RDS-TR-5-82

ON THE CONTROL OF MECHANICAL MANIPULATORS

C. S. G. Lee
M. J. Chung

April 1982

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109

This work was supported in part by the National Science Foundation Grant ECS-8106954 and the

TABLE OF CONTENTS

Abstract

The dynamic performance of computer-controlled manipulators is directly linked to the formulation of the dynamic model of manipulators and its corresponding control law. Various formulations are available to describe the dynamic models of mechanical manipulators and most notable of these are the Lagrange-Euler and the Newton-Euler formulations. This report describes an efficient position plus derivative control in the joint variable space for a PUMA[*] robot arm whose dynamic equations of motion are formulated by the Newton-Euler method. The recursive controller compensates the inertia loading, the nonlinear coupling reaction forces between joints and the gravity loading effects. Using a PDP 11/45 computer, the controller equations can be computed within 3 ms, which is sufficient for real-time control. Computer simulation of the performance of the control law is included for discussion.

[*] PUMA is a trademark of Unimation Inc.

1. Introduction

A mechanical manipulator can be modeled as an open-loop articulated chain with several rigid bodies (links) connected in series by either revolute or prismatic joints. One end of the chain is attached to a supporting base while the other end is free and attached with a tool (the end-effector) to manipulate objects or perform assembly tasks. The motion of the joints result in relative motion of the links.

The purpose of manipulator control is to maintain the dynamical response of an electromechanical manipulator in accordance with some pre-specified system performance and desired goals. In general, the control problem consists of (i) obtaining dynamic models of the physical system and (ii) specifying corresponding control laws or strategies to achieve the desired system response and performance. This report deals with the second part of the control problem of computer-controlled manipulators, and in particular, the PUMA robot arm.

A priori information needed for control is a set of differential equations describing the dynamic behavior of the manipulator. Though various approaches are available to formulate the robot arm dynamics such as the Lagrange-Euler [Uic65], the "Recursive-Lagrange" [Hol80], the Newton-Euler [LWP80], the Lagrange form of D'Alembert Principle [LLN82], and more recently the "Gibbs-Appell" [HoT80] formulation, two main approaches remain to be used by most researchers to systematically derive the dynamic model of the manipulator - the Lagrange-Euler and the Newton-Euler formulations. After obtaining the dynamic equations of motion of the manipulator, a suitable control law must be designed to compute the necessary feedback torques/forces to actuate the joints for every set point ($\vartheta^d, \dot{\vartheta}^d, \ddot{\vartheta}^d$) in a pre-planned trajectory. Bejczy [Bej74] based on the Lagrangian formulation has shown that the dynamic equations of motion for a 6-jointed manipulator are highly nonlinear and consists of inertia loading, coupling reaction forces between joints and gravity loading effects. Hence, the control law must be designed to compensate all these nonlinear effects. A position plus derivative control based on the computed torque technique has been used previously to servo a Stanford arm [Mar73] whose dynamic equations of motion are formulated by the Lagrange-Euler approach. However, the dynamic equations of motion as formulated by the Lagrange-Euler method have been shown to be computationally inefficient [TML80,Pau72], and real-time control based on the 'complete' dynamic model has been found difficult to achieve if not impossible [Pau72]. A simple control law in joint space which compensates the inertia loading, the coupling reaction forces between joints and the gravity loading is shown through the "Equivalence Formulation" [TML80,LCT82] to have the same control effects as the one obtained by the computed torque technique. This control law is based on the dynamic equations of motion formulated by the Newton-Euler method. Computer simulation of the performance of the proposed control law for a PUMA robot arm on a VAX-11/780 computer shows the expected result.

In the following sections, vectors are represented in boldface lower case alphabets while matrices are in boldface upper case alphabets.

2. Kinematics and Notation for Manipulators

A mechanical manipulator consists of a sequence of rigid bodies, called links, connected by either revolute or prismatic joints. Each pair of joint-link constitutes one degree of freedom. Hence for an n degree-of-freedom manipulator, there are n pairs of joint-link with link 0 attached to a supporting base where an inertial coordinate frame is established. In order to describe the translational and rotational relationship between adjacent links, a Denavit-Hartenberg matrix representation for each link is used [DeH55] and shown in Figure 1. From Figure 1, an orthonormal coordinate frame system (x_i, y_i, z_i) is assigned to the i^{th} link, where the z_i axis passes through the axis of motion of joint i+1, and the x_i axis is normal to the z_{i-1} axis, while the y_i axis completes the right hand rule. With this orthonormal coordinate frame, link i is characterized by two parameters: a_i, the common normal distance between the z_{i-1} and z_i axes, and α_i, the twist angle measured between the z_{i-1} and z_i axes in a plane perpendicular to a_i; and joint i which connects link i-1 to link i is characterized by a distance parameter d_i measured between the x_{i-1} and x_i axes and a revolute joint variable ϑ_i which is between the normals and measured in a plane normal to the joint axis. If joint i is prismatic, then it is characterized by an angle parameter ϑ_i and a joint variable d_i. With the coordinate frames established for adjacent links (link i and link i-l), one can relate the relationship between the adjacent coordinate frames (*i^{th}* and *$i-1^{th}$ frames*) by performing the following four operations (see Figure 1): (a) Rotate an angle of ϑ_i about the z_{i-i} axis (the x_{i-1} and x_i *axes are aligned*). (b) Translate a distance of d_i along the z_{i-1} axis (the x_{i-1} and x_i *axes are coincident*). (c) Translate a distance of a_i along the x_i axis (*the two origins are coincident*). (d) Rotate an angle of α_i about the x_i axis.

