where d denotes a first-order perturbation with respect to Cartesian variables; the superscript o signifies that the quantity is evaluated at a reference configuration, i.e.,

$$d\, T_P^o = \left. \frac{\partial T_P}{\partial Z_P} \right|_{\text{evaluated at o}} \Delta$$

and Δ denotes perturbed quantity, which is expressed in terms of the variables in the Cartesian space. The perturbation of matrix T_P can be obtained as

$$dT_P = \begin{bmatrix} 0 & -d\tilde{r}_P \\ 0 & 0 \end{bmatrix} \qquad (7)$$

where the partial derivative of the position vector of point P can be expressed as

$$dr_P = d\hat{r}_P - \tilde{r}_P\, d\pi_P \qquad (8)$$

Similarly, the acceleration relationship, Eq. 5, can be expanded as

$$\dot{Y}_P = [T_P^o + d\, T_P^o + O(\Delta)^2]\ \dot{\hat{Y}}_P - [V_P^o + d\, V_P^o + O(\Delta)^2] \qquad (9)$$

The derivatives of the position state variable and the Cartesian variable can be related from Eq. 6 as

$$dZ_P = T_P\ d\hat{Z}_P \qquad (10)$$

and the derivative of the Cartesian velocity vector can be written as

$$dY_P = d\,(T_P\ \hat{Y}_P)$$

$$= T_P\, d\,\hat{Y}_P + dT_P\ \hat{Y}_P \qquad (11)$$

where

$$dT_P\ \hat{Y}_P = \begin{bmatrix} \tilde{\omega}_P & -\tilde{\omega}_P \tilde{r}_P \\ 0 & 0 \end{bmatrix} d\,\hat{Z}_P \qquad (12)$$

where $d\,\hat{Y}_P$ and $d\,\hat{Z}_P$ are the perturbations of velocity and position state vectors. Based on the relationships in Eqs. 9, 10, and 11, the derivative of the variables in the Cartesian space can be expressed in terms of the derivatives of the state variables.

3. Relative Kinematics of Two Contiguous Bodies

In this section, a first-order Taylor approximation is derived to represent relative kinematic relationships between contiguous bodies that are constrained by a kinematic joint, as shown in Fig. 3. The relative kinematic relation between the velocities of the two contiguous bodies i and j is defined as [13]

$$\hat{Y}_j = \hat{Y}_i + B_{ij}\, \dot{q}_{ij} \qquad (13)$$

where

$$\mathbf{B}_{ij} = \begin{bmatrix} \frac{\partial \mathbf{d}_{ij}}{\partial \mathbf{q}_{ij}} + (\tilde{\mathbf{r}}_j + \tilde{\mathbf{s}}_{ji})\mathbf{H}_{ij} \\ \mathbf{H}_{ij} \end{bmatrix}$$

Relationships between virtual displacements and rotations of these bodies are obtained by replacing the velocity vectors in Eq. 13 with virtual translational and rotational vectors; i.e.,

$$\delta\hat{\mathbf{Z}}_j = \delta\hat{\mathbf{Z}}_i + \mathbf{B}_{ij}\,\delta\mathbf{q}_{ij} \tag{14}$$

Taking the time derivative of Eq. 13 yields the acceleration relationship

$$\dot{\hat{\mathbf{Y}}}_j = \dot{\hat{\mathbf{Y}}}_i + \mathbf{B}_{ij}\,\ddot{\mathbf{q}}_{ij} + \mathbf{D}_{ij} \tag{15}$$

where

$$\mathbf{D}_{ij} = \dot{\mathbf{B}}_{ij}\,\dot{\mathbf{q}}_{ij}$$

As shown in Eqs. 13, 14, and 15, the state variables of body j are expressed in terms of the state variable of body i and the joint variables used to define the relative motion between bodies i and j. The first-order Taylor expansions of Eqs. 14 and 15 with respect to the state variable of body i and the joint variables can be expressed as

$$\delta\hat{\mathbf{Z}}_j = \delta\hat{\mathbf{Z}}_i + [\,\mathbf{B}^o_{ij} + d\mathbf{B}^o_{ij} + O(\Delta)^2]\,\delta\mathbf{q}_{ij} \tag{16}$$

$$\dot{\hat{\mathbf{Y}}}_j = \dot{\hat{\mathbf{Y}}}_i + [\,\mathbf{B}^o_{ij} + d\mathbf{B}^o_{ij} + O(\Delta)^2]\,\ddot{\mathbf{q}}_{ij} + [\,\mathbf{D}^o_{ij} + d\mathbf{D}^o_{ij} + O(\Delta)^2] \tag{17}$$

where $d\mathbf{B}_{ij}$ and $d\mathbf{D}_{ij}$ are computed in terms of the state variables of body i and the joint variables.

Linearized equations of motion are generated based on the linearized joint kinematic relationships and linearized relationships between the state space variables of bodies i and j. From Eqs. 13 and 14, relations between the perturbations of state variables and relative coordinates are expressed as

$$d\hat{\mathbf{Z}}_j = d\hat{\mathbf{Z}}_i + \mathbf{B}_{ij}\,d\mathbf{q}_{ij} \tag{18}$$

$$d\hat{\mathbf{Y}}_j = d\hat{\mathbf{Y}}_i + d\mathbf{B}_{ij}\,\dot{\mathbf{q}}_{ij} + \mathbf{B}_{ij}\,d\dot{\mathbf{q}}_{ij} \tag{19}$$

$$d\dot{\hat{\mathbf{Y}}}_j = d\dot{\hat{\mathbf{Y}}}_i + d\mathbf{B}_{ij}\,\ddot{\mathbf{q}}_{ij} + \mathbf{B}_{ij}\,d\ddot{\mathbf{q}}_{ij} + d\mathbf{D}_{ij} \tag{20}$$

4. Linearized Equations of Motion of a Tree Structural Mechanism

The linearized equations of motion of a tree structure mechanism that contains n joints and n+1 bodies, as shown in Fig. 4, are presented in this section. By going through the procedure of variational derivation [13] and by replacing the nonlinear kinematic relationships with their first-order approximations, one can generate the linearized equations of motion for an open-chain system. Applying the linearized kinematic relationships to the recursive variational approach yields the linearized equations of motion that are written in terms of the joint variables.

