565227 p15
N90-23017

Development of Efficient Computer Program for Dynamic Simulation of Telerobotic Manipulation

J. Chen & Y. J. Ou

Department of Mechanical Engineering
University of Maryland
College Park, MD 20742

Abstract

Research in robot control has generated interests in computationally efficient forms of dynamic equations for multi-body systems. For a simply connected open-loop linkage, dynamic equations arranged in recursive form has been found to be particularly efficient. A general computer program capable of simulating open-loop manipulator with arbitrary number of links has been developed based on an efficient recursive form of Kane's dynamic equations. Also included in the program is some of the important dynamics of the joint drive system, i.e., the rotational effect of the motor rotors. Further efficiency is achieved by the use of symbolic manipulation program to generate the Fortran simulation program tailored for a specific manipulator based on the parameter values given. This paper describes the formulations and the validation of the program, and it also shows some results.

Introduction

In the development of a robotic manipulator, simulation program can be an important design tool. It can be used to support detailed mechanical design by revealing the constraint forces and torques at different locations during certain maneuvers. It can also be applied to test different control laws without concerns of damaging the actual manipulators. If real time simulation can be developed, training of telerobot operators and testing of actual control hardware and software can become possible.

The success of a simulation in providing the useful and accurate information depends on the model fidelity, the formulation of equations of motion and the numerical solution of the equations. There is no such thing as "the" simulation of a dynamical system because the fidelity of the model determines what the results are like. There is always room for higher fidelity and so there is no end to it. But quite often, a modest increase of model fidelity is accompanied by a significant increase in equation complexity and numerical difficulty, and thus computation time. To achieve reasonable efficiency in the computation, one has to investigate the merits of different solution algorithms, different dynamical formulations and different levels of model fidelity. Additionally, one has to validate that the program is correctly representing the model.

Many researchers have worked on efficient formulations of dynamic equations for robot manipulators[2-9]. Most of them model robot as consisting of rigid bodies connected together with revolute or translational joints. Details of the joint drive systems have been mostly ignored. It is shown in [7] that joint drive systems have potentially significant effects on robot dynamics and hence should be included in the model. Also shown in [7] is a procedure to obtain the dynamical equations of a robot with a speed-reduction drive system from the equations of a direct drive robot. This procedure will be followed to develop a more comprehensive robot simulation program.

It has been known [2,5,6] that the important aspect of efficient formulations is the recursive development of kinematic and dynamic quantities to reduce the number of transformations among vector bases. It is also known that recursive Lagrange's formulation is still less efficient than the recursive Newton-Euler's formulations. However, Newton-Euler's formulation will not be advantageous if more complicated model of the system is analyzed. Since the program under development is anticipated to be expanded for more comprehensive modeling of manipulator systems, Kane's method is chosen because of its systematic features. An efficient formulation has been developed by applying recursive schemes in Kane's equations for a general manipulator system. The forward and backward recursions are established based on the bounds on the summation signs in the equations.

If properly developed, it is expected that a customized simulation program for a particular manipulator should be more efficient than a general purpose simulation program. For the development of simulation program, there is always a trade-off between generality and efficiency. But through the application of symbolic manipulation to eliminate unnecessary computations that occur for a particular model, it is possible to improve simultaneously the generality and the efficiency of a simulation program. Symbolic manipulation language MACSYMA has been used to develop a program called MSP (Manipulator Simulation Program) for manipulators that are made up of a single chain of any number of rigid bodies connected by revolute joints. Gear reduction effects of some simple joint drive systems are also efficiently incorporated in the program following the procedure in [7].

Independent formulation and programming of the system kinetic energy and the system angular momentum about a base-fixed point on the 1st joint axis are developed for validation purposes. Test cases which involve conservation of these quantities have been selected to validate the simulation programs. The objective of this paper is to present the formulation involved in the development of this program. Computation efficiency and significance of gear reduction effect are also to be discussed.

Mathematic Model

An open chain manipulator with N degrees of freedom as shown in Fig. 1 is analyzed for the development of MSP. Each link is driven with a motor and a gear reduction mechanism, an example of which is shown in Fig. 2. The base is considered fixed in the earth E (assumed to be an inertial reference frame). Couples are generated at motors through electromagnetic interactions, and gear reductions amplify the resulted moments on the links about the joints. It is assumed that the motor rotor and its rigidly attached part is the only massive element in a joint drive system that will contribute to the modifications of the equations of motion from that of a multibody direct drive system.

The links are labeled consecutively B_1 to B_N starting from the link connected to the base. The base is referred to as link B_0. The ideal revolute joints between links are numbered such that joint i connects link B_i to link B_{i-1}. An orthogonal unit vector basis $\underline{x}_i$, $\underline{y}_i$ and $\underline{z}_i$ fixed in B_i is defined in such a way that the unit vector $\underline{z}_i$ (i = 1,..., N) is directed along the axis of joint i. A particular configuration called the null configuration of a manipulator is one in which relative joint angles between links are all equal to zeros. The joint angles q_i (i = 1,..., N) are positive when right-handed rotation from the null configuration about $\underline{z}_i$ occurs. In this paper, the motor driving link B_i is assumed to be mounted on link B_{i-1} and unit vector $\underline{e}_i$ is defined to be parallel to the rotation axis of the motor rotor.