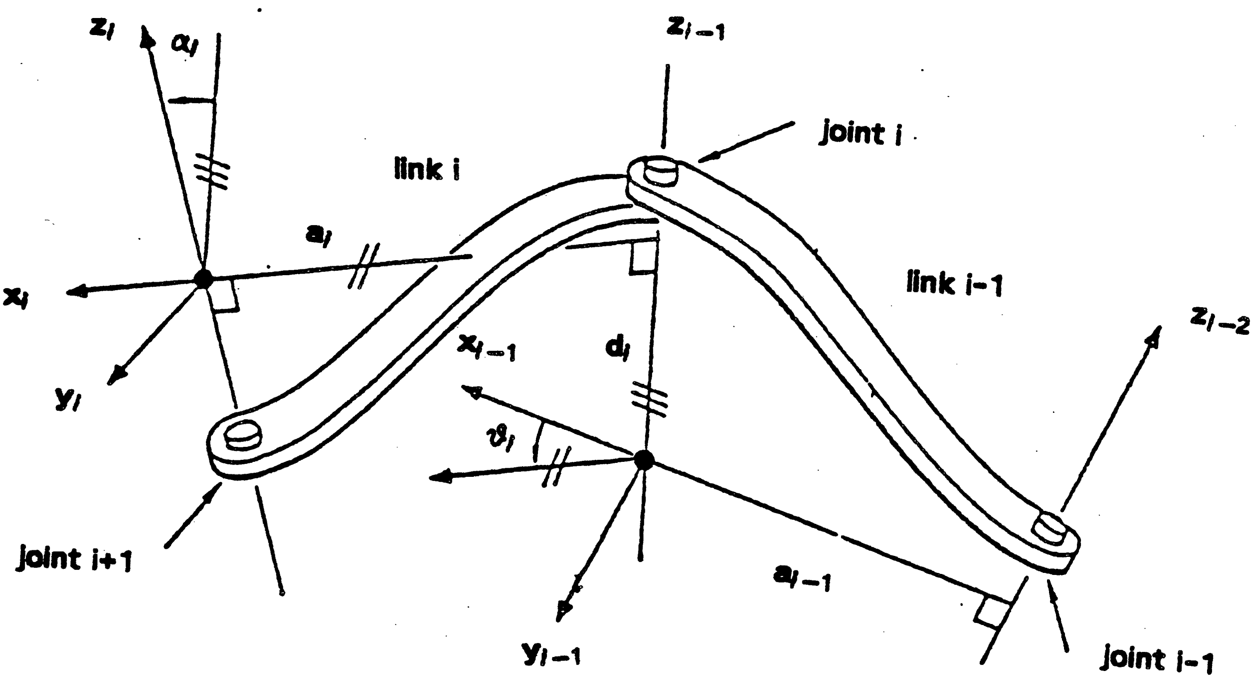

Figure 1 Parameters of a Link Coordinate System

These four operations may be expressed by a 4 × 4 homogeneous coordinate transformation matrix as:

$$T^{i}_{i-1} = \begin{bmatrix} \cos\vartheta_i & -\cos\alpha_i \sin\vartheta_i & \sin\alpha_i \sin\vartheta_i & a_i \cos\vartheta_i \\ \sin\vartheta_i & \cos\alpha_i \cos\vartheta_i & -\sin\alpha_i \cos\vartheta_i & a_i \sin\vartheta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.1)$$

$$= \begin{bmatrix} R^{i}_{i-1} & p^{i}_{i-1} \\ 0 & 1 \end{bmatrix} \qquad (2.2)$$

The upper left 3x3 submatrix of T^{i}_{i-1} is called the rotation matrix R^{i}_{i-1} while the upper right 3x1 vector is called the position vector p^{i}_{i-1} . One can view the rotation matrix R^{i}_{i-1} as a transformation matrix which maps a vector $r_i = (x, y, z)^T$ expressed in the i^{th} coordinate frame into the $(i-1)^{th}$ coordinate frame with both origins coincided at one point, and the position vector as the displacement vector of the origin of the i^{th} coordinate frame from the origin of the $(i-1)^{th}$ coordinate frame.

The above kinematics relationship between adjacent links will be used in the following sections to derive the dynamic equations of motion and show the equivalence of the two controllers based on the two most popular arm dynamics formulations.

3. Dynamics of Manipulators

The dynamic equations of motion for a PUMA robot arm can be obtained from known physical laws (Newtonian and Lagrangian mechanics) and physical measurements (link inertias and geometric parameters). The actual derivation is based on the Lagrangian/Newtonian formulation applied to open articulated chains represented in Denavit-Hartenberg matrix notation form. The equations of motion for a six-jointed manipulator have been derived previously by Bejczy[Bej74], Paul[Pau72] and Uicker[Uic65] using the Lagrangian generalized coordinates. The equations of motion derived from the Lagrangian and Newtonian formulations will be briefly presented here.