4.1 Variational Equations in Cartesian Space

The variational form of the Newton-Euler equations of motion for an n-body system is written as [3]

$$\mathbf{0} = \sum_{i=0}^{n} \delta\mathbf{Z}^T_i\,(\mathbf{M}_i\dot{\mathbf{Y}}_i - \mathbf{Q}_i) \tag{21}$$

where $\mathbf{M}_i$ is the mass matrix and $\mathbf{Q}_i$ is the generalized force. The variational equation must hold for all kinematically admissible variations $\delta\mathbf{Z}_i$, i = 1,...,n; i.e., the kinematic constraints on the system must be satisfied by $\delta\mathbf{Z}_i$. Then approximate variational equations can be generated from a set of approximations of the generalized mass and force for each body, which are expressed as

$$\mathbf{M}_i = \mathbf{M}_i^o + d\mathbf{M}_i^o + O(\Delta)^2 \qquad (22)$$

$$\mathbf{Q}_i = \mathbf{Q}_i^o + d\mathbf{Q}_i^o + O(\Delta)^2 \qquad (23)$$

where $d\mathbf{M}_i$ and $d\mathbf{Q}_i$ are expressed in terms of Cartesian variables.

4.2 Variational Equations in State Vector Space

Approximated variational equations in state vector space can be obtained by substituting Eqs. 6, 10, 22, and 26 into the equations of motion in Cartesian space and by replacing Cartesian variables with the state variables. The approximated variational equations can be written as

$$0 = \sum_{i=0}^{n} \delta\hat{\mathbf{Z}}_i^T \{[\mathbf{T}_i^{T^o}\mathbf{M}_i^o\mathbf{T}_i^o\dot{\hat{\mathbf{Y}}}_i - \mathbf{T}_i^{T^o}(\mathbf{M}_i^o\mathbf{V}_i^o + \mathbf{Q}_i^o] + [(d\mathbf{T}_i^T\mathbf{M}_i^{o}\mathbf{T}_i^o + \mathbf{T}_i^{T^o}d\mathbf{M}_i^o\mathbf{T}_i^o +$$

$$\mathbf{T}_i^{T^o}\mathbf{M}_i^o d\mathbf{T}_i^o\dot{\hat{\mathbf{Y}}}_i - (d\mathbf{T}_i^{T^o}\mathbf{M}_i^o\mathbf{V}_i^o + \mathbf{T}_i^{T^o}d\mathbf{M}_i^o\mathbf{V}_i^o + \mathbf{T}_i^{T^o}\mathbf{M}_i^o\mathbf{V}_i^o + d\mathbf{T}_i^{T^o}\mathbf{Q}_i^o + \mathbf{T}_i^{T^o}d\mathbf{Q}_i^o)] + O(\Delta)^2\} \qquad (24)$$

where $\mathbf{V}_i$ is a velocity coupling term, which is defined in Eq. 5. In order to simplify Eq. 24, the notation of the generalized mass matrix $\hat{\mathbf{M}}_i$ and force vector $\hat{\mathbf{Q}}_i$ in the state vector space [13] will be used henceforth. The equations of motion are thus expressed as

$$0 = \sum_{i=0}^{n} \delta\hat{\mathbf{Z}}_i^T \{(\hat{\mathbf{M}}_i^o\dot{\hat{\mathbf{Y}}}_i - \hat{\mathbf{Q}}_i^o) + d\,\hat{\mathbf{M}}_i^o\dot{\hat{\mathbf{Y}}}_i - d\hat{\mathbf{Q}}_i^o + O(\Delta)^2\} \qquad (25)$$

where

$$d\hat{\mathbf{M}}_i = d\mathbf{T}_i^T\mathbf{M}_i\mathbf{T}_i + \mathbf{T}_i^T d\mathbf{M}_i\mathbf{T}_i + \mathbf{T}_i^T\mathbf{M}_i d\mathbf{T}_i$$

$$d\hat{\mathbf{Q}}_i = d\mathbf{T}_i^T\mathbf{M}_i\mathbf{V}_i + \mathbf{T}_i^T d\mathbf{M}_i\mathbf{V}_i + \mathbf{T}_i^T\mathbf{M}_i d\mathbf{V}_i + d\mathbf{T}_i^T\mathbf{Q}_i + \mathbf{T}_i^T d\mathbf{Q}_i$$

and $d\mathbf{M}_i$, $d\mathbf{T}_i$, $d\mathbf{V}_i$ and $d\mathbf{Q}_i$, which are expressed in terms of the perturbations of the Cartesian variables, can be rewritten in terms of the perturbations of the state variables by substituting the relationship between the Cartesian variables and the state variables into their expressions.