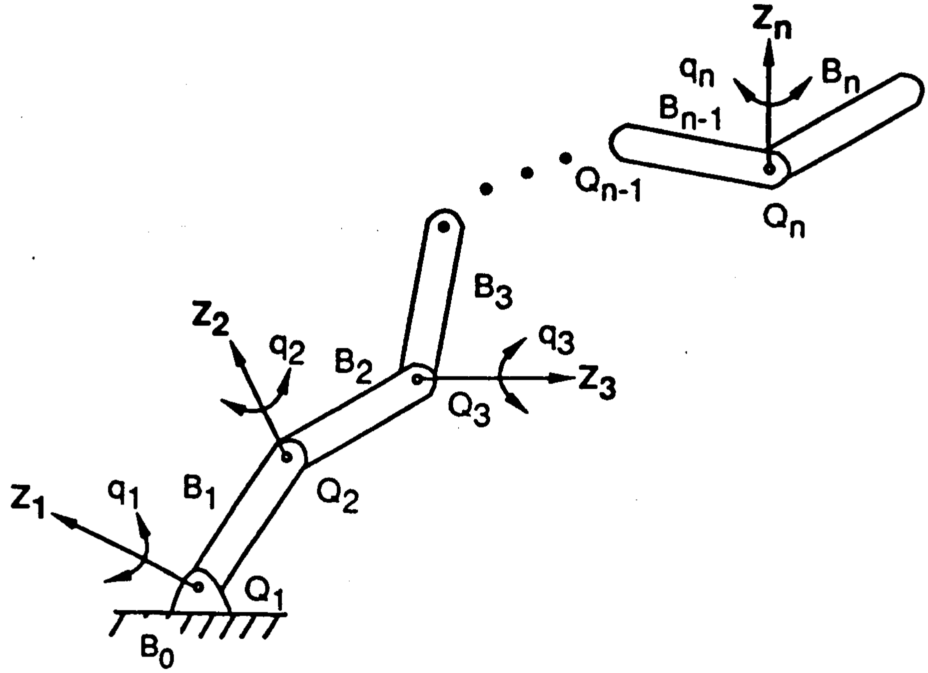


Fig. 1. Multi-Link System

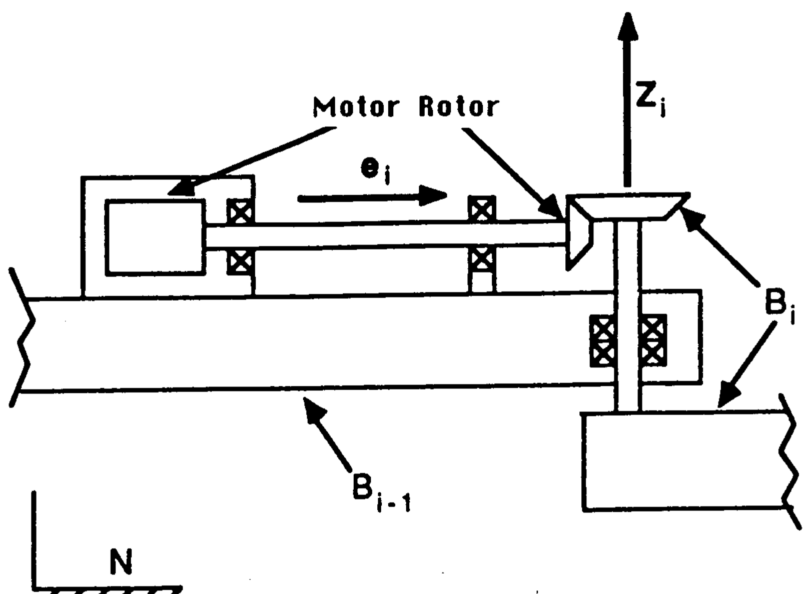


Fig. 2. Motor Rotor and Gear Reduction Mechanism

Formulation of Dynamical Equation

The following presentation of the formulations will be in terms of vectors and dyadics which are quantities independent of unit vector bases and can be represented by column and square matrices, respectively, when expressed in a particular basis.

Kane's dynamical equations[1] are

$$F_r + F_r^* = 0 \qquad (r = 1,\dots,N) \quad (1)$$

where F_r and F_r^* are the generalized active and inertia forces associated with the r-th generalized speed, respectively. Since the system is holonomic, the number of generalized speeds are equal to the number of degrees of freedom. Here the generalized speeds are chosen to be simply the derivatives of the generalized coordinates. Assuming that the only contributing active forces are the motor torques $T_r\underline{e}_r$, the gravitational forces and the external load on the last link, represented by a force $\underline{F}^e$ acting through the mass center and a torque $\underline{T}^e$, one can write

$$F_r = \mu_r T_r + \sum_{i=1} M_i(\underline{V}_{i,r}\cdot \underline{G}) + \underline{V}_{N,r}\cdot \underline{F}^e + \underline{\omega}_{N,r}\cdot \underline{T}^e \qquad (r = 1,\dots,N) \quad (2)$$

where μ_r is the gear ratio for the r-th joint, M_i is the mass of i-th link, $\underline{G}$ is the gravitational acceleration vector, and $\underline{V}_{i,r}$ and $\underline{\omega}_{i,r}$ are the r-th partial velocity of B_i^*, mass center of B_i, and the r-th partial angular velocity of B_i, in E, respectively. The generalized inertia forces are due to the inertia force and torque associated with each link, and they are

$$F_r^* = -\sum_{i=1}^{N} M_i(\underline{V}_{i,r}\cdot \underline{a}_i) - \sum_{i=1}^{N} \underline{\omega}_{i,r}\cdot(\hat{\underline{I}}_i\cdot \underline{\alpha}_i + \underline{\omega}_i\times \hat{\underline{I}}_i\cdot \underline{\omega}_i) \qquad (r = 1,\dots,N) \quad (3)$$

where $\underline{a}_i$ and $\underline{\alpha}_i$ are the acceleration of B_i^* and the angular acceleration of B_i in E, respectively, and $\hat{\underline{I}}_i$ is the central inertia dyadic of B_i. Therefore, the dynamic equations become

$$\sum_{i=1}^{N} M_i(\underline{V}_{i,r}\cdot \underline{a}_i) + \sum_{i=1}^{N} \underline{\omega}_{i,r}\cdot(\hat{\underline{I}}_i\cdot \underline{\alpha}_i + \underline{\omega}_i\times \hat{\underline{I}}_i\cdot \underline{\omega}_i)$$

$$= \mu_r T_r + \sum_{i=1}^{N} M_i(\underline{G}\cdot \underline{V}_{i,r}) + \underline{V}_{N,r}\cdot \underline{F}^e + \underline{\omega}_{N,r}\cdot \underline{T}^e \qquad (r = 1,\dots,N) \quad (4)$$

From Fig. 1, we can obtain the following kinematic equations.

$$\underline{\omega}_i = \sum_{j=1}^{i} \dot{q}_j\, \underline{z}_j \qquad (i = 1,\dots,N) \quad (5)$$

$$\underline{V}_i = \sum_{j=1}^{i} \dot{q}_j(\underline{z}_j\times \underline{r}_{ji}^c) \qquad (i = 1,\dots,N) \quad (6)$$

$$\underline{\omega}_{i,r} = \underline{z}_r\, \xi_r^i \qquad (i = 1,\dots,N) \quad (7)$$

$$\underline{V}_{i,r} = (\underline{z}_r\times \underline{r}_{ri}^c)\, \xi_r^i \qquad (i = 1,\dots,N) \quad (8)$$

where $\underline{r}_{ji}^c$ is the position vector from Q_j to B_i^* and

$$\xi_r^i = \begin{cases} 1 & \text{if } i \ge r \\ 0 & \text{if } r > i \end{cases} \qquad (i,r = 1,\dots,N) \quad (9)$$

The angular and linear accelerations can be derived as

$$\underline{\alpha}_i = \sum_{j=1}^{i} \ddot{q}_j\, \underline{z}_j + \underline{\alpha}'_i \qquad (i = 1,\dots,N) \quad (10)$$

$$\underline{a}_i = \sum_{j=1}^{i} \ddot{q}_j\, (\underline{z}_j\times \underline{r}_{ji}^c) + \underline{a}'_i \qquad (i = 1,\dots,N) \quad (11)$$