3.1. Lagrange-Euler Formulation

Consider a position vector expressed in homogeneous coordinates, $\mathbf{p} = (x, y, z, 1)^T$, which points from the base coordinate system to a differential mass, *dm*, located in the i^{th} link. $\mathbf{p}$ can be written as:

$$\mathbf{p}=\mathbf{T}_0^i\mathbf{r}_i\,, \quad \text{and} \quad \mathbf{r}_i = (x_i\,, y_i\,, z_i\,, 1)^T \tag{3.1}$$

where $\mathbf{r}_i$ is the position of the differential mass dm represented in the i^{th} coordinate frame.

The velocity of this differential mass with respect to the base coordinate frame (an inertial frame) is:

$$\mathbf{v}_0^i = \frac{d\mathbf{p}}{dt} = \left[\sum_{j=1}^{i} \frac{\partial \mathbf{T}_0^i}{\partial \vartheta_j}\dot{\vartheta}_j\right]\mathbf{r}_i \qquad ;\ \text{for } i=1,2,\ldots n \tag{3.2}$$

The associated kinetic energy dK_i is $\frac{1}{2}Tr(\mathbf{v}_0^i(\mathbf{v}_0^i)^T)dm$ which equals:

$$dK_i = \frac{1}{2}\sum_{j=1}^{i}\sum_{k=1}^{i} Tr\left\{\frac{\partial \mathbf{T}_0^i}{\partial \vartheta_j}\mathbf{r}_i(\mathbf{r}_i)^T dm\left[\frac{\partial \mathbf{T}_0^i}{\partial \vartheta_k}\right]^T \dot{\vartheta}_j\dot{\vartheta}_k\right\} \tag{3.3}$$

When each link is integrated over its entire mass and the kinetic energies of all links are summed, we have:

$$K.E. = \sum_{i=1}^{n}\int dK_i = \sum_{i=1}^{n}\left\{\frac{1}{2}Tr\left\{\sum_{j=1}^{i}\sum_{k=1}^{i}\frac{\partial \mathbf{T}_0^i}{\partial \vartheta_j}\mathbf{J}_i\left[\frac{\partial \mathbf{T}_0^i}{\partial \vartheta_k}\right]^T\dot{\vartheta}_j\dot{\vartheta}_k\right\}\right\} \tag{3.4}$$

where $\mathbf{J}_i$ is defined as:

$$\mathbf{J}_i^i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2}+\bar{r}_x^2 m_i & \bar{r}_x\bar{r}_y m_i & \bar{r}_x\bar{r}_z m_i & \bar{r}_x m_i \\ \bar{r}_y\bar{r}_x m_i & \frac{I_{xx}-I_{yy}+I_{zz}}{2}+\bar{r}_y^2 m_i & \bar{r}_y\bar{r}_z m_i & \bar{r}_y m_i \\ \bar{r}_z\bar{r}_x m_i & \bar{r}_z\bar{r}_y m_i & \frac{I_{xx}+I_{yy}-I_{zz}}{2}+\bar{r}_z^2 m_i & \bar{r}_z m_i \\ \bar{r}_x m_i & \bar{r}_y m_i & \bar{r}_z m_i & m_i \end{bmatrix}$$

The total potential energy of the arm is the sum of the potential energy of each link expressed in the base coordinate frame:

$$P.E. = \sum_{i=1}^{6} P_i = \sum_{i=1}^{6} -m_i \mathbf{g}\mathbf{T}_o^i \bar{\mathbf{r}}_i \tag{3.5}$$

where

$\bar{\mathbf{r}}_i$ is the position vector of the center of mass of link i expressed in the i^{th} coordinate frame.

$\mathbf{g}$ is the gravity row vector = $(g_x, g_y, g_z, 0)$ and $|\ \mathbf{g}\ | = 9.8062 m/s^2$

Applying the Lagrange-Euler equations of motion to the Lagrangian function $L = K.E. - P.E.$, we obtain the necessary generalized torque τ_i for joint i to drive the i^{th} link of the arm:

$$\tau_i = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\vartheta}_i}\right] - \frac{\partial L}{\partial \vartheta_i} = \sum_{k=i}^{6} \sum_{j=1}^{k} Tr\left\{\frac{\partial \mathbf{T}_o^k}{\partial \vartheta_j}\mathbf{J}_k\left(\frac{\partial \mathbf{T}_o^k}{\partial \vartheta_i}\right)^T\right\}\ddot{\vartheta}_j$$

$$+ \sum_{m=i}^{6} \sum_{j=1}^{m} \sum_{k=1}^{m} Tr\left\{\frac{\partial^2 \mathbf{T}_o^m}{\partial \vartheta_j \partial \vartheta_k}\mathbf{J}_m\left(\frac{\partial \mathbf{T}_o^m}{\partial \vartheta_i}\right)^T\right\}\dot{\vartheta}_j\dot{\vartheta}_k - \sum_{j=i}^{6} m_j \mathbf{g}\frac{\partial \mathbf{T}_o^j}{\partial \vartheta_i}\bar{\mathbf{r}}_j \quad ; \text{for } i=1,2,...,6 \tag{3.6}$$