4.3. Linearized Equations of Motion in Joint Space

The approximated variational equations of motion in state vector space can be rewritten in terms of joint variables. As the results in Section 3 indicate, the variables in state vector space can be transformed into joint variables. The linearized equations in joint space can be obtained by applying the following procedures. Substituting the approximated kinematic relationship between bodies n and n-1 into the variational equations yields

$$0 = \sum_{i=0}^{n-1} \delta\hat{\mathbf{Z}}_i^T \{(\hat{\mathbf{M}}_i^o\dot{\hat{\mathbf{Y}}}_i - \hat{\mathbf{Q}}_i^o) + d\hat{\mathbf{M}}_i^o\dot{\hat{\mathbf{Y}}}_i - d\hat{\mathbf{Q}}_i^o + O(\Delta)^2\} +$$

$$\delta\hat{\mathbf{Z}}_{n-1}^T \{(\hat{\mathbf{M}}_n^o\dot{\hat{\mathbf{Y}}}_n - \hat{\mathbf{Q}}_n^o) + d\hat{\mathbf{M}}_n^o\dot{\hat{\mathbf{Y}}}_n - d\hat{\mathbf{Q}}_n^o + O(\Delta)^2\} + \qquad (26)$$

$$\delta q_n^T \{ (B_n^{T^o} + dB_n^{T^o} + O(\Delta)^2) [(\hat{M}_n^o \dot{Y}_n - \hat{Q}_n^o) + d\hat{M}_n^o \dot{Y}_n - d\hat{Q}_n^o + O(\Delta)^2]\}$$

where the perturbations are taken with respect to the state variables of the inboard bodies and the relative variables that are used to defined the relative motion between the bodies n and n-1. Moreover, the relative kinematic matrix $B_{n-1\ n}$ is denoted as B_n to simplify the expressions during the derivation.

Because joint n is not subject to any relative constraint between the connected bodies, the virtual displacement of the joint coordinate is arbitrary. Thus the coefficient of δq_n^T is equal to zero; i.e.,

$$0 = (B_n^{T^o} + dB_n^{T^o} + O(\Delta)^2) [(\hat{M}_n^o \dot{Y}_n - \hat{Q}_n^o) + d\hat{M}_n^o \dot{Y}_n - d\hat{Q}_n^o + O(\Delta)^2] \quad (27)$$

Substituting the first-order Taylor expansion of the acceleration vector into state vector space, and substituting the relationships between state and joint spaces into Eq. 27, gives the equation of motion corresponding to joint n.

$$0 = B_n^{T^o} (\hat{M}_n^o \dot{Y}_n^o - \hat{Q}_n^o) + d\,(B_n^{T^o} \hat{M}_n^o \dot{Y}_n^o) - d(B_n^{T^o} \hat{Q}_n^o) + O(\Delta)^2) \quad (28)$$

where

$$d\,(B_n^T \hat{M}_n \dot{Y}_n) = dB_n^T \hat{M}_n \dot{Y}_n + B_n^T d\hat{M}_n \dot{Y}_n + B_n^T \hat{M}_n d\dot{Y}_n$$

$$d(B_n^T \hat{Q}_n) = dB_n^T \hat{Q}_n + B_n^T d\hat{Q}_n$$

and $d\hat{M}_n$ and $d\hat{Q}_n$ can be written in terms of the state variables of body n-1 and the relative variables that are used to define the relative motion between bodies n and n-1. Moreover, the equation of motion corresponding to joint n at the reference configuration is

$$0 = B_n^{T^o} (\hat{M}_n^o \dot{Y}_n^o - \hat{Q}_n^o) \quad (29)$$

Substituting the relationship in Eq. 29 into Eq. 28 and omitting higher order terms yields

$$0 = d\,(B_n^{T^o} \hat{M}_n^o \dot{Y}_n^o) - d(B_n^{T^o} \hat{Q}_n^o) \quad (30)$$

where $\dot{Y}_n$ can be expressed in terms of the derivative of a set of independent variables

x (= [$\dot{Y}_0^T$ Y_0^T $\hat{Z}_0^T$ $\ddot{q}_1^T$ $\dot{q}_1^T$ q_1^T ... $\ddot{q}_n^T$ $\dot{q}_n^T$ $q_n^T]^T$) by substituting the relationships from Eqs. 18, 19, and 20 recursively for j from n to 1.

Following the same arguments used in obtaining Eq. 30, one can obtain the linearized equations of motion corresponding to joint i as

$$\begin{aligned} 0 = {} & B_i^{T^o}\, d(K_i \dot{Y}_i + K_{i+1} B_{i+1} \ddot{q}_{i+1} + \dots + K_n B_n \ddot{q}_n - L_i)^o + \\ & dB_i^{T^o} (K_i \dot{Y}_i + K_{i+1} B_{i+1} \ddot{q}_{i+1} + \dots + K_n B_n \ddot{q}_n - L_i)^o \\ & \text{for } i = 1, \dots, n \end{aligned} \quad (31)$$

where $K_i = \hat{M}_i + K_{i+1}$, $L_i = \hat{Q}_i + L_{i+1} - K_{i+1} D_{i+1}$, $K_n = \hat{M}_n$, and $L_n = \hat{Q}_n$.

By repeating the above procedure in backward path sequence to the base body, one can obtain the linearized equation of motion for the base body as

$$d(K_0 \dot{Y}_0 + K_1 B_1 \ddot{q}_1 + \dots + K_n B_n \ddot{q}_n - L_0)^o = 0 \quad (32)$$

For the open-chain mechanism, the linearized equations of motion at the reference configuration, which are represented in Eqs. 31 and 32, can be obtained recursively. During the derivation, every perturbed term in the linearized equations can be computed either from the perturbed variables in the Cartesian variables, which are computed analytically, or from the analytically linearized relationships among the Cartesian, state vector, and joint coordinate spaces.

5. Numerical Examples

In this section, the applications of the recursive linearization algorithm are illustrated by two examples: a two-link manipulator and a robot arm with seven degrees of freedom. The accuracy and computational efficiency of the proposed algorithm are demonstrated by comparing the models obtained from the recursive algorithm with those obtained from the analytical approach and from a numerical perturbation method. In the case of the two-link system, an exact linearization is accomplished by the use of the symbolic manipulator (MACSYMA) [15]. However, to generate the exact linearized model for a complicated system is very difficult, even with a symbolic manipulator. In the second example, a numerical perturbation is applied to the robot with seven degrees of freedom in order to generate a reference linearized model with which the results of the recursive linearization algorithm are compared.

