Proceedings of 23rd Conference
on Decision and Control
Las Vegas, NV, December 1984

A COMPUTATIONAL TECHNIQUE FOR INVERSE KINEMATICS*

W.A. Wolovich
Division of Engineering
Brown University
Providence, RI 02912

H. Elliott
Department of Electrical
and Computer Engineering
University of Massachusetts
Amherst, MA 01003

1. INTRODUCTION

Generally speaking, most industrial robots today possess a kinematic structure characterized by one or more revolute joints. However, many of the tasks they are required to perform involve positioning in a "rectangular", cartesian space, such as straight line trajectory motion between two cartesian points or end effector alignment with respect to a defined point on some planar surface. In such cases, control is actually implemented at the joint level; i.e. in "joint space", which will be denoted by the n (≥ 2) -dimensional joint space configuration vector, $\underline{\theta}$, while motion is specified in "cartesian space", which will be denoted by the n-dimensional cartesian space configuration vector $\underline{z}$. In view of these observations, it is not difficult to see that the transformation problem from one space to the other represents perhaps the most fundamental one in robotic manipulation and control. Furthermore, it is of interest to note that this transformation question does not, as yet, have a complete and straightforward solution in the general case.

To be more specific, the important, earlier work of Denavit and Hartenberg [1] laid the foundation for a general solution to the *forward kinematic problem*. In particular, it is now possible to express $\underline{z}$ in terms of $\underline{\theta}$ in a relatively straightforward manner for virtually any known kinematic configuration through the use of homogeneous matrix transformations; i.e. a nonlinear, n-dimensional vector valued function, $\underline{G}(\underline{\theta})$ can be obtained with relative ease such that

$$\underline{z} = \underline{G}(\underline{\theta}) \tag{1}$$

However, the converse or *inverse kinematic problem*, namely solving for a particular joint configuration $\underline{\theta}$ in terms of a given cartesian configuration $\underline{z}$ is not nearly as straightforward. It should be noted, however, that the work of Uicker [2] and Pieper [3], using the Denavit-Hartenberg formulation, is most important in this regard. In particular, for most current industrial robots, especially those characterized by "spherical wrists", the inverse kinematic problem has been solved, and closed-form analytic solutions exist for determining $\underline{\theta}$ in terms of $\underline{z}$. It should be noted, however, that these solutions are generally recursive and, more often than not, non-unique, although specific additional knowledge regarding the particular manipulator configuration can usually be used to resolve the non-uniqueness question. Nonetheless, as noted by Paul [4], "obtaining a solution for the joint coordinates requires intuition and is the most difficult problem we will encounter."

In view of the preceeding, the main purpose of this paper is to present a new, computational (numerical) solution to the "general version" of the inverse kinematic problem. In particular, it will be shown that a relatively simple dynamical system can be constructed which, when "driven" by a desired, time-varying, cartesian configuration vector, $\underline{z}_d(t)$, produces not only the corresponding $\underline{\theta}_d(t)$, such that $\underline{G}(\underline{\theta}_d(t)) = \underline{z}_d(t)$, but also $\dot{\underline{\theta}}_d(t)$ and, if required, $\ddot{\underline{\theta}}_d(t)$, as well. The ability to solve the general inverse kinematic problem in real time is fundamental to the implementation of certain more advanced control algorithms, as will be shown.

* This work was supported in part by Grants ECS-8402921 and AFOSR 83-0359.

Finally, it is of interest to note that the solution which will be presented to the inverse kinematic problem is generally applicable in the sense that it can be used to solve any set of n nonlinear algebraic equations in n unknowns, under the constraints outlined in the next section, and not just particular sets of equations which arise from kinematic chains. Nonetheless, the field of robotics, and the inverse kinematic problem in particular, represents an ideal application of the procedure, and it is for this reason that the dynamical system technique is presented here for the first time.