Because of its matrix structure, this formulation is appealing from a control viewpoint in that it gives a set of closed form differential equations as:

$$\mathbf{D}(\vartheta)\ddot{\vartheta} + \mathbf{H}(\vartheta,\dot{\vartheta}) + \mathbf{G}(\vartheta) = \tau \tag{3.7}$$

where:

$\mathbf{D}(\vartheta)$ = a 6x6 *inertia acceleration matrix*

$$D_{ik} = \sum_{j=max(i,k)}^{6} Tr\left\{ \frac{\partial \mathbf{T}_0^j}{\partial \vartheta_k} \mathbf{J}_j \left[\frac{\partial \mathbf{T}_0^j}{\partial \vartheta_i} \right]^T \right\} \quad ; \text{for } i,k=1,2,...,6 \tag{3.8}$$

$\mathbf{H}(\vartheta , \dot{\vartheta})$ = a 6x1 *nonlinear Coriolis* and *Centrifugal vector*

$$H_{ikm} = \sum_{j=max(i,k,m)}^{6} Tr\left\{ \frac{\partial^2 \mathbf{T}_0^j}{\partial \vartheta_k \partial \vartheta_m} \mathbf{J}_j \left[\frac{\partial \mathbf{T}_0^j}{\partial \vartheta_i} \right]^T \right\} \dot{\vartheta}_k \dot{\vartheta}_m \quad ; \text{for } k,m=1,2,...,6 \tag{3.9}$$

$\mathbf{G}(\vartheta)$ = a 6x1 *gravity loading vector of the links*

$$G_i = - \sum_{j=i}^{6} m_j \ \mathbf{g} \frac{\partial \mathbf{T}_0^j}{\partial \vartheta_i} \bar{\mathbf{r}}_j \quad ; \text{for } i=1,2,...,6 \tag{3.10}$$

$$\vartheta = (\vartheta_1 , \vartheta_2 , \cdots , \vartheta_6)^T$$

$$\dot{\vartheta} = (\dot{\vartheta}_1 , \dot{\vartheta}_2 , \cdots , \dot{\vartheta}_6)^T$$

$$\ddot{\vartheta} = (\ddot{\vartheta}_1 , \ddot{\vartheta}_2 , \cdots , \ddot{\vartheta}_6)^T$$

$$\tau = (\tau_1 , \tau_2 , \tau_3 , \tau_4 , \tau_5 , \tau_6)^T$$

= external applied torques for the joints

This form provides more insight to the coupling effects between joints and to designing a control law that compensates all these nonlinear effects easily. Computationally, however, the Lagrangian formulation is extremely inefficient as compared with other formulations.

3.2. Newton-Euler Formulation

The Newton-Euler equations of motion of a manipulator consist of a set of compact forward and backward recursive equations. The most significant of this formulation is the computation time of the applied torques could be reduced tremendously so that real-time open-loop control is possible. A brief derivation of the formulation based on [LWP80] is presented here for completeness.

The forward recursive equations propagate linear velocity, linear acceleration, angular velocity, angular acceleration, total link forces and moments from the base to the end-effector of the manipulator. For manipulators having all the rotary joints, these equations are:

Forward Equations: $i = 1, 2, \cdots, n$

$$\omega_i = \mathbf{R}_i^{i-1}(\omega_{i-1} + \mathbf{z}_o\dot{\vartheta}_i) \tag{3.11}$$

$$\alpha_i = \mathbf{R}_i^{i-1}[\alpha_{i-1} + \mathbf{z}_o\ddot{\vartheta}_i + \omega_{i-1} \times \mathbf{z}_o\dot{\vartheta}_i] \tag{3.12}$$

$$\mathbf{a}_i = \alpha_i \times \mathbf{r}_i + \omega_i \times \left[\omega_i \times \mathbf{r}_i\right] + \mathbf{R}_i^{i-1}\mathbf{a}_{i-1} \tag{3.13}$$

$$\bar{\mathbf{a}}_i = \alpha_i \times \bar{\mathbf{r}}_i + \omega_i \times [\omega_i \times \bar{\mathbf{r}}_i] + \mathbf{a}_i \tag{3.14}$$

$$\mathbf{F}_i = m_i\bar{\mathbf{a}}_i \tag{3.15}$$

$$\mathbf{N}_i = \mathbf{I}_i\alpha_i + \omega_i \times \mathbf{I}_i\omega_i \tag{3.16}$$

The backward recursive equations of motion propagate, from the end-effector to the base of the manipulator, the forces and moments exerted on link i by link i-1.

Backward Equations: $i = n, n-1, \cdots, 1$

$$\mathbf{f}_i = \mathbf{R}_i^{i+1}\mathbf{f}_{i+1} + \mathbf{F}_i \tag{3.17}$$

$$\mathbf{n}_i = \mathbf{R}_i^{i+1}\mathbf{n}_{i+1} + \mathbf{r}_i \times \mathbf{f}_{i+1} + (\mathbf{r}_i + \bar{\mathbf{r}}_i) \times \mathbf{F}_i + \mathbf{N}_i \tag{3.18}$$

$$\tau_i = \mathbf{n}_i^T(\mathbf{R}_i^{i-1}\mathbf{z}_o) \tag{3.19}$$

with the "usual" initial conditions of $\omega_o = 0$, $\mathbf{a}_o = g\mathbf{z}_o$, $\alpha_o = 0$, $g = 9.8062 m/s^2$, f_{n+1} = external force exerted on the hand and n_{n+1} = external moment exerted on the hand.

where:

$\mathbf{R}_i^{i-1}$ = the rotation matrix which transforms a vector from its representation in the $i-1^{th}$ coordinate system to its equivalent in the i^{th} coordinate system.