5.1 A Two-Link Manipulator

In this subsection, a two-link manipulator, as shown in Fig. 7, is modeled and tested. Since all the joints are revolute, one independent coordinate is assigned to each joint. The manipulator can be modeled as a system of two differential equations. For this system, the linearization can also be carried out analytically by using the symbolic manipulator (MACSYMA). Therefore, it is possible to check the accuracy of the recursive linearization algorithm by comparing the linear models obtained from both approaches: recursive linearization and MACSYMA implementation.

The recursive linearization produced a linearized model at the specified configuration that was defined by setting θ_1, θ_2, $\dot{\theta}_1$, and $\dot{\theta}_2$ to zero. The linearized equation of motion is written as

$$\begin{bmatrix} d\dot{\theta}_2 \\ d\dot{\theta}_1 \\ d\ddot{\theta}_2 \\ d\ddot{\theta}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -5.191 & 3.4612 & 0 & 0 \\ 1.1537 & -4.0380 & 0 & 0 \end{bmatrix} \begin{bmatrix} d\theta_2 \\ d\theta_1 \\ d\dot{\theta}_2 \\ d\dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -0.0824 & 0.0294 \\ 0.0294 & -0.0176 \end{bmatrix} \begin{bmatrix} dT_2 \\ dT_1 \end{bmatrix} \tag{33}$$

where T_1 and T_2 are actuating torques that are applied at the revolute joints. At the same specified configuration, the symbolic manipulation generated the exact linearized model, which is identical to the one obtained from the recursive linearization approach. A comparison of the numerical and analytical results shows that the proposed recursive linearization algorithm can generate a correct linearized model at a given configuration.

After the linearized model is obtained, a linear controller can be designed by applying the linear model to existing control design tools. A linear regulator is designed to control the motion of the manipulator by using the Pro-Matlab package. The pole placement algorithm [16] is used to compute the full state feedback gain matrix for the nonlinear dynamic model. The effectiveness of this regulator is tested by applying an initial deviation of the system and using the regulator to stabilize it. As expected, the linearized model can well represent the nonlinear model. Therefore, a small initial deviation is tested first. The results of 0.05 radian initial deviations are shown in Figs. 6 and 7. The nonlinear system can be stabilized by the linear regulator. Similar results are presented in Figs. 8 and 9 for 1.0 radian initial deviations.

However, the pole placement algorithm for a multiple-input multiple-output system does not have a unique solution [17]. The feedback gain obtained from the Pro-Matlab package is an iterating solution, which is designed to find an insensitive set for the configuration change. However, this algorithm requires a lot of computation to generate an optimal gain matrix. Thus, this algorithm cannot be used for an on-line computation for the real time simulation. To fulfill the on-line computation requirement, a simple and stable pole placement algorithm is needed. A case particularly interesting is to determine the feedback controller [17] in such a way that the closed loop equation is decomposed into a set of n decoupled second-order differential equations.

$$0 = d\ddot{\theta}_i + 2\xi_i \omega_i d\dot{\theta}_i + \omega_i^2 d\theta_i \,; \qquad i = 1, ..., n \tag{34}$$

where the damping factor ξ_i and the undamped frequency ω_i of each tracking error are specified by the designer. Defining the nxn constant diagonal matrices $\Lambda_1 = \text{diag}_i\{2\xi_i \omega_i\}$ and $\Lambda_2 = \text{diag}_i\{\omega_i^2\}$, one can obtain the desired decoupled closed loop equation from Eq. 34 as

$$\mathbf{0} = d\ddot{\theta} + \Lambda_1 d\dot{\theta} + \Lambda_2 d\theta \tag{35}$$

The closed loop equation of the linearized model with a proportional-derivative (PD) controller can be written in a second order differential equation form as

$$\mathbf{0} = \mathbf{M}\, d\ddot{\theta} + (\mathbf{P}_1 - \mathbf{K}_V) d\dot{\theta} + (\mathbf{P}_2 - \mathbf{K}_P)\, d\theta \tag{36}$$

where $\mathbf{K}_P$ is the position feedback gain matrix and $\mathbf{K}_V$ is the velocity feedback gain matrix. Equating coefficients in Eqs. 35 and 36 gives the desired closed-loop feedback gain matrices as

$$\mathbf{K}_V = \mathbf{M}\Lambda_1 - \mathbf{P}_1 \tag{37}$$

$$\mathbf{K}_P = \mathbf{M}\Lambda_2 - \mathbf{P}_2 \tag{38}$$

Consequently, a linear regulator is designed to control the dynamic system. As shown in Figs. 10, 11, 12, and 13, the linear regulator can stabilize the nonlinear dynamic model for both small and large initial deviation cases.

5.2 A Robot with Seven Degrees of Freedom

Figure 10 shows a robot arm that has seven degrees of freedom. The system consists of eight bodies, including the base body, which is designated as ground. The adjacent bodies are connected by revolute joints. Joints 1 to 7 are identified as Shoulder Roll, Shoulder Pitch, Elbow Roll, Elbow Pitch, Wrist Roll, Wrist Pitch, and Toolplate Roll.

Since adjacent bodies are connected by revolute joints, one generalized coordinate is assigned to each joint. The motion of this system can be described by seven generalized coordinates; the dynamic system is thus formulated as a system of seven differential equations. When a reference configuration is selected, a linearized model can be generated at this configuration using the proposed linearization algorithm.