C-4

where

$$\underline{\alpha}'_i = \sum_{j=1}^{i} \dot{q}_j(\underline{\omega}_j \times \underline{z}_j) \tag{12}$$

$$\begin{aligned}\underline{a}'_i &= \sum_{j=1}^{i} \dot{q}_j \frac{{}^E d}{dt} (\underline{z}_j \times \underline{r}^c_{ji}) \\ &= \underline{\alpha}'_i \times \underline{r}^c_{ji} + \underline{\omega}_i \times(\underline{\omega}_i \times \underline{r}^c_{ii}) + \sum_{j=1}^{i-1} \dot{q}_j \underline{z}_j \times (\frac{{}^E d}{dt} \underline{r}^L_{ji})\end{aligned} \tag{13}$$

and $\underline{r}^L_{ji}$ is the position vector from Q_j to Q_i. A left superscript on a time differentiation symbol represents the reference frame in which the differentiation is to be performed[1]. With equations (5-13) substituted, the equations of motion can be rewritten as

$$\sum_{i=1}^{N} A'_{ri} \ddot{q}_i = \mu_r T_r - \hat{T}_r \qquad (r = 1, \ldots, N) \tag{14}$$

where

$$A'_{ri} = \underline{z}_r \cdot \underline{f}_{ri} \tag{15}$$

$$\underline{f}_{ri} = \sum_{j=i}^{N} [\ \hat{\underline{\underline{I}}}_j \cdot \underline{z}_i + M_j\ \underline{r}^c_{rj} \times (\underline{z}_i \times \underline{r}^c_{ij})\ ] \qquad (i \geq r) \tag{16}$$

$$\hat{T}_r = \underline{z}_r \cdot \hat{\underline{T}}_r \tag{17}$$

$$\hat{\underline{T}}_r = \{ \sum_{i=r}^{N} [M_i (\underline{r}^c_{ri} \times \hat{\underline{a}}_i) + \underline{T}'_i]\} - \underline{T}^e - (\underline{r}^c_{rN} \times \underline{F}^e) \tag{18}$$

$$\hat{\underline{a}}_i = \underline{a}'_i - \underline{G} \tag{19}$$

$$\underline{T}'_i = \hat{\underline{\underline{I}}}_i \cdot \underline{\alpha}'_i + \underline{\omega}_i \times \hat{\underline{\underline{I}}}_i \cdot \underline{\omega}_i \tag{20}$$

Because of symmetry, only upper triangular terms of matrix $[A'_{ri}]$ need to be evaluated. The following formulations are used to evaluate the vector quantity $\underline{f}_{ri}$.

$$\underline{f}_{ii} = \underline{\underline{I}}_i \cdot \underline{z}_i \qquad (i = 1, \ldots, N) \tag{21}$$

$$\begin{aligned}\underline{\underline{I}}_i &= \sum_{j=i}^{N} \{\ \hat{\underline{\underline{I}}}_j + M_j[(\underline{r}^c_{ij})^2 \underline{\underline{U}} - \underline{r}^c_{ij}\ \underline{r}^c_{ij}\ ]\} \\ &= \underline{\underline{I}}_{i+1} + \underline{\underline{K}}_i + 2(\underline{r}^L_{i(i+1)} \cdot \underline{r}^*_{i+1})\underline{\underline{U}} - \underline{r}^L_{i(i+1)}\underline{r}^*_{i+1} - \underline{r}^*_{i+1}\underline{r}^L_{i(i+1)}\end{aligned}$$

$$(i = N-1, \ldots, 1) \tag{22}$$

$$\underline{\underline{K}}_i = \hat{\underline{\underline{I}}}_i + M_i[(\underline{r}^c_{ii})^2 \underline{\underline{U}} - \underline{r}^c_{ii}\ \underline{r}^c_{ii}] + (\sum_{j=i+1}^{N} M_j)[(\underline{r}^L_{i(i+1)})^2 \underline{\underline{U}} - \underline{r}^L_{i(i+1)}\underline{r}^L_{i(i+1)}]$$

$$(i = 1, \ldots, N) \tag{23}$$

$$\begin{aligned}\underline{r}^*_i &= \sum_{j=i}^{N} M_j\ \underline{r}^c_{ij} \\ &= \underline{r}^*_{i+1} + M_i\ \underline{r}^c_{ii} + (\sum_{j=i+1}^{N} M_j)\ \underline{r}^L_{i(i+1)} \qquad (i = N-1, \ldots, 2)\end{aligned} \tag{24}$$

The dyadic quantity $\underline{\underline{K}}_i$ is a constant in B_i, i.e, if $\underline{\underline{K}}_i$ is expressed in terms of $\underline{x}_i$, $\underline{y}_i$, $\underline{z}_i$ basis, the coefficients are constants. Equations (22) and (24) are backward recursive formulas that can be evaluated by establishing the following:

$$\begin{aligned}\underline{\underline{I}}_N &= \hat{\underline{\underline{I}}}_N + M_N[(\underline{r}^c_{NN})^2 \underline{\underline{U}} - \underline{r}^c_{NN}\ \underline{r}^c_{NN}\ ] \\ &= \underline{\underline{I}}^{B_N/Q_N}\end{aligned} \tag{25}$$

$$\underline{r}^*_N = M_N\ \underline{r}^c_{NN} \tag{26}$$

where $\underline{\underline{I}}^{B_N/Q_N}$ is the inertia dyadic of B_N relative to Q_N. For the off-diagonal terms of $[A_{ri}]$, a backward recursive formula involving r, i.e.,

$$\underline{f}_{ri} = \underline{f}_{(r+1)i} + \underline{r}^{L}_{r(r+1)} \times (\underline{z}_i \times \underline{r}^{*}_i) \qquad \left(\begin{matrix} i = N, \ldots, 2 \\ r = i-1, \ldots, 1 \end{matrix}\right) \qquad (27)$$

can be used with $\underline{f}_{ii}$ being the starting vector that should have been evaluated from equation (21).

For $\hat{\underline{T}}_r$, the following forward recursive formulations can be used to evaluate the necessary quantities.

$$\underline{\omega}_i = \underline{\omega}_{i-1} + \dot{q}_i \underline{z}_i \qquad (i = 1, \ldots, N) \qquad (28)$$

$$\underline{\alpha}'_i = \underline{\alpha}'_{i-1} + \dot{q}_i (\underline{\omega}_i \times \underline{z}_i) \qquad (i = 1, \ldots, N) \qquad (29)$$

$$\underline{u}_i = \hat{\underline{a}}_i - \underline{\underline{A}}_i \cdot \underline{r}^{c}_{ii} = \underline{u}_{i-1} + \underline{\underline{A}}_{i-1} \cdot \underline{r}^{L}_{(i-1)i} \qquad (i = 1, \ldots, N) \qquad (30)$$

where

$$\underline{\underline{A}}_i = \underline{\alpha}'_i \times \underline{\underline{U}} + (\underline{\omega}_i \times \underline{\underline{U}}) \cdot (\underline{\omega}_i \times \underline{\underline{U}}) \qquad (i = 1, \ldots, N) \qquad (31)$$

The starting values for equations (28-30) are

$$\underline{\omega}_0 = 0 \qquad (32)$$

$$\underline{\alpha}'_0 = 0 \qquad (33)$$

$$\underline{u}_0 = -\underline{G} \qquad (34)$$

The introduction of dyadic quantity $\underline{\underline{A}}_i$ is to reduce the overall computation by reusing it in two equations in the remaining formulations. The dyadic obtained by cross multiplying a vector with a unit dyadic can be represented by a skewsymmetric square matrix when it is expressed in a particular unit vector basis. This skew symmetric square matrix is commonly encountered when a cross multiplication of column matrices is replaced with a matrix multiplication.