ω_i = the angular velocity of link i with respect to the i^{th} coordinate system.

α_i = the angular acceleration of link i with respect to the i^{th} coordinate system.

$\mathbf{r}_i$ = the position vector of the origin of the $i-1^{th}$ frame with respect to the i^{th} coordinate system.

$\bar{\mathbf{r}}_i$ = the position vector of the center of mass of link i with respect to the i^{th} coordinate system.

$\mathbf{a}_i$ = the linear acceleration vector of link i with respect to the i^{th} coordinate system.

$\bar{\mathbf{a}}_i$ = the linear acceleration vector of the center of mass of link i with respect to the i^{th} coordinate system.

$\mathbf{I}_i$ = the inertia matrix about center of mass of link i with respect to the i^{th} coordinate system.

$\mathbf{F}_i$ = the total external force vector exerted on link i with respect to the i^{th} coordinate system.

$\mathbf{N}_i$ = the total external moment vector exerted on link i with respect to the i^{th} coordinate system.

$\mathbf{f}_i$ = the force vector exerted on link i by link i-1 .

$\mathbf{n}_i$ = the moment vector exerted on link i by link i-1 .

τ_i = the applied torque exerted on link i

4. Computed Torque Technique Based on the Newton-Euler Formulations

Given the equations of motion of a manipulator as in Eqs. (3.7)-(3.10) (Lagrange-Euler formulation) or Eqs. (3.11)-(3.19) (Newton-Euler formulation), the control problem is to find appropriate feedback torques/forces to servo all the joints of the manipulator in real-time to track a desired position trajectory as closely as possible. Several methods are available in accomplishing this task. Most notably of these are: (i) Resolved Motion Rate Control (RMRC)[Whi69], (ii) Cerebellar Model Articulation Controller (CMAC)[Alb75], (iii) Near-minimum-time control [KaB71], and (iv) Computed torque technique [Mar73, Pau72].

The RMRC is a technique for determining the joint angle rates required to cause a manipulator end point (or tool) to move in the directions which are expressed in the hand or world coordinate system. In order to find the required $\dot{\vartheta}$ the inverse Jacobian matrix $J(\vartheta)^{-1}$ is required. One of the drawbacks of this method is the added computation load needed to find the inverse Jacobian matrix and the singularity problem associated with the matrix inversion.

The CMAC is a table look-up control method which based on neuro-physiological theory. It computes control functions by referring to a table stored in the computer memory rather than by solution of analytic equations. For useful applications several problems such as memory size management and accuracy need to be solved.

Due to the nonlinearity and complexity of the dynamical model of manipulator, a closed form solution of the optimal control is very difficult, if not impossible. Near-minimum-time control is based on the linearization of the equations of motion about the nominal trajectory and linear feedback and/or suboptimal control law are obtained analytically. This control method is still too complex to be useful for manipulators with four or more degree of freedom and furthermore it neglects the effect of unknown external loads.

One of the basic control schemes is the computed torque technique [Pau72,Mar73] based on the Lagrange-Euler equations of motion.

The computed torque technique assumes that one can accurately compute the counterparts of $\mathbf{D}(\vartheta)$, $\mathbf{H}(\vartheta , \dot{\vartheta})$, and $\mathbf{G}(\vartheta)$ in Eq. (3.7) to minimize their nonlinear effects and use a position plus derivative control to servo the joints [Pau72]. Thus the structure of the control law has the form of:

$$\tau = \mathbf{D}_a(\vartheta)\left[\ddot{\vartheta}^d + \mathbf{K}_v(\dot{\vartheta}^d - \dot{\vartheta}) + \mathbf{K}_p(\vartheta^d - \vartheta)\right] + \mathbf{H}_a(\vartheta , \dot{\vartheta}) + \mathbf{G}_a(\vartheta) \tag{4.1}$$

where

$\mathbf{K}_v$ is an $n \times n$ derivative feedback gain matrix.

$\mathbf{K}_p$ is an $n \times n$ position feedback gain matrix.

n is the number of degree of freedom of a manipulator.