The configuration that is shown in Fig. 10 is selected as a reference configuration: the angles of all the joints are zero ($q_1, q_2, ..., q_7 = 0$), and the velocities of all the joints are also zero ($\dot{q}_1, ..., \dot{q}_7 = 0$). At this reference configuration, the linearized model that is obtained from the recursive linearization algorithm is expressed as

$$d\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{M}^{-1}\mathbf{P}_2 & \mathbf{M}^{-1}\mathbf{P}_1 \end{bmatrix} d\mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{P}_3 \end{bmatrix} d\mathbf{u} \tag{39}$$

where $\mathbf{x} = [q_1\ q_2\ q_3\ q_4\ q_5\ q_6\ q_7\ \dot{q}_1\ \dot{q}_2\ \dot{q}_3\ \dot{q}_4\ \dot{q}_5\ \dot{q}_6\ \dot{q}_7]^T$, $\mathbf{u}$ is the actuating torque vector, $\mathbf{M}$ is the generalized mass matrix that is expressed in terms of joint variables, and

$M^{-1}P_1 = 0_{7x7}$

$$M^{-1}P_2 = \begin{bmatrix} 15.737 & 0 & -1.607 & 0 & -.1451 & 0 & -.18e-8 \\ 0 & 15.228 & 0 & 2.8549 & 0 & -.3485 & 0 \\ .41697 & 0 & 9.2375 & 0 & -.4480 & 0 & -.13e-7 \\ 0 & -11.44 & 0 & -6.361 & 0 & .5637 & 0 \\ -29.11 & 0 & 3.711 & 0 & 21.533 & 0 & .93e-7 \\ 0 & -5.0345 & 0 & -13.19 & 0 & -15.76 & 0 \\ 14.914 & 0 & 5.9428 & 0 & -25.38 & 0 & -.58e-7 \end{bmatrix}$$

$P_3 = I$

In this case, to generate a closed-form analytical expression for the linearized model is too difficult for accuracy checking even if the symbolic manipulation is employed. Instead, two comparisons were derived to make certain that the linearized model in Eq. 39 accurately represents the nonlinear model. In the first comparison, both the nonlinear and linearized models were perturbed with the same amount, and then the resulted acceleration changes were examined. In the second comparison, two linearized models—one obtained from the recursive approach and the other obtained from the numerical perturbation method—are examined.

In the first comparison, perturbation of the generalized coordinate **x** by 10^{-6} incurs the relative error of the acceleration changes between the linearized and nonlinear model as

$$\frac{\| d\ddot{x}^* - d\ddot{x} \|_2}{\| d\ddot{x}^* \|_2} = 2.739\ e{-6} \qquad (40)$$

where $d\ddot{x}$ is the acceleration change obtained from the linearized model and $d\ddot{x}^*$ is obtained from the nonlinear model.

In the second comparison, a simple numerical perturbation without any convergence checking is implemented to generate a linearized model, which will serve as a reference in comparing the recursive linearization with the numerical perturbation. Comparing the linearized model obtained from the numerical method with those obtained from the recursive approach, one can observe that at the given configuration both approaches generate nearly identical linearized models, in which the relative difference is less than 10^{-6}.

However, the recursive algorithm proves to be more efficient than the numerical perturbation method. In comparison with the numerical perturbation method, the recursive linearization took half the cpu time to generate a linear model, even though the numerical perturbation method used here was a relatively simple one. If a convergence checking algorithm was employed for the numerical perturbation method, it would take even longer to generate a linear model.

The simple pole placement used in the previous example was used again to design a linear regulator. The desired closed-loop poles were selected to make the simulation results similar to the experimental results. After properly selecting the desired poles, we used this linear regulator to control the nonlinear dynamic model. The step response of Joint 4 is shown in Fig. 11. From this result, it is clear that a simple regulator based on a linearized model simulates the behavior of a complicate control system around a reference configuration.

6. Conclusion

In these examples, we have shown that the proposed linearization algorithm is both efficient and accurate in generating a linear model at given configurations. These linear models are converted to the standard state space forms, which are convenient for linear control design. Moreover, the driving force input for a required motion around a given configuration can be predicted by using the linearized model. When a large gross motion is involved in a prescribed trajectory, more than one linearized model may be necessary for robust control. In such a case, the computation of linearization must be fast enough to update the linearized model before it fails to represent the system adequately. With the emerging parallel processing computers and computation algorithm, the use of successive linearization will be possible for on-line adaptive control.