2. THE SOLUTION

Let any two n-dimensional vector valued functions of time, $\underline{z}(t)$ and $\underline{\theta}(t)$, be related by a known $\underline{G}(\underline{\theta})$, as in (1). Assume that each (i-th) row, $G_i(\underline{\theta})$, of $\underline{G}(\underline{\theta})$ is at least once differentiable in all of its parameters, $\underline{\theta}$, which allows definition of the (nxn) *Jacobian* matrix associated with this relationship; i.e.

$$\dot{\underline{z}}(t) = \underline{J}(\underline{\theta})\,\dot{\underline{\theta}}(t), \tag{2}$$

with each element,

$$J_{ij}(\underline{\theta}) = \frac{\partial G_i(\underline{\theta})}{\partial \theta_j}. \tag{3}$$

Now let $\underline{J}^T(\underline{\theta}_s)$ represent the transpose of the Jacobian, and $\underline{\int}_n$ denote a string of n parallel integrators. Furthermore, suppose that $\underline{\theta}_d(t)$ represents a solution to equation (1) for any given, desired cartesian trajectory, $\underline{z}_d(t)$. In view of these preliminaries, the main result of this paper can now be stated and formally established.

THEOREM: Consider the dynamical system of Figure 1. If (i) $\underline{K}$ is positive definite, (ii) the magnitude of the determinant of the Jacobian is bounded both from above and away from zero from below for the particular trajectory $\underline{z}_d(t)$ considered, and (iii) the magnitude of $\dot{\underline{z}}_d(t)$ is bounded from above, then there exists a positive scalar s and a time $T > 0$ such that for all time $t > T$,

$$\|\underline{\theta}_s(t) - \underline{\theta}_d(t)\| \leq s \tag{4}$$

Furthermore, s can be made arbitrarily small by increasing the minimum eigenvalue of $\underline{K}$.

Proof: First define a joint configuration error signal, namely,

$$\underline{e}(t)=\underline{\theta}_s(t)-\underline{\theta}_d(t), \tag{5}$$

so that

$$\underline{\theta}_s(t)=\underline{e}(t)+\underline{\theta}_d(t) \tag{6}$$

Next, deleting the explicit reference to time dependence for convenience, note that the dynamical system of Figure 1 implies the relation:

$$\dot{\underline{\theta}}_s=-\underline{K}\,\underline{J}^T(\underline{\theta}_s)[\underline{G}(\underline{\theta}_s)-\underline{G}(\underline{\theta}_d)] \tag{7}$$

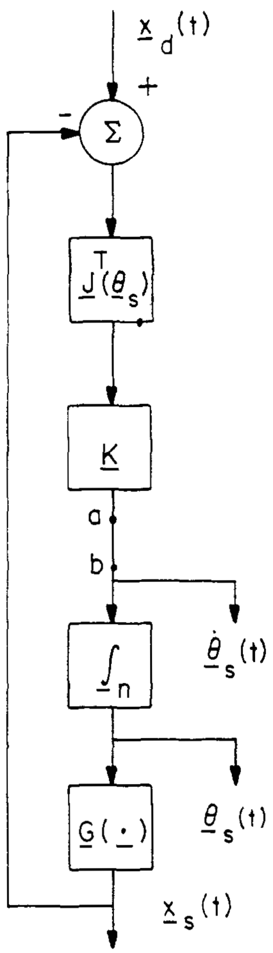

FIGURE 1

AN INVERSE KINEMATIC DYNAMICAL SYSTEM

For notational convenience, define

$$\Sigma(\underline{\varepsilon},\underline{\theta}_d)=\underline{G}(\underline{\varepsilon}+\underline{\theta}_d)-\underline{G}(\underline{\theta}_d), \tag{8}$$

so that differentiating (5), in view of (6)-(8), yields the result:

$$\dot{\underline{\varepsilon}}=-\underline{K}\underline{J}^T(\underline{\varepsilon}+\underline{\theta}_d)\Sigma(\underline{\varepsilon},\underline{\theta}_d)-\dot{\underline{\theta}}_d \tag{9}$$

Now define the time-varying *Lyapunov function* of the error signal $\underline{\varepsilon}$, namely

$$V(\underline{\varepsilon},t)=1/2\ \Sigma^T(\underline{\varepsilon},\underline{\theta}_d)\Sigma(\underline{\varepsilon},\underline{\theta}_d) \tag{10}$$