Substituting τ from Eq. (4.1) into Eq. (3.7), we have:

$$\begin{aligned}\mathbf{D}(\vartheta)\ddot{\vartheta} + \mathbf{H}(\vartheta,\dot{\vartheta}) + \mathbf{G}(\vartheta) = \mathbf{D}_a(\vartheta)&\left[\ddot{\vartheta}^d + \mathbf{K}_v(\dot{\vartheta}^d - \dot{\vartheta}) + \mathbf{K}_p(\vartheta^d - \vartheta)\right] \\ &+ \mathbf{H}_a(\vartheta , \dot{\vartheta}) + \mathbf{G}_a(\vartheta)\end{aligned} \tag{4.2}$$

If $\mathbf{D}_a(\vartheta)$, $\mathbf{H}_a(\vartheta , \dot{\vartheta})$, $\mathbf{G}_a(\vartheta)$ are equal to $\mathbf{D}(\vartheta)$, $\mathbf{H}(\vartheta , \dot{\vartheta})$, $\mathbf{G}(\vartheta)$ respectively, then Eq. (4.2) reduces to:

$$\mathbf{D}(\vartheta)\left[\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}\right] = 0 \tag{4.3}$$

where $\mathbf{e} = \vartheta^d - \vartheta$ and $\dot{\mathbf{e}} = \dot{\vartheta}^d - \dot{\vartheta}$

Since $\mathbf{D}(\vartheta)$ is always non-singular, if the values of $\mathbf{K}_v$ and $\mathbf{K}_p$ are chosen so that the characteristic roots of Eq. (4.3) have negative real parts, then the position error vector $\mathbf{e}$ approaches zero asymptotically.

However, the computation of the joint torques from Eq. (4.1) is very inefficient if the dynamic model is based on the complete Lagrange-Euler equations of motion. Because of this reason, it is common to simplify Eq. (4.1) by neglecting the velocity-related coupling term $H_a(\vartheta,\dot{\vartheta})$ and the off-diagonal elements of the acceleration-related matrix $D_a(\vartheta)$. That is, the structure of the control law has the form of:

$$\tau = diag[\ D_a(\vartheta)\ ]\left[\ddot{\vartheta}^d + K_v(\dot{\vartheta}^d - \dot{\vartheta}) + K_p(\vartheta^d - \vartheta)\right] + G_a(\vartheta) \qquad (4.4)$$

In the next section we will show the effects of neglecting these terms when the controller (as in Eq. (4.4)) is based on the simplified Lagrange-Euler equations of motion.

In order to utilize the complete equations of motion, an analogous control law derived from the computed torque based on the Newton-Euler equations of motion is proposed. The analogous control law can be obtained by substituting $\ddot{\vartheta}_i$ in Eqs. (3.11)-(3.19).

$$\ddot{\vartheta}_i^d + \sum_{s=1}^{n} K_v^{is}(\dot{\vartheta}_s^d - \dot{\vartheta}_s) + \sum_{s=1}^{n} K_p^{is}(\vartheta_s^d - \vartheta_s)$$

or (4.5)

$$\ddot{\vartheta}_i^d + \sum_{s=1}^{n} K_v^{is}\dot{e}_s + \sum_{s=1}^{n} K_p^{is}e_s$$

where K_v^{is} and K_p^{is} are the derivative and position feedback gains for joint i respectively and $e_s = \vartheta^d{}_s - \vartheta_s$ is the position error for joint s.

The values of feedback gain matrices K_v and K_p can be determined systematically as follow:

If K_v is a symmetric and semi-positive definite matrix and K_p is a symmetric and positive definite matrix, and the rank of $\left[K_v \mid K_pK_v \mid \cdots \mid K_p^{n-1}K_v\right] = n$, then the position error vector e approaches zero asymptotically.

[LCT82,TML80] show the equivalence of the proposed recursive controller and the control law obtained by the computed torque technique based on the Lagrange-Euler equations of motion. Since the recursive controller is based on the complete dynamic equations, it is expected that the performance of this controller is always better than the controller based on the simplified Lagrange-Euler equations of motion.

In the remaining of this section, the computational complexity of the proposed recursive control law based on the Newton-Euler equations of motion and the analogous control law obtained from the Lagrange-Euler equations of motion is tabulated. Also the feasibility of the real time control of a PUMA robot arm using the recursive controller is discussed.

As a mean of comparing their computational complexity, their efficiency is determined based on the number of mathematical operations (multiplications and additions) in terms of the number of joints of the robot arm, n . The number of mathematical operations of some of the terms in both control laws may be slightly different from other papers [TML80,Hol80] due to the method of implementation of the control algorithms in programming.

In this study, the homogeneous transformation matrices T^{i}_{i-1} are computed first and then other relevant terms such as the velocity-related, the acceleration-related and the gravity loading terms in the Lagrange-Euler equations of motion are computed respectively. The number of mathematical operations of the control laws based on these two formulations are tabulated in Table 1 and Table 2. In general, for a six-jointed robot arm with rotary joints, the number of mathematical operations in the control law as in Eq. (4.1) based on the Lagrangian formulation is about 100 times more than that of the Newton-Euler formulation.