References

1. Bodley, C., Devers, A., Park, A., and Frisch, H., A digital Computer Program for Dynamic Iteration and Simulation of Controls and Structures (DISCO), NASA Tech. Paper 1219, May 1978.
2. Liang, C.G., and Lance, G.M., Dynamic Analysis and Control Synthesis of Integrated Mechanical Systems, Ph.D. Thesis, University of Iowa, 1985.
3. Haug, E.J., Computer Aided Kinematics and Dynamics of Mechanical Systems, Volume I: Basic Method, Allyn & Bacon, Boston,1989.
4. Sohoni, V.N., and Whitesell, J., "Automatic Linearization of Constrained Dynamical Models," Journal of Mechanism, Transmission, and Automation in Design, Transactions of the ASME, vol. 108, Sept., 1986, p. 300-304.
5. Singh, R.P., Likins, P.W., and VenderVoortt, R.J., "Automated Dynamics and Control Analysis of Constrained Multibody System," Robotic & Manufacturing Automation, 1985, p. 109-113.
6. Vukobratovic, M. and Nenad, K., "Computer-oriented Method for Linearization of Dynamic Models of Active Spatial Mechanisms," Mechanism and Machine Theory, Vol. 17, No. 1, 1982, p.21-32.
7. Neuman, C., "Linearization and Sensitivity Functions of Dynamic Robot Models," IEEE Transactions on Systems, Man, and Cybernetics, vol. 14, no. 6, 1984, p.805-818.
8. Balafoutis, C.A., Misra, P., and Patel, R.V., "Recursive Evaluation of Linearized Dynamic Robot Models," IEEE J. of Robotics and Automation, Vol. RA-2, No. 3, 1986, p. 146-155.
9. Jonker, J.B., A Computer-Oriented Method for Linearization of The Dynamic Mechanism Equations, Lab. Report no. 822, Department of Mechanical Engineering, Delft University of Technology, The Netherlands, 1986.
10. Bad, D.S. and Haug, E.J., "A recursive Formulation for Constrained Mechanical Systems, Part I- Open Loop," Mechanics of Structures and Machines, Vol. 15, No. 4, 1987.
11. Bad, D.S. and Haug, E.J., "A recursive Formulation for Constrained Mechanical Systems, Part II- Closed Loop," Mechanics of Structures and Machines, Vol. 15, No. 4, 1987.
12. Bad, D.S., Haug, E.J., and Kuhl, J.G., "A recursive Formulation for Constrained Mechanical Systems, Part III- Parallel Processor Implementation," Mechanics of Structures and Machines, Vol. 16, No. 2, 1988.
13. Bae, D.S., Hwang, R.S., and Haug, E.J., "A Recursive Formulation for Real-Time Dynamic Simulation," Submitted to Journal of Mechanisms, Transmissions, and Automation in Design.
14. Hwang, R.S., Bae, D.S., Kuhl, J.G., and Haug, E.J., "Parallel Processing for Real-Time Dynamic Simulation," Submitted to Journal of Mechanisms, Transmissions, and Automation in Design.
15. VAX UNIX Macsyma Reference Manual, Version 11, Symbolics, Inc. 1985.
16. Kautsky, J., Nichols, N.K., and Van Dooren, P., "Robust Pole Assignment in Linear State Feedback," International Journal of Control, Vol. 41, No. 5, p. 1129-1155, 1985.

17. Brady, Hollerbach, Johnson, Lozano-P'eroz, and Mason, editor, Robot Motion: Planning and Control, The MIT Press, Cambridge, 1983.

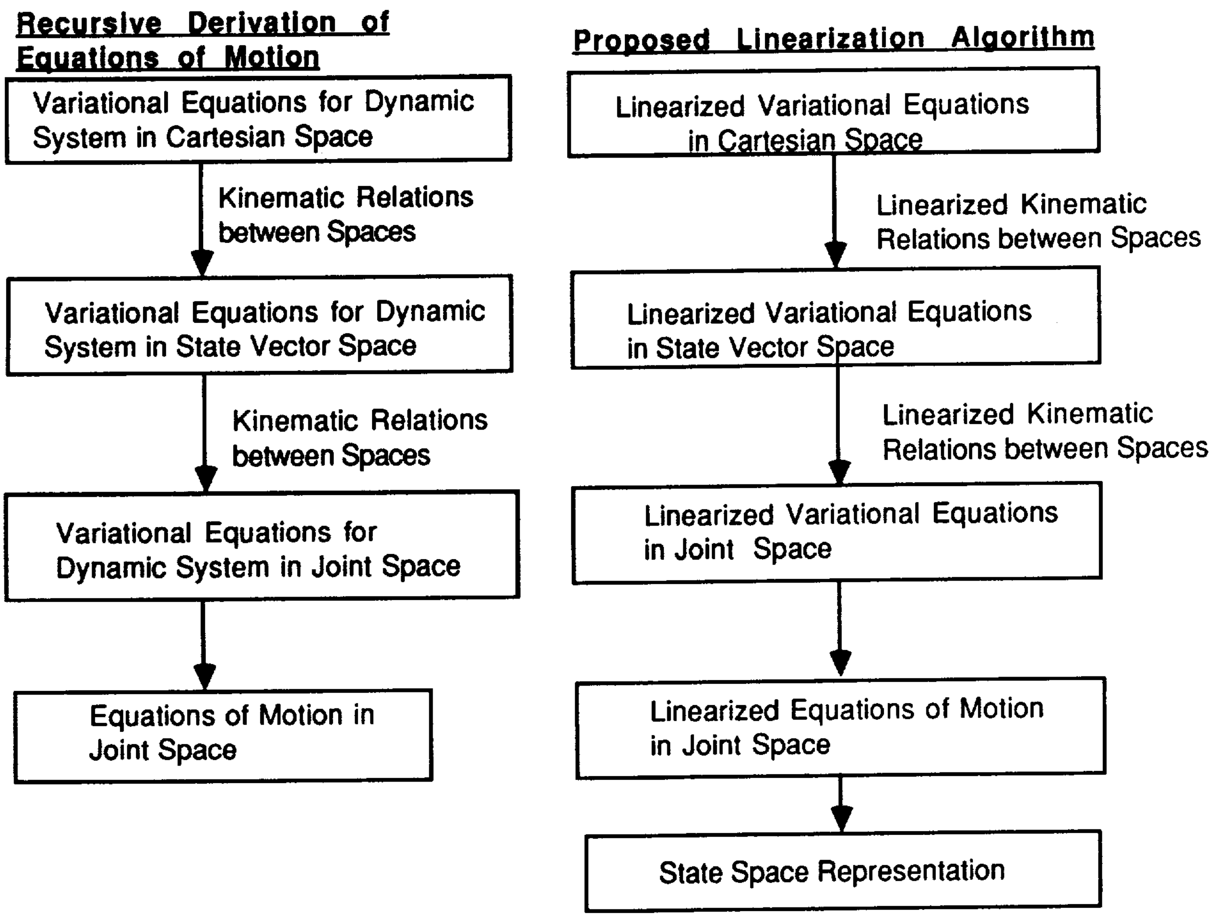

Figure 1. Recursive Derivation Flow for Dynamic Equations of Motion and Proposed Linearization Algorithm