The following backward recursive formulation can be used to evaluate the kinetic quantities $\hat{\underline{T}}_i$:

$$\underline{p}_i = \underline{p}_{i+1} + \underline{\underline{A}}_i \cdot \left[\left(\sum_{j=i+1}^{N} M_j \right) \underline{r}^{L}_{i(i+1)} + M_i \, \underline{r}^{c}_{ii} \right] \qquad (i = N-1, \ldots, 1) \qquad (36)$$

$$\hat{\underline{T}}_i = \hat{\underline{T}}_{i+1} + \underline{r}^{L}_{i(i+1)} \times \underline{p}_{i+1} + \underline{L}_i + \left[M_i \, \underline{r}^{c}_{ii} + \left(\sum_{j=i+1}^{N} M_j \right) \underline{r}^{L}_{i(i+1)} \right] \times \underline{u}_i \qquad (i = N-1, \ldots, 1) \qquad (37)$$

where

$$\underline{L}_i = \left\{ \underline{\underline{A}}_i \cdot \left[\underline{\underline{K}}_i - \frac{1}{2} (\underline{\underline{K}}_i : \underline{\underline{U}}) \, \underline{\underline{U}} \right] \right\}_v \qquad (i = 1, \ldots, N) \qquad (38)$$

and a subscript v next to a dyadic in equation (38) denotes the vector of the dyadic, which is formed by summing the cross products of the prefactors and the postfactors of all the dyads in the dyadic. The reason for using the expression in equation (38) is to reduce computation counts. In fact, $\underline{L}_i$ can be expressed in a form identical to equation (20). Conversely, equation (20) can be replaced by

$$\underline{T}'_i = \left\{ \underline{\underline{A}}_i \cdot \left[\hat{\underline{\underline{I}}}_i - \frac{1}{2} (\hat{\underline{\underline{I}}}_i : \underline{\underline{U}}) \, \underline{\underline{U}} \right] \right\}_v \qquad (i = 1, \ldots, N) \qquad (39)$$

where the expression in brackets [ ] is the dyadic whose representation in a particular unit vector basis is the inertia matrix with half its trace subtracted from each diagonal element. The dyadic operations used above follow the convention introduced by Gibbs[10] in late eighteen hundreds.

The starting values for the recursive equations, equations (36) and (37), are

$$\underline{p}_N = M_N \, \underline{\underline{A}}_N \cdot \underline{r}^{c}_{NN} \qquad (40)$$

$$\hat{\underline{T}}_N = M_N \, \underline{r}^{c}_{NN} \times (\underline{u}_N - \underline{F}^{e}/M_N) + \underline{L}_N - \underline{T}^{e} \qquad (41)$$

Equations (14) represent the equations of motion of a direct drive open-chain system. The modifications to equations (14) for an open-chain system with motors and gear reduction mechanisms shown in Fig. 2 are based on the difference between generalized inertia forces contributing to these two systems. They are[7]

$$(F_r^*)_s = (F_r^*)_{s'} + G_r^* \qquad (r = 1,\dots,N) \qquad (42)$$

where subscript s represents a manipulator system S which has a motor and gear reduction mechanism in each link similar to that shown in Fig. 2, while subscript s' represents the manipulator system S' which has the same mass and inertia distribution as system S, but the motor rotors and gear reduction mechanisms are considered to be fixed in the links on which they are mounted. Here and throughout this paper, the motor driving i-th joint is assumed to be mounted on link B_{i-1} (i=1,...,N). System S', therefore, represents a direct-drive system. The difference terms between these two systems are

$$G_r^* = \begin{cases} 0 & (r > i) \\ -\mu_r J_r(\mu_r \ddot{q}_r + \underline{\alpha}_{r-1} \cdot \underline{e}_r) & (r = i) \\ \sum_{i=r+1}^{N} -\mu_i J_i[(\underline{z}_r \cdot \underline{e}_i)\ddot{q}_i - \dot{q}_i(\underline{\omega}_{i-1} \times \underline{z}_r) \cdot \underline{e}_i] & (r < i) \end{cases} \qquad (43)$$

With the additional terms added, the dynamic equations for S become

$$\sum_{i=1}^{N} A_{ri}\, \ddot{q}_i = \mu_r T_r - \hat{T}_r + G_r \qquad (r = 1,\dots,N) \qquad (44)$$

where

$$A_{ri} = A'_{ri} + \begin{cases} \mu_i J_i(\underline{z}_r \cdot \underline{e}_i) & (\text{if } r \neq i) \\ \mu_i^2 J_i & (\text{if } r = i) \end{cases} \qquad (45)$$

$$G_r = -\mu_r J_r\, \underline{\alpha}'_{r-1} \cdot \underline{e}_r + \sum_{i=r+1}^{N} \mu_i J_i \dot{q}_i (\underline{\omega}_{i-1} \times \underline{z}_r) \cdot \underline{e}_i \qquad (46)$$

It can be noticed that matrix $[A_{ri}]$ is still symmetric. Hence, only those difference terms in the upper triangle of matrix $[A_{ri}]$ need to be evaluated.

Symbolic Manipulation

Direct numerical approach in evaluating the above equations can be inefficient if there are terms involving multiplication with 0 or 1, or addition with 0. The use of multi-dimensional arrays in a general purpose program further reduces the computational efficiency. These are some of the reasons why a general simulation program cannot achieve the highest possible efficiency. Symbolic language such as MACSYMA can be applied to eliminate these inefficiencies. Theoretically, one can use MACSYMA to derive equations explicitly in terms of all joint angles and their derivatives and then to reorganize the equations for efficient computation. But for a manipulator with high number of degrees of freedom, this requires enormous memory space and CPU time, and cannot be optimally simplified because the simplification is limited by the capability of MACSYMA. It has been found that the recursive formulation as presented above is particularly advantageous because computation is already optimized. Only simple symbolic operations need to be applied to generate the recursive equations of motion in FORTRAN coding, and hence the computer time required for this process is not excessively long.

Program Validation

Complete validation of a simulation program is next to impossible. But without being subject to some forms of validation, a program cannot be trusted. When a program is applied on a reasonably complicated system, one cannot rely on simple statements like "the results make sense" as validation.

For a manipulator system with 3 or more links, intuition as to its motion when subject to certain inputs does not work well at all. A more systematic approach need to be adopted. This is a very important subject for the dynamic simulation of robots, but it does not seem to attract much attention. The approach taken by the authors was to select some conditions under which the system will have some scalar quantities, such as energy or measure numbers of an angular momentum vector, that are conserved throughout the motion. Since there exists a slight possibility that chance may cause errors not to be detected, an independently developed program is used to evaluate the validation quantities. Formulation of the system kinetic energy, potential energy and the system angular momentum about a base-fixed point on the first joint axis for validation purposes will be described next followed by the description of test cases chosen.