In view of the bound on the magnitude of the determinant of the Jacobian, it follows that for $\underline{\varepsilon}\neq 0$,

$$V(\underline{\varepsilon},t)>0 \tag{11}$$

Since

$$\dot V(t)=\frac{\partial V(\underline{\varepsilon},t)}{\partial \underline{\varepsilon}}\ \frac{\partial \underline{\varepsilon}}{\partial t}+\frac{\partial V(\underline{\varepsilon},t)}{\partial t}, \tag{12}$$

it follows that along trajectories of (9),

$$\dot V=-\Sigma^T(\underline{\varepsilon},\underline{\theta}_d)\underline{J}(\underline{\varepsilon}+\underline{\theta}_d)\underline{K}\underline{J}^T(\underline{\varepsilon}+\underline{\theta}_d)\Sigma(\underline{\varepsilon},\underline{\theta}_d)-\dot{\underline{\theta}}_d^T\underline{J}^T(\underline{\theta}_d)\Sigma(\underline{\varepsilon},\underline{\theta}_d) \tag{13}$$

Let λ_J denote the minimum eigenvalue of $\underline{J}(\underline{\varepsilon}+\underline{\theta}_d)\underline{J}^T(\underline{\varepsilon}+\underline{\theta}_d)$, noting that $\lambda_J>0$ by assumption. Furthermore, let λ_K denote the minimum eigenvalue of $\underline{K}$, noting that $\lambda_K>0$, since $\underline{K}$ is positive definite. In view of these definitions, it follows from (13) that

$$\dot V\leq-\lambda_K\lambda_J||\Sigma(\underline{\varepsilon},\underline{\theta}_d)||^2-\dot{\underline{\theta}}_d^T\underline{J}^T(\underline{\theta}_d)\Sigma(\underline{\varepsilon},\underline{\theta}_d) \tag{14}$$

Since $\dot{\underline{x}}_d$ is bounded by assumption, $\dot{\underline{\theta}}_d=\underline{J}^{-1}(\underline{\theta}_d)\dot{\underline{x}}_d$ is also bounded; i.e. for all $t>0$,

$$||\dot{\underline{\theta}}_d||\leq c_1 \tag{15}$$

Furthermore, by initial assumption,

$$||\underline{J}(\underline{\theta}_d)||\leq c_2, \tag{16}$$

so that

$$\dot V\leq-\lambda_K\lambda_J||\Sigma(\underline{\varepsilon},\underline{\theta}_d)||^2+c_1c_2||\Sigma(\underline{\varepsilon},\underline{\theta}_d)|| \tag{17}$$

By now defining $c=c_1c_2$ and "completing the square",

$$\dot V\leq-\left[\sqrt{\lambda_K\lambda_J}\,||\Sigma(\underline{\varepsilon},\underline{\theta}_d)||-\frac{c}{\sqrt{2\lambda_K\lambda_J}}\right]^2+\frac{c^2}{2\lambda_K\lambda_J} \tag{18}$$

In view of this result, it follows that as λ_K is increased, $\dot V$ will remain negative for all but "smaller and smaller" values of $||\Sigma(\underline{\varepsilon},\underline{\theta}_d)||$. Therefore, by increasing λ_K sufficiently, $\underline{\varepsilon}(t)$, as given by (5), can be made arbitrarily small after some time T (during which initial mismatch transients will "settle out"). It thus follows that (4) can be satisfied for s arbitrarily small, thus establishing the main result as stated.

3. REMARKS

A new technique for solving the inverse kinematic problem has now been presented and formally established. It is of interest to note that this solution was obtained as a consequence of some preliminary compliant control investigations by the authors of this paper. In particular, it was noted in numerous earlier works; e.g. [4][5], that a fundamental relationship between the vector of joint torques, $\underline{\tau}$, and the cartesian force vector, $\underline{f}$, which produced them can be expressed in terms of the transpose of the Jacobian, namely

$$\underline{\tau}=\underline{J}^T(\underline{\theta})\underline{f} \tag{19}$$

By then utilizing the *generalized spring* observation; i.e. that force errors can be represented as distance errors multiplied by an appropriate (spring) constant, it follows that a cartesian distance vector, when multiplied by a "spring constant" matrix, can be treated as a cartesian force vector. In light of (19), therefore, multiplication of this force vector by the transposed Jacobian produces a torque vector which, in essence, is a signal which could be used to directly control joint motion. These observations serve to motivate the closed loop dynamical system depicted in Figure 1.