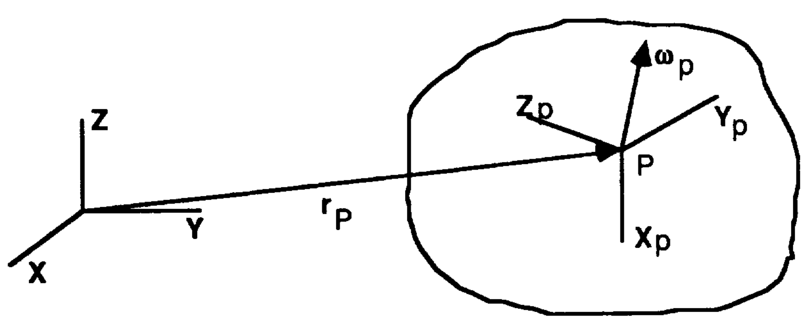

Figure 2. Body and Global Frames Representations

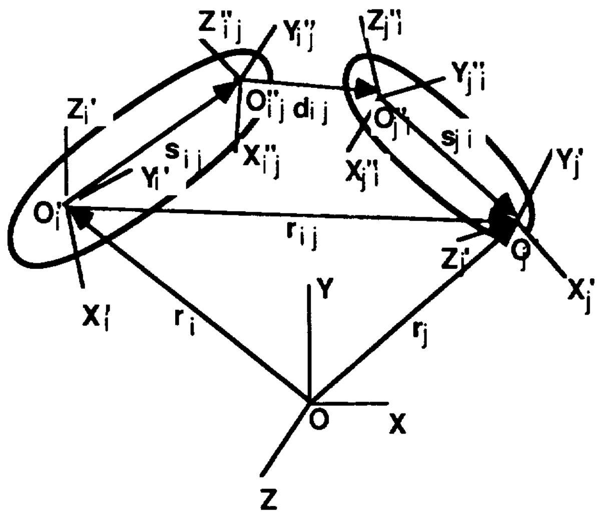

Figure 3. Pair of Contiguous Bodies

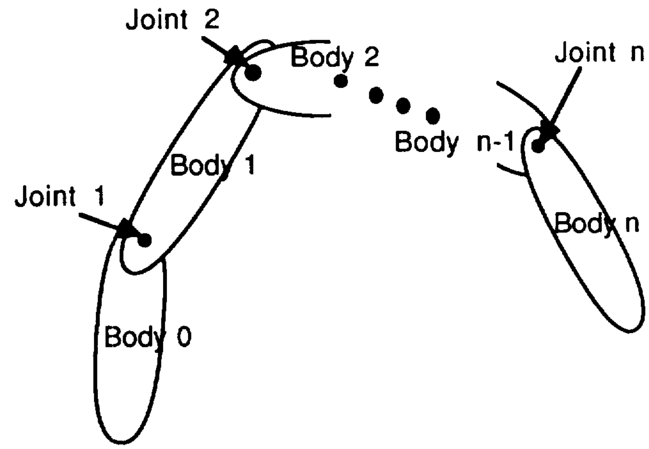

Figure 4. An n-Joint Open -Chain Mechanism

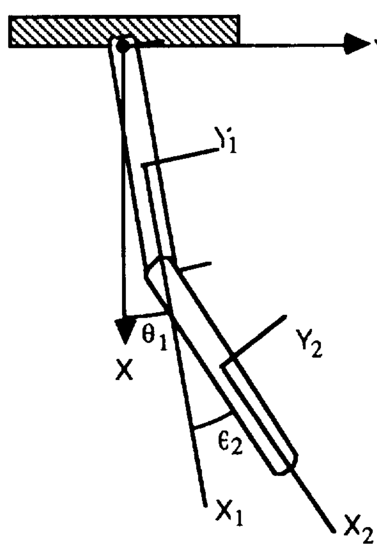

Figure 5. A Two Links Manipulator Arm

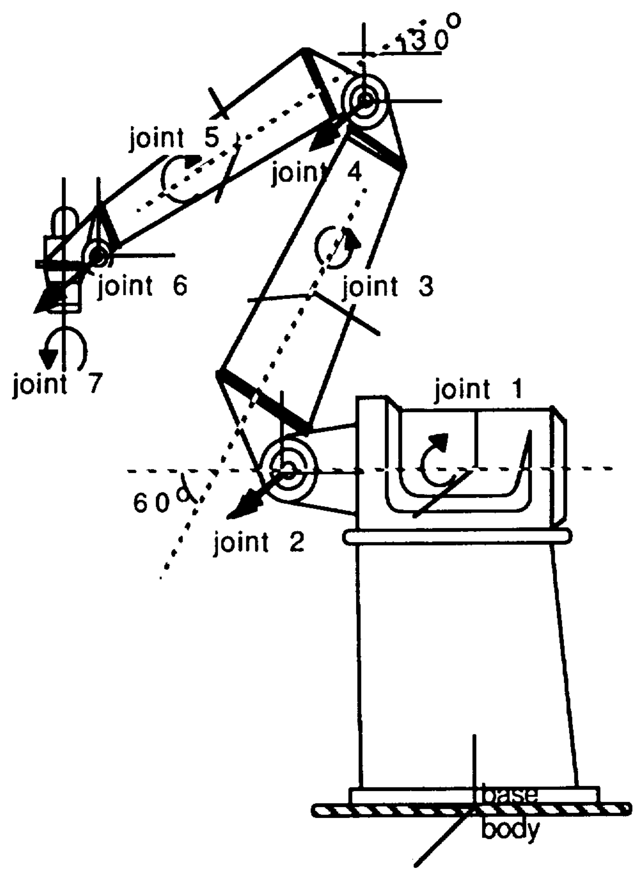

Figure 10. A Seven Degrees of Freedom Robot