The kinetic energy formulation for a direct-drive manipulator S' is

$$K' = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} \dot{q}_i \dot{q}_j \tag{47}$$

For a manipulator with gear reduction in its drive system, slight modification is required. Consider two manipulator system S and S' as described before. If they have the same motion, then the kinetic energies K and K' of systems S and S', respectively, are related by[7]

$$K = K' + \sum_{i=1}^{N} \left(\frac{1}{2} \mu_i^2 J_i \dot{q}_i^2 + \mu_i J_i \dot{q}_i \, \underline{\omega}_{i-1} \cdot \underline{e}_i \right) \tag{48}$$

The potential energy V of the system are due to gravity only, and it is

$$V = -\sum_{i=1}^{N} M_i \, \underline{G} \cdot \underline{r}_{1i}^{c} \tag{49}$$

There is no difference between the potential energy expressions for S and S' because they have the same mass distribution. The angular momentum vector $\underline{H}'$ of a direct-drive manipulator S' about Q_1 is

$$\underline{H}' = \sum_{j=1}^{N} \sum_{B_j} \underline{r}_1^P \times (m_P \, {}^E\underline{v}^P) \tag{50}$$

where P is a generic particle in B_j, $\sum_{B_j}$ represent the summing over all the particles in B_j, $\underline{r}_1^P$ is the position vector from Q_1 to P, and m_P and ${}^E\underline{v}^P$ are, respectively, the mass and the velocity in E of P. Only the measure number of $\underline{H}'$ in $\underline{z}_1$ direction is used in the validation process. Comparing the kinetic energy formulation with that of $\underline{H}'$, one can obtain

$$\underline{H}' \cdot \underline{z}_1 = \sum_{i=1}^{N} A_{1i} \dot{q}_i \tag{51}$$

where A_{1i} can be found in Eq. (15) with r = 1. The angular momenta $\underline{H}$ and $\underline{H}'$ about Q_1 of systems S and S', respectively, are related by

$$\underline{H} = \underline{H}' + \sum_{i=1}^{N} \mu_i J_i \dot{q}_i \underline{e}_i \tag{52}$$

Hence, equations (48-52) provide the necessary formulas for the evaluation of the conservation quantities.

The validation cases used are

I. Conservation of total energy, K+V :

$T_r = 0 \quad (r = 1, \ldots, N)$

II. Conservation of $\underline{H} \cdot \underline{z}_1$:

a. $g = 0$, $T_1 = 0$, $\mu_1 = 1$, $\underline{e}_1 = \underline{z}_1$

b. $g \neq 0$, $T_1 = 0$, $\mu_1 = 1$, $\underline{e}_1 = \underline{z}_1 = \pm \underline{G}/g$

where g is the gravitational constant.

Under conditions described above, many simulation runs of manipulators have been performed to validate the program. The results of these runs show that numerical variations of the conservation quantities are of the order of magnitude that is appropriately correlated to the absolute integration error tolerances. Here integration subroutines using Adams-Bashforth-Moulton predictor-corrector scheme and using Runge-Kutta-Verner method have been separately applied for numerical integration.

Discussions

For a general manipulator, the total numbers of operations to obtain A_{ri} and $\mu_r T_r - \hat{T}_r$ (r,i = 1,...,N) in Eq. (14) for a direct drive manipulator are 11N(N-1)+150(N-1)-15 multiplications and 7N(N-1)+119(N-1)-14 additions. Table 1 lists the numbers of operations required for each of the equations in the evaluation of matrix $[A_{ri}]$ and $\hat{T}_r$. The counting of operations follows that presented in [6]. Notice that unit vector basis transformation has to be performed in each recursion step. Here external force and torque applied on the last link are not included. Therefore, for a general 6 link manipulator, 1075 multiplications and 791 additions are required. This is much lower than the 1541 multiplications and 1196 additions needed in Method 3 of [2]. For the six dof PUMA 600 presented in [9], Our MSP program generates FORTRAN code that requires 351 multiplications and 281 additions to perform the computation that takes 392 multiplications and 294 additions in [8].

Table 1. Number of Operations

equations	multiplications	additions
(22)	57(N-1)-33	36(N-1)-25
(24)	8(N-1)	7(N-1)-3
(27)	11N(N-1)-8(N-1)	7N(N-1)-4(N-1)
(28)	8(N-1)	5(N-1)
(29)	10(N-1)	6(N-1)
(30)	17(N-1)	13(N-1)
(31)	6(N-1)	9(N-1)+1
(36)	17(N-1)	13(N-1)
(37)	20(N-1)	19(N-1)
(38)	15(N-1)+3	15(N-1)+1
(40)	9	6
(41)	6	5
Total	11N(N-1)+150(N-1)-15	7N(N-1)+119(N-1)-14

Adding $\ddot{q}_i \underline{z}_i$ to the right hand side of equation (29), equations (28-41) together with equation (17) become inverse dynamic formulation for a direct drive robot. This inverse dynamic evaluation is similar to algorithm 3 in [6]. By applying MACSYMA to the formula, some unnecessary computations can be removed. For instance, if N is 6, the number of computation for the manipulator with twist angles equal to 0° or 90° is 340 multiplications and 290 additions compared to 388 multiplications and 370 additions in [6]. For the simpler manipulator with $\underline{r}^c_{ii}$ and $\underline{r}^L_{i(i+1)}$ having only one nonzero element, the numbers of computation are 245 multiplications and 204 additions compared to 277 multiplications and 255 additions in [6]. The reductions are due to some additional multiplications and additions with zero quantities that are counted in [6] because the authors of [6] did not actually expand the equations for the counting.

In order to give additional indication of efficiency, the 7 link Robot Research Corporation [11] manipulator shown in Fig. 3 with parameters listed in the Appendix is considered. The numbers of operations for the system in the form of Eq. (14) are 651 multiplications and 505 additions with effects of motor rotors included. The numbers for direct-drive system are 548 multiplications and 439 additions. Therefore, adding the effects of motor rotors requires 103 multiplications and 66 additions, which is about 17% of the total computation needed for the direct-drive system.

In the interest of demonstrating the effect of motor rotors, a constant motor torque (T_1= 0.625 N-m) is applied on the first joint of the 7 link manipulator shown in Fig. 3. Two sets of equations are solved for comparison. Set 1 is equation (14), which represents the direct drive system S' and set 2 is equation (44), which is the complete equations of system S. The results from set 1 are shown in Figs. 4 and 5 while those from set 2 are in Figs. 6 and 7. The differences of the results are so substantial that it is clear that set 1 is not representative of the actual system.

Among all the additional terms due to motor rotor, the terms $\mu_i^2 J_i \ddot{q}_i$ (i=1,..., N) are most significant due to the large values of μ_i. Another set of equations, set 3, established by adding only $\mu_i^2 J_i$ to diagonal elements A_{ii} of $[A'_{ri}]$ matrix in equations (14), is also solved for comparison. A sinusoidal motor torque (T_1= 3.125 cos 0.8Πt kg-m) is applied on the first joint of the manipulator. Some of the results from set 2 are shown in Fig. 8 while the differences of the results between set 2 and set 3 are shown in Fig. 9. It is clear that the differences are relatively small in the duration of 5 seconds.