Based on a PDP 11/45 computer and its manufacturer's specification sheet, an ADD (integer addition) instruction requires 300 *ns* and a MUL (integer multiply) instruction requires 3.3 μs. If we assume that for each ADD and MUL instruction, we need to fetch data from the core memory and the memory cycle time is 450 *ns* , then the proposed recursive control law based on the Newton-Euler equations of motion

Controller based on Lagrange-Euler Ethetauations of Motion	Multiplications	Additions
$\mathbf{T}_j^{l}$	$32n(n-1)$	$24n(n-1)$
$-m_j\mathbf{g}\dfrac{\partial\mathbf{T}_o^{j}}{\partial\vartheta_l}\bar{\mathbf{r}}_j$	$4n(9n-7)$	$n\dfrac{(51n-45)}{2}$
$\sum_{j=l}^{n} m_j\mathbf{g}\dfrac{\partial\mathbf{T}_o^{j}}{\partial\vartheta_l}\bar{\mathbf{r}}_j$	0	$\dfrac{1}{2}n(n-1)$
$Tr\left\{\dfrac{\partial\mathbf{T}_o^k}{\partial\vartheta_j}\mathbf{J}_k\left(\dfrac{\partial\mathbf{T}_o^k}{\partial\vartheta_l}\right)^t\right\}$	$\dfrac{128}{3}n(n+1)(n+2)$	$\dfrac{65}{2}n(n+1)(n+2)$
$\sum_{k=max(l,j)}^{n} Tr\left\{\dfrac{\partial\mathbf{T}_o^k}{\partial\vartheta_j}\mathbf{J}_k\left(\dfrac{\partial\mathbf{T}_o^k}{\partial\vartheta_l}\right)^t\right\}$	0	$\dfrac{1}{6}n(n-1)(n+1)$
$Tr\left\{\dfrac{\partial^2\mathbf{T}_o^m}{\partial\vartheta_j\partial\vartheta_k}\mathbf{J}_m\left(\dfrac{\partial\mathbf{T}_o^m}{\partial\vartheta_l}\right)^t\right\}$	$\dfrac{128}{3}n^2(n+1)(n+2)$	$\dfrac{65}{2}n^2(n+1)(n+2)$
$\sum_{m=max(l,j,k)}^{n} Tr\left\{\dfrac{\partial^2\mathbf{T}_o^m}{\partial\vartheta_j\partial\vartheta_k}\mathbf{J}_m\left(\dfrac{\partial\mathbf{T}_o^m}{\partial\vartheta_l}\right)^t\right\}$	0	$\dfrac{1}{6}n^2(n-1)(n+1)$
$\ddot{\vartheta}^d + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}$	$2n$	$4n$
$\tau = \mathbf{D}_a(\ddot{\vartheta}^d + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e})$ $+ \mathbf{H}_a(\vartheta,\vartheta) + \mathbf{G}_a(\vartheta)$	$n^2(n+2)$	$n^2(n+1)$
Total	$\dfrac{128}{3}n^4 + \dfrac{515}{3}n^3$	$\dfrac{98}{3}n^4 + \dfrac{787}{6}n^3$
Mathematical Operations	$+\dfrac{850}{3}n^2 + \dfrac{82}{3}n$	$+\dfrac{640}{3}n^2 + \dfrac{131}{6}n$

where n = number of degree-of-freedom of the robot arm

Table 1 Breakdown of Mathematical Operations of the Controller

Based on Lagrange-Euler Formulation

Controller based on Newton-Euler Ethetauations of Motion	Multiplications	Additions
ω_i	$9n$	$7n$
α_i	$9n$	$9n$
$\mathbf{a}_i$	$27n$	$22n$
$\bar{\mathbf{a}}_i$	$15n$	$14n$
$\mathbf{F}_i$	$3n$	0
$\mathbf{f}_i$	$9(n-1)$	$9n-6$
$\mathbf{N}_i$	$24n$	$18n$
$\mathbf{n}_i$	$21n-15$	$24n-15$
$\ddot{\vartheta}_i^d + K_v\dot{e}_i + K_p e_i$	$2n$	$4n$
Total Mathematical Operations	$119n-24$	$107n-21$

where n = number of degree-of-freedom of the robot arm

Table 2 Breakdown of Mathematical Operations of the Controller Based on Newton-Euler Formulation

requires approximately 3 *ms.* to compute the necessary joints torques to servo all the joints of a PUMA robot arm for a trajectory set point. This certainly is quite acceptable for the time delay in the servo loop and thus allows one to perform real-time feedback control on a PUMA robot arm with all its dynamics taken into consideration.

5. Computer Simulation Results

This section discusses the computer simulation result of the proposed recursive control law and compares it with that of the simplified Lagrange-Euler equation of motion.

A computer simulation study to evaluate the performance of the above control laws for a PUMA robot arm was carried out on a VAX-11/780 computer. In this simulation, the six-jointed manipulator moves from an initial joint angles $\vartheta_{initial} = (0^\circ, 45^\circ, 45^\circ, 0^\circ, 0^\circ, 0^\circ)^T$ to a final joint angles $\vartheta_{final} = (90^\circ, -45^\circ, 135^\circ, 90^\circ, 90^\circ, 90^\circ)^T$. The required time for this motion is 1 second. In this trajectory, the PUMA robot arm is fully stretched at 0.5 seconds. At this position, $\vartheta_{0.5sec} =$ ($45^\circ, 0^\circ, 90^\circ, 45^\circ, 45^\circ, 45^\circ$), the torques due to the gravity have the maximum values and the absolute values of joint velocity of the arm also becomes the maximum. The accelerations are sharply changed from the maximum values to the minimum values or vice versa. The sampling time is chosen to be 0.01 second.