They further serve to illuminate some important earlier control schemes which achieve cartesian space control via joint angle manipulations using the transposed Jacobian [6][7]. Indeed, it might be observed at this point that the control techniques presented in these latter two references are not at all unlike the dynamical system of Figure 1. In particular, if the integrator section in Figure 1 were replaced by the joint dynamics of an actual robot, to achieve on-line, closed loop control, then the resulting configuration would be similar to those in [6] and [7]. It should be emphasized, however, that the main goals outlined in these earlier references are on-line, real time control, and not explicitly solving the general inverse kinematic problem.

It should be noted that the dynamical system of Figure 1 will produce $\underline{\theta}(t)$ and $\dot{\underline{\theta}}(t)$ as "outputs", when driven by a desired cartesian configuration vector, $\underline{x}_d(t)$. In certain cases, however, it may be useful to obtain joint acceleration information as well; i.e. $\ddot{\underline{\theta}}(t)$. As it turns out, this is generally possible provided the dynamical system of Figure 1 is augmented by an additional string of n parallel integrators, closed under a (stabilizing) gain matrix $\underline{A}$. In particular, consider such a configuration, as depicted in Figure 2. If this system is inserted between terminals a and b of the Figure 1 system, it is then

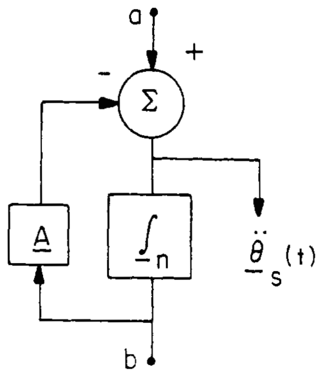

FIGURE 2

AN ADDENDUM TO ACQUIRE $\ddot{\underline{\theta}}_s$ (t)

possible to extract not only joint position and velocity information, but joint acceleration information as well provided, of course, that the resulting dynamical system satisfies stability properties similar to those which characterize the Figure 1 system. It can be shown that this is indeed the case under some rather general and unrestrictive conditions. However, due to spatial limitations, a formal proof of stability of the combined Figure 1 and Figure 2 system will not be presented here. However, a simulation study employing the combined systems will be outlined in the next section to illustrate the utility of such a dynamical systems approach to solving the general inverse kinematic problem.

Finally, it might be noted the key elements which allow one to solve the general inverse kinematic problem using a dynamical system are (i) arbitrarily fast convergence properties associated with the closed loop system and (ii) the explicit use of $\underline{G}(\cdot)$ along with the series integrators in the feedforward path. Indeed, any combination of matrix multipliers which precedes the series integrators and $\underline{G}(\cdot)$, and produces a stable and rapidly convergent closed loop system, can be used to solve the general inverse kinematic problem. In this report, it has been shown that the series combination of $\underline{J}^T(\underline{\theta}_s)$ followed by $\underline{K}$ represents one such series combination. However, it is conjectured that others also exist; e.g. the replacement of $\underline{J}^T(\underline{\theta}_s)$ by $\underline{J}^{-1}(\underline{\theta}_s)$ in Figure 1 could very well represent an acceptable alternative.

4. SIMULATIONS

In order to illustrate the employment of this dynamical system to solve the inverse kinematic problem, consider a three link robot whose joints are similar to the first three joints of the "Stanford Arm"; i.e. the first two joints, denoted as θ_1 and θ_2 are revolute, while the third, denoted as h, is prismatic.

The task chosen for the robot to perform consists of tracking a straight line path in cartesian (x,y,z) space from the point (3,0,5) at time, t=0, through the point (0,2,5) and beyond, accelerating from zero velocity to 1 unit-length/second, as illustrated in Figure 3. This particular task was simulated on a time shared, main-frame computer using the IBM continuous systems simulation language DSL (Dynamic Simulation Language). Furthermore, the double integrator dynamical system (Figure 1 with Figure 2) was employed to solve the general inverse kinematic problem; i.e. to obtain trajectories for not only the joint positions, but their velocities and accelerations as well.