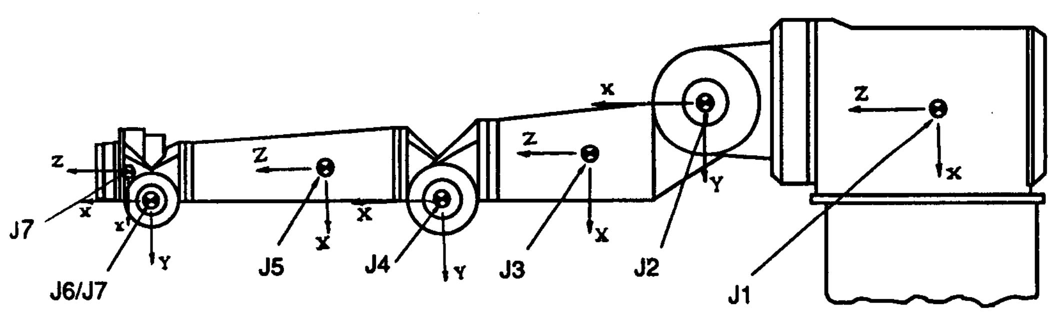


Fig. 3. A 7 Link Manipulator

Conclusions

The initial development of an efficient simulation program for telerobotic manipulators is described. An efficient recursive formulation of Kane's dynamic equations for a class of manipulators which are made up of links connected with revolute joints and driven by motors with reduction mechanisms is presented. The recursions are established according to the summation bounds in the equations. Comparison of operation counts with other published formalisms shows advantages of the present approach. Furthermore, effects of rotor inertia and speed reduction are included in the formulation to yield a more faithful model of the actual system. Symbolic manipulation is also applied to generate customized simulation program for additional improvement of the computational efficiency. Aside from the discussions on efficiency, steps taken to validate the simulation program are presented. Finally, simulation results show that effects of rotors in drive system with high speed reduction cannot be ignored in the simulation of a manipulator.

Acknowledgement

The funding support from NASA Goddard Space Flight Center (Grant NAG5-1019) for this work is greatly appreciated. The authors also like to thank Mark Routson for his help in developing the program for validation and Win-bin Shieh for testing the simulation program MSP.

Reference

[1] Kane, T. R. and D. A. Levinson , Dynamics: Theory and applications, McGraw-Hill Book Company, 1985.
[2] Walker, M. W. and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", Journal of Dynamic Systems, Measurement, and Control, vol. 104, Sep. 1982.
[3] Rosenthal, D. E. and M. A. Sherman, "Symbolic Multibody Equations via Kane's Method(SD/EXACT)" AAS/AIAA Astrodynamics Specialist Conference, New York, Aug. 22-25, 1983
[4] Kane, T. and D. Levinson, "The Use of Kane's Dynamical Equations in Robotics", the Int. J. of Robotics Research Vol. 2, No. 3, 1983.
[5] Hollerbach, J. M., "A recursive Largrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", IEEE, Vol. SMC-10, No. 11, Nov. 1980.
[6] Balafoutis, C. A., R. V. Patel and P. Misra, "Efficient Modeling and Computation of Manipulator Dynamics Using Orthogonal Cartesian Tensors", IEEE J. of Robotics and Automation, Vol. 4, NO. 6. Dec. 1988.
[7] Chen, J., "The Effects of Gear Reduction on Robot Dynamics". Submitted to ASME Journal of Dynamic System, Measurement and Control.
[8] Neuman C. P. and J. J. Murray, "Customized Computational Robot Dynamics", Journal of Robotic Systems, Vol. 4, No. 4, 1987, pp. 503-526.
[9] Paul, R. P., M. Rong and H. Zhang, "The Dynamics of the PUMA Manipulator", in Proceedings of 1983 American Control Conference, San Francisco, California.
[10]Gibbs, J. W. and E. B. Wilson, Vector Analysis, Yale University Press, New Haven, 1931 (seventh printing).
[11]Karlen, J. P., J. M. Thompson and J. D. Farrell, "Design and Control of Modular, Kinematically-Redundant Manipulators", Second AIAA/NASA/USAF Symposium on Automation, Robotics and Advanced Computing for the National Space Program, March 9-11, 1987.

Fig. 4. Solution of Equations Set 1
(joints 1–3)

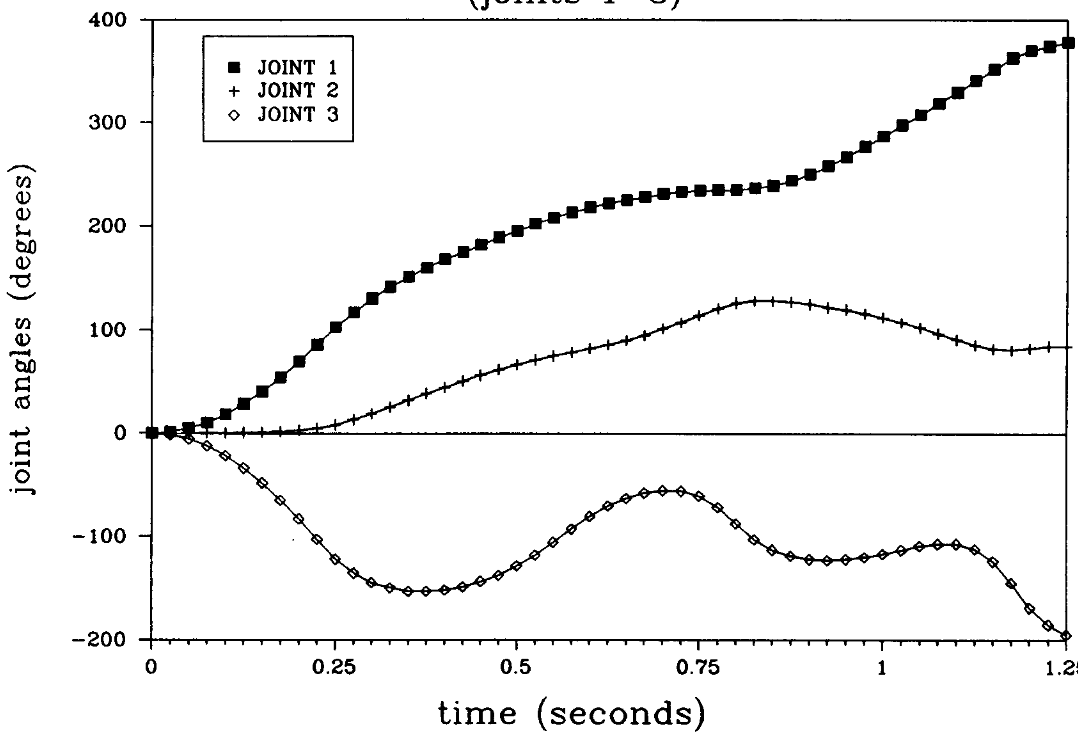


Fig. 5. Solution of Equations Set 1
(joints 4–7)