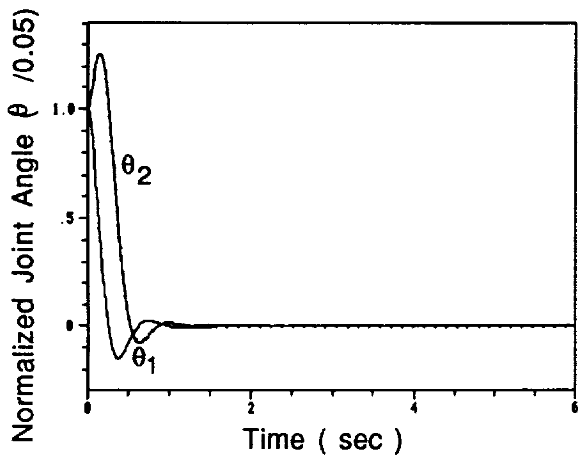

Figure 6. Response of small Initial Deviation

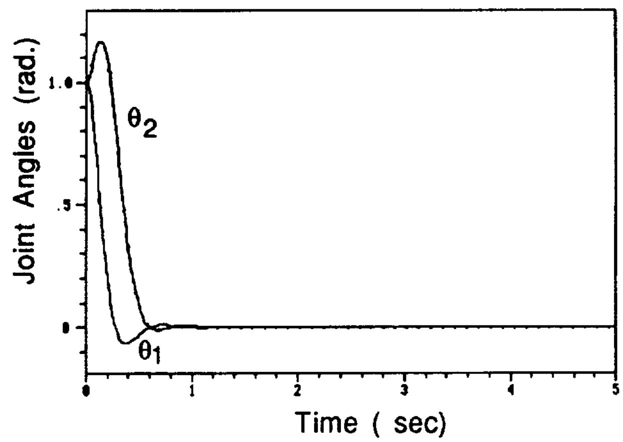

Figure 7. Response of large Initial Deviation

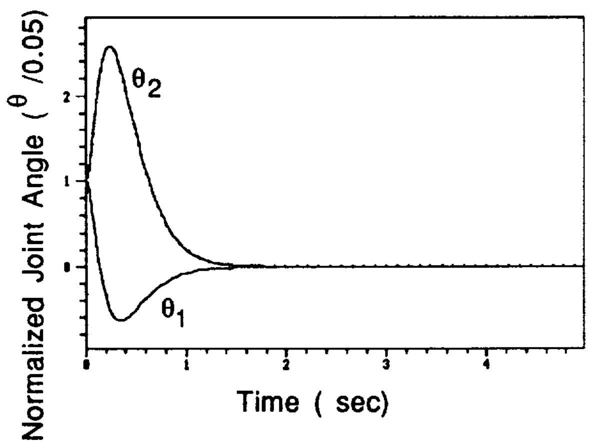

Figure 8. Response of small Initial Deviation

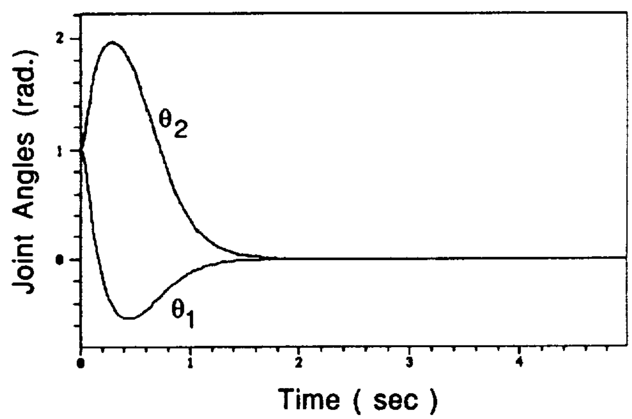

Figure 9. Response of large Initial Deviation

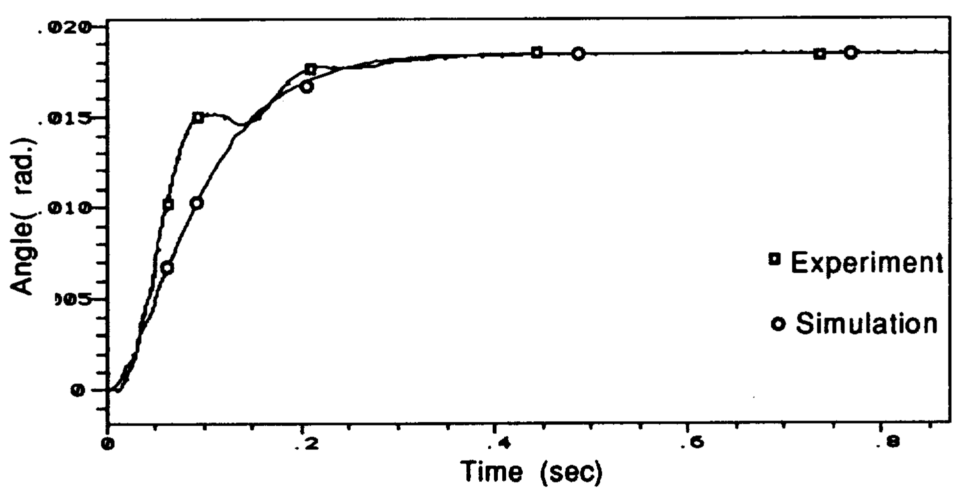

Figure 11. Step Response of Joint 4