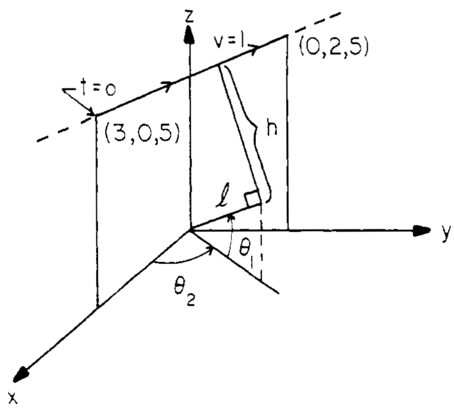

FIGURE 3

THE TASK

The results of this particular simulation are presented in Figures 4 thru 7, which are essentially self-explanatory; i.e. Figures 4-6 display the positions, velocities, and accelerations for θ_1 , θ_2 , and h, respectively, as functions of time, while Figure 7 shows the disparity between the desired and actual trajectories in the x direction. It might be noted that in this particular simulation, $\underline{K}$ was defined as a diagonal matrix, with each diagonal entry equal to 100, while $\underline{A}$ was also defined as a diagonal matrix, with each diagonal entry equal to 30. The time required to solve the inverse kinematic problem was dependent on the number of users on the system at the particular time the simulation was run. The use of an "appropriately sized" dedicated machine, however, would enable one to solve the inverse kinematic problem "as fast as required" for actual use in real-time control. It should be noted that the plots that were obtained represent a "smooth" and satisfactory solution to the inverse kinematic problem.

These plots and the numerical data associated with them (which was not explicitly presented) could be used in a variety of ways to control actual robots. In particular, it might be noted that some of the more innovative control schemes, such as the *inverse problem* technique [8][9], also called the *computed torque* technique [10][11], require explicit computation of the inverse positions, velocities, and accelerations for real time, on-line control. The utilization of an inverse kinematic dynamical system, as outlined here, could prove most useful in the practical implementation of such techniques.

REFERENCES

[1] Denavit, J. and R.S. Hartenberg *A Kinematic Notation for Lower Pair Mechanism Based on Matrices* J. Applied Mechanics 22 June, 1955 pp. 215-221

[2] Uicker, J.J. *On the Dynamic Analysis of Spatial Linkages Using 4 by 4 Matrices* Ph.D Thesis Dept. of Mechanical Engineering and Astronautical Sciences, Northwestern University, 1965

[3] Pieper, D.L. *The Kinematics of Manipulators Under Computer Control* Ph.D Thesis Department of Computer science, Stanford University, 1968

[4] Paul, R.P. *Robot Manipulators: Mathematics, Programming, and Control* MIT Press, 1981

[5] Groome, R.C. *Force Feedback Steering of a Teleoperator System* M.S. Thesis, Dept. of Aeronautics and Astronautics, MIT, 1972

[6] Salisbury, J. Kenneth and Craig, John J. *Articulated Hands: Force Control and Kinematic Issues* Int. Journal of Robotics Research Vol. 1 (1) Spring, 1982 pp. 4-17

[7] Craig, John J. *Introduction to Robotics: Mechanics and Control* Addison Wesley, Spring, 1985

[8] Paul, R.P. *Modelling, Trajectory Calculation, and Servoing of a Computer Controlled Arm* Stanford University Artificial Laboratory, AIM 177, November, 1972

[9] Raibert, M.H. and Horn, B.K.P. *Manipulator Control using the Configuration Space Method* Industrial Robot 5 June, 1978 pp. 69-73

[10] Markiewicz, B.R. *Analysis of the Computer Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator* Jet Propulsion Laboratory, California Institute of Technology, TM 33-601, March, 1973