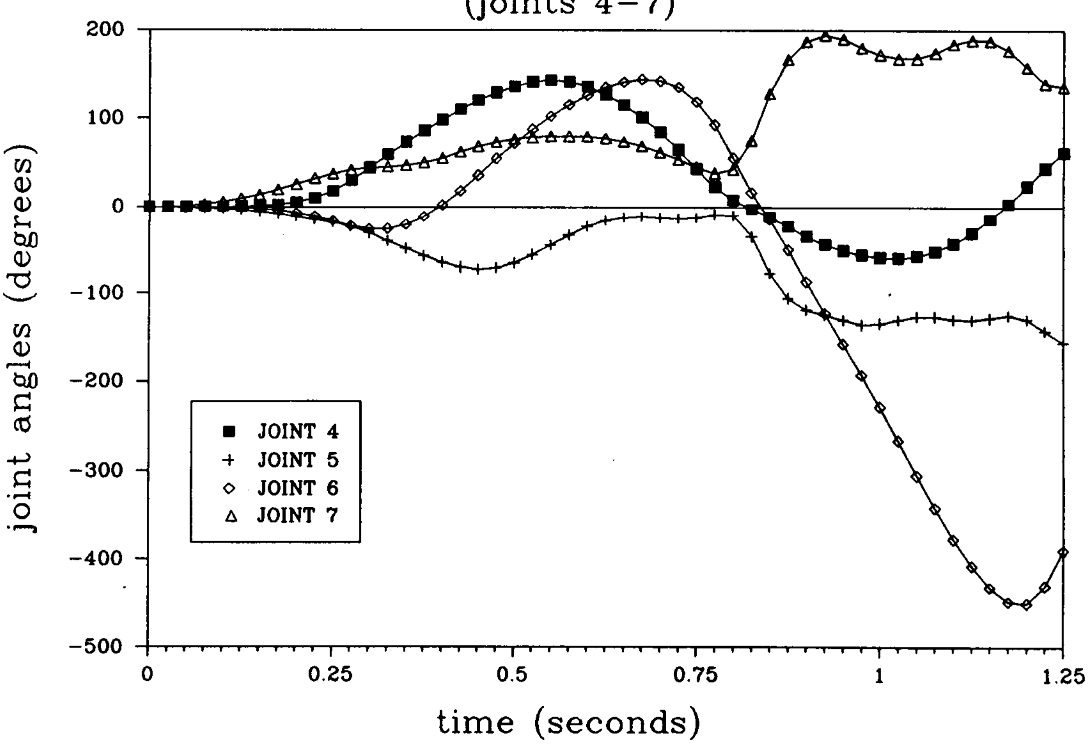


Fig. 6. Solution of Equations Set 2
(joints 1–3)

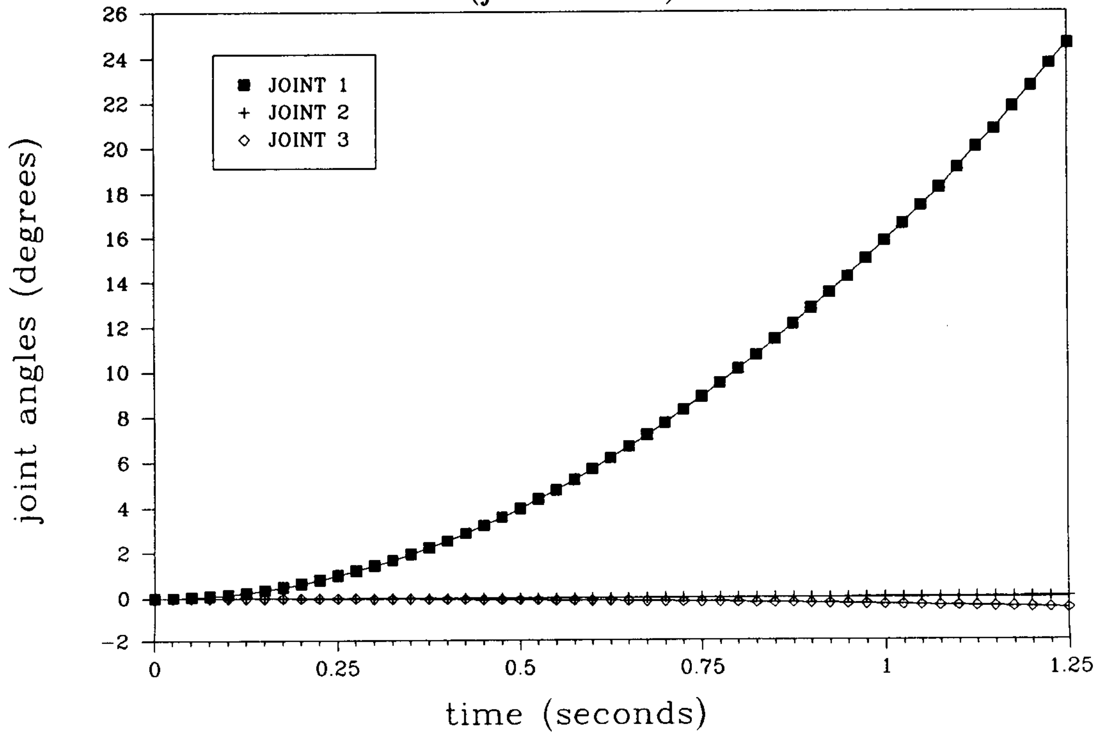


Fig. 7. Solution of Equations Set 2
(joints 4–7)

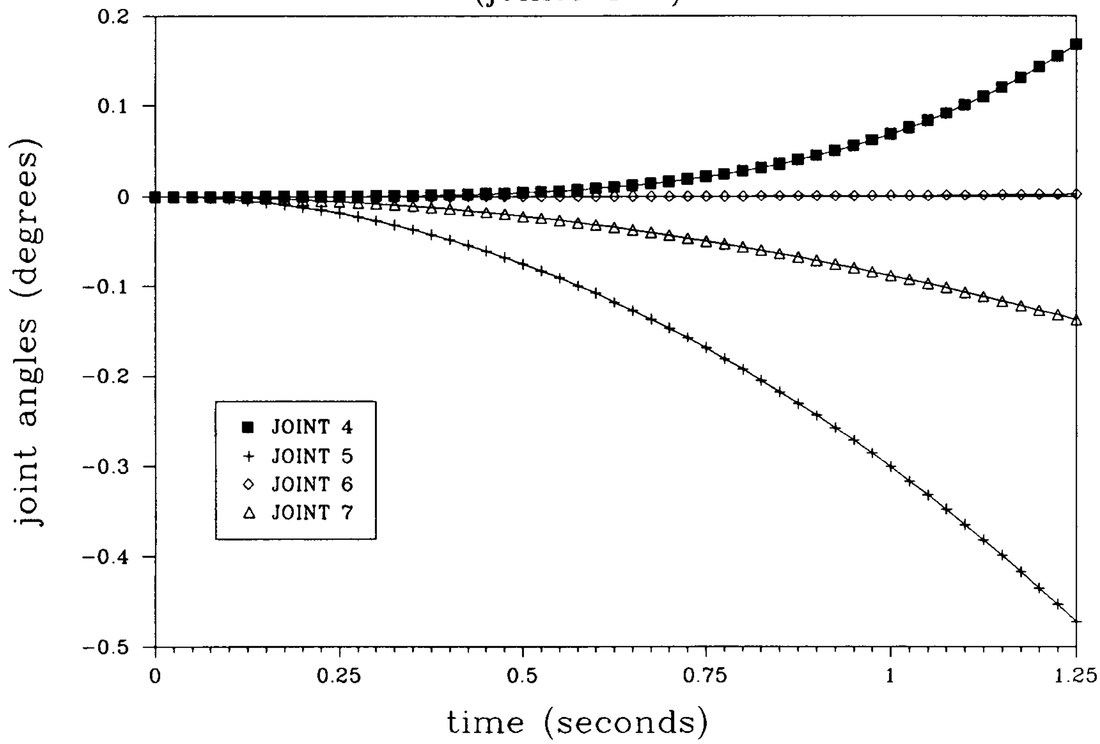


Fig. 8. Solution of Equations Set 2
(joints 1,5,6,7)

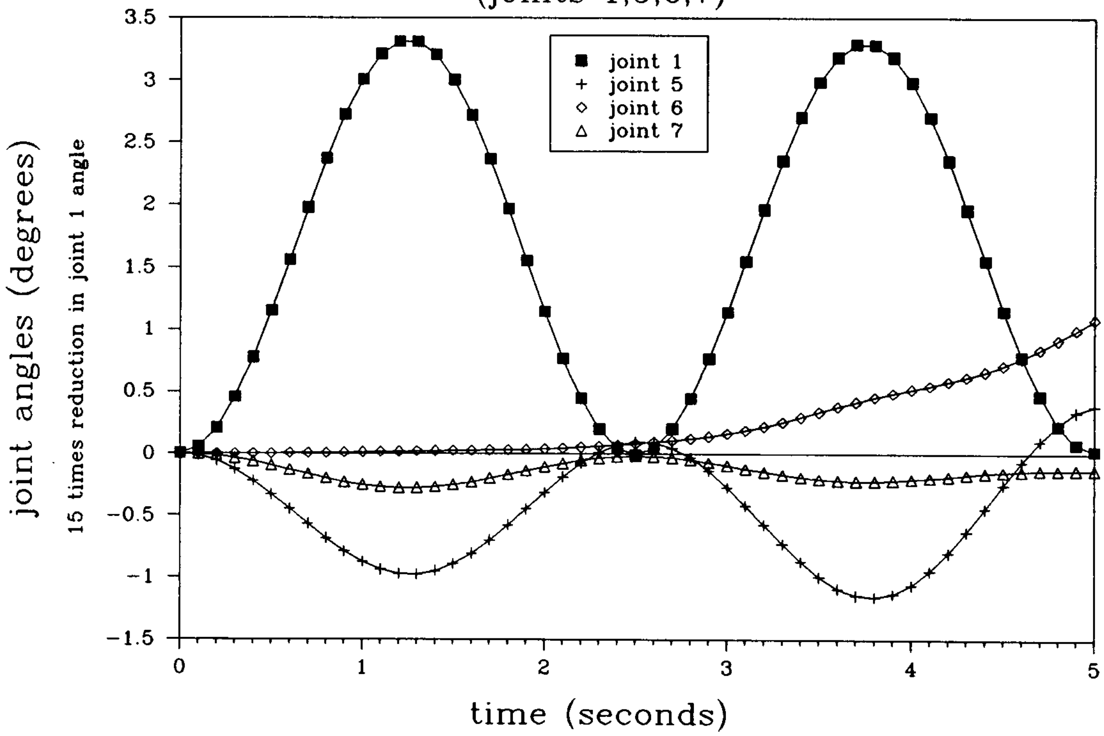
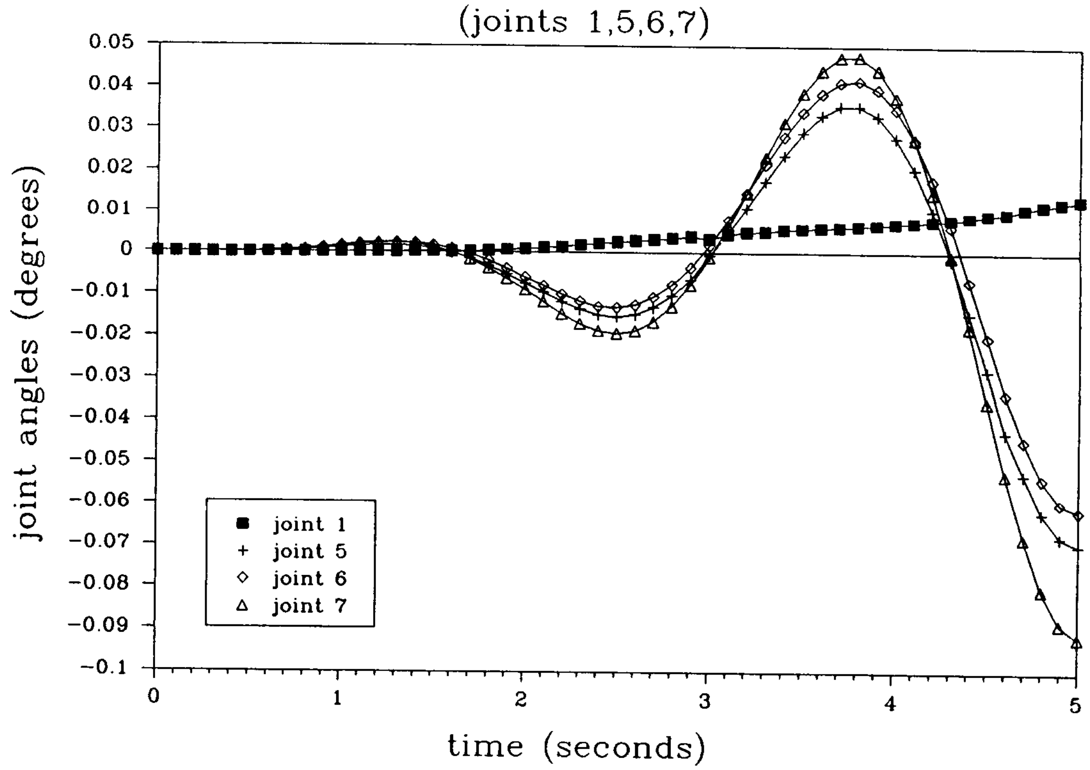


Fig. 9. Difference of Solutions of Set 2 & 3
(joints 1,5,6,7)



Appendix

The parameters of a 7 link Robot Research Corporation manipulator used in the simulation from which results are presented in this paper are in the following.

Mass (including motor and speed reduction mechanism): (kg)

link	1	2	3	4	5	6	7
mass	104.2	55.4	28.8	20.4	11.1	4.7	4.7

Moment of inertia (including motor and speed reduction mechanism): (kg-m^2)

	Ixx	Iyy	Izz
link 1	2.05	2.05	0.7
link 2	0.4	0.4	0.3
link 3	0.6	0.6	0.2
link 4	0.2	0.2	0.2
link 5	0.1	0.1	0.04
link 6	0.03	0.03	0.03
link 7	0.03	0.03	0.03

Speed reduction ratio, rotor inertia (including attachment) and rotor axis (in link fixed unit vector basis):

	speed ratio	rotor inertia (kg-m^2)	rotor axis
base	160	0,003	[0, 0, 1]
link 1	160	0.003	[1, 0, 1]
link 2	200	0.0003	[0, 1, 0]
link 3	200	0.0003	[1, 0, 0]
link 4	200	0.0002	[0, 1, 0]
link 5	200	0.0002	[1, 0, 0]
link 6	160	0.0002	[0, 1, 0]

Joint to mass center position vector (m) r^c_{ii}, i=1,..., 7 :

	X	Y	Z
link 1	0	0	0
link 2	0	0	0
link 3	0	0	0
link 4	0	0	0
link 5	0	0	0
link 6	0	0	0
link 7	0	0	0

Joint to joint position vector (m) $r^L_{i(i+1)}$, i=1,..., 6 :

	X	Y	Z
link 1	0.0	0.0	0.5
link 2	0.3	0.1	0.0
link 3	0.1	0.0	0.3
link 4	0.3	-0.1	0.0
link 5	0.08	0.0	0.4
link 6	0.19	-0.08	0.0