[11] Bejczy, A.K. *Robot Arm Dynamics and Control* ibid TM 33-669 February, 1974

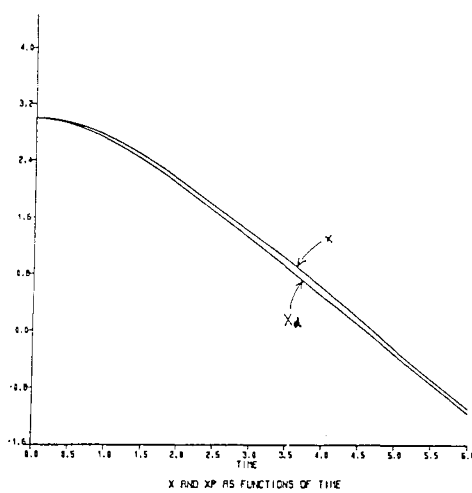

FIGURE 7

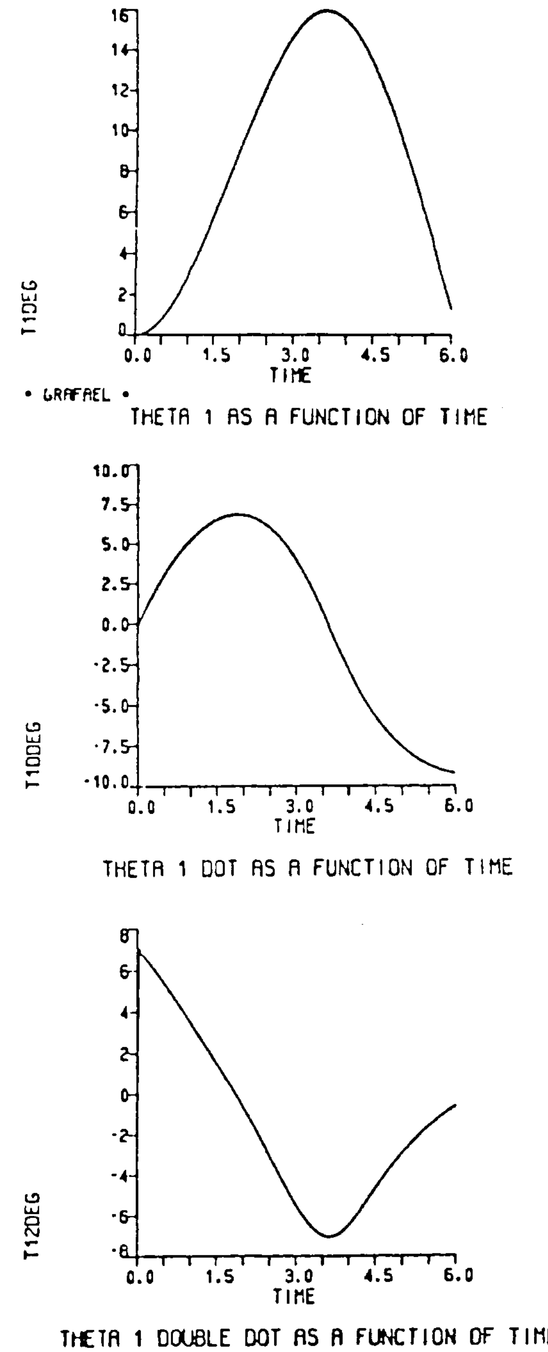

FIGURE 4

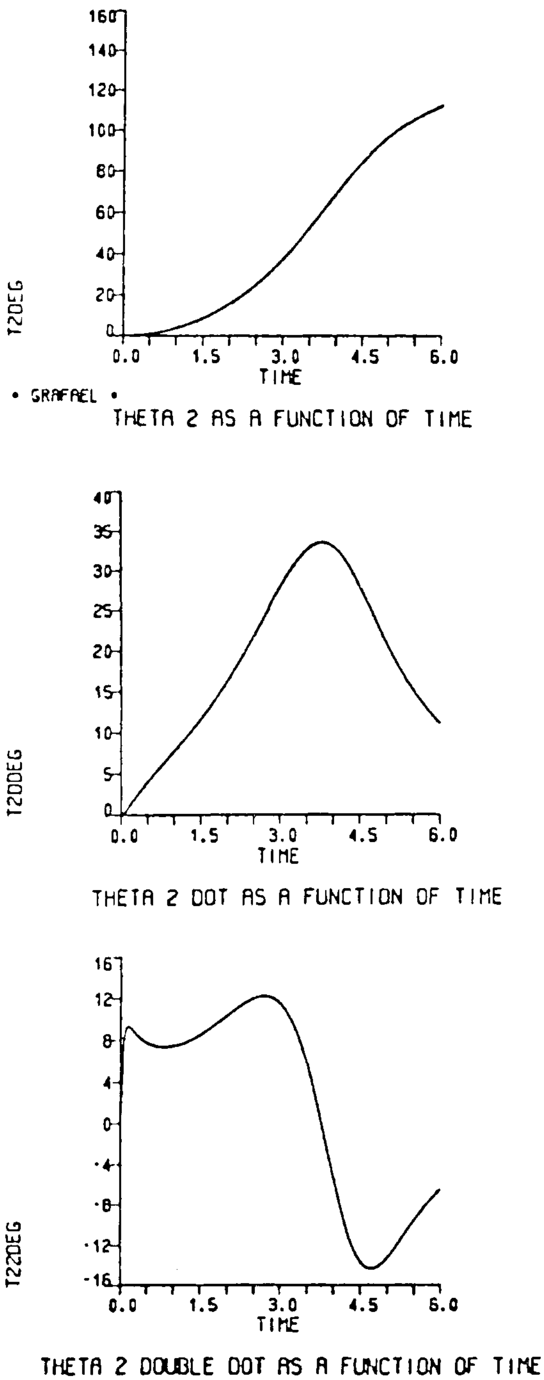

FIGURE 5

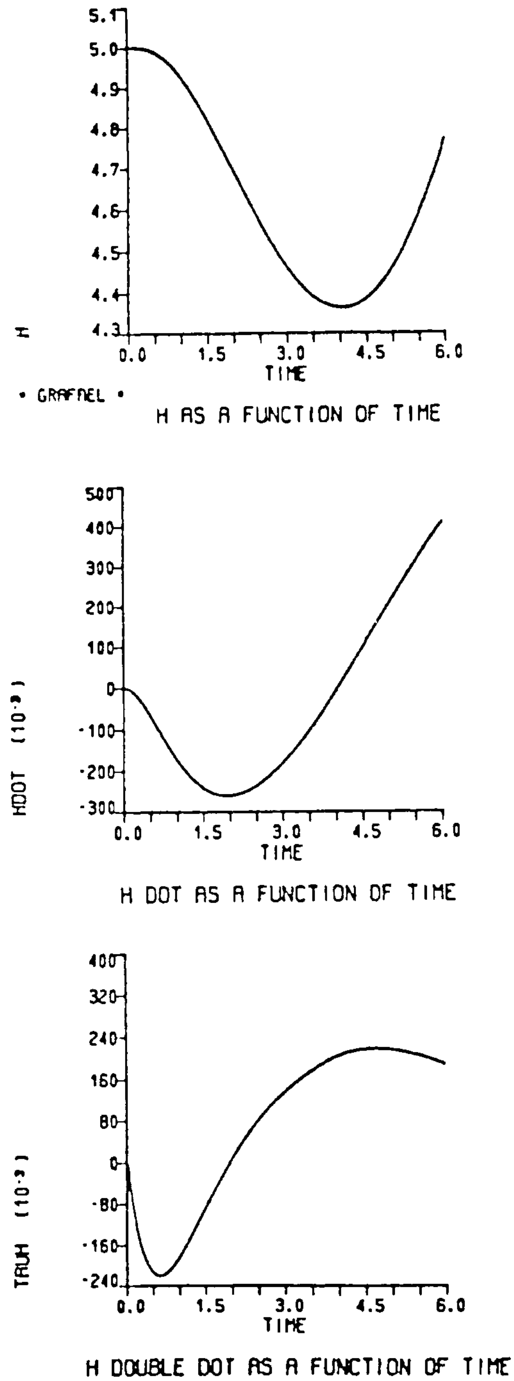

FIGURE 6