The feedback gain matrices K_v and K_p of the control law are kept constant for the whole motion execution to facilitate the comparison of both control laws. The elements of K_v and K_p are assigned according to the stability criterion as outlined in Eq. (4.3). The principal diagonal elements of K_p are assigned the value of 100 and the diagonal elements of K_v to $2\sqrt{K_p} = 20$. Again to simplify the comparison, all the non-diagonal elements of K_v and K_p are zeró which neglect the position and derivative error effects between joints.

With reference to the dynamic equations of motion as in Eqs. (3.11) - (3.19) and Eq. (3.6) the numerical values used in this simulation are:

d_1 = 0.664 *meter*

a_2 = 0.432 *meter*

d_2 = 0.1495 *meter*

d_4 = 0.432 *meter*

d_6 = 0.130 *meter*

$\mathbf{m}$ = [2.27, 15.91, 6.82, 3.18, 0.91, 0.45] *Kg*

daig $\mathbf{I}_1$ = [0.0071, 0.0267, 0.0267] *Kg–meter*^{2}

daig $\mathbf{I}_2$ = [0.1000, 0.7300, 0.8025] *Kg–meter*^{2}

daig $\mathbf{I}_3$ = [0.0222, 0.2160, 0.2245] *Kg–meter*^{2}

daig $\mathbf{I}_4$ = [0.0020, 0.0010, 0.0010] *Kg–meter*^{2}

daig $\mathbf{I}_5$ = [0.0030, 0.0030, 0.0004] *Kg–meter*^{2}

daig $\mathbf{I}_6$ = [0.0050, 0.0050, 0.0003] *Kg–meter*^{2}

$\mathbf{r}_1 = [\, 0., -0.664, 0.]^T$ *meter*

$\mathbf{r}_2 = [\, 0.432, 0., 0.1495]^T$ *meter*

$\mathbf{r}_3 = [\, 0., 0., 0.]^T$ *meter*

$\mathbf{r}_4 = [\, 0., -0.432, 0.]^T$ *meter*

$r_5 = [\ 0., 0., 0.]^T$ *meter*

$r_6 = [\ 0., 0., 0.13]^T$ *meter*

$\bar{r}_1 = [\ 0., 0, 0.073]^T$ *meter*

$\bar{r}_2 = [\ -0.432, 0., 0.]^T$ *meter*

$\bar{r}_3 = [\ 0., 0., 0.1]^T$ *meter*

$\bar{r}_4 = [\ 0., 0, 0.1]^T$ *meter*

$\bar{r}_5 = [\ -0., 0., 0.01]^T$ *meter*

$\bar{r}_6 = [\ 0., 0., -0.05]^T$ *meter*

Figure 2 shows the flow-chart for the computer simulation program implementation. Figures 3-5 show the preplanned position, velocity and acceleration trajectory for joint 1. In this simulation each link moves 90 degrees from its original position. It is expected that velocity trajectory and acceleration trajectory for each joint are the same as joint 1 except joint 2 whose trajectories are reversed.

Since the complete Lagrange-Euler equations of motion require long computational time, it is common to simplify the Lagrange-Euler equations of motion by neglecting the off-diagonal terms in the acceleration-related matrix and the Coriolis and centrifugal terms. Though the simplified Lagrange-Euler equations of motion has an advantage for computational time, the neglected terms becomes significant when the robot arm is moving at high speeds. In order to show the effects of neglecting the off-diagonal elements of the acceleration-related term and the velocity-related terms (Coriolis and centrifugal), the applied torques have been computed for the following cases: (a) the Lagrange-Euler equations of motion with the off-diagonal elements of the acceleration-related matrix set to zero, (b) the Lagrange-Euler equations of motion without the Coriolis and centrifugal terms and (c) the

Lagrange-Euler equations of motion with both the off-diagonal elements in the acceleration-related matrix and the Coriolis and centrifugal terms set to zero. Figures 6-11 show the applied torques computed from the Newton-Euler equations of motion and from the case (a). Figures 12-17 show the applied torques from the Newton-Euler equations of motion and from the case (b). The applied torques from the Newton-Euler equations of motion and from the case (c) are shown in Figures 18-23. For this particular motion through a given trajectory (fast movement), the off-diagonal terms in the acceleration-related matrix are large and dominant. In joints 2,3,4 and 5, the differences between the applied torques computed from the recursive control law and the simplified control law are large and as a result it is expected to have large position errors in joints 2,3,4 and 5.

In Figures 24-41, the position errors between the two dynamic models, the Newton-Euler equations of motion and the simplified Lagrange-Euler equations of motion, are shown for each joint for various loading conditions. The recursive controller based on the Newton-Euler equations of motion always shows better performance for various loading conditions and various trajectories. Although the position errors from the proposed control technique are slightly "oscillatory" about the desired position set points, they are always small. The simulation results are tabulated in Table 3.

Since the manipulator is a highly nonlinear and complex system, further improvements in the performance of the control law can be done by using adaptive feedback gains. Our future work will focus on finding proper adaptive control strategies for industrial robots whose loads are varying within a task cycle time [LeC82].

		Simplified model		Complete model	
Various Loading Conditions	Joint	Trajectory Tracking		Trajectory Tracking	
		Max. Error (radian)	Max. Error (mm)	Max. Error (radian)	Max. Error (mm)
No Load	1	0.0194	19.40	0.0040	4.00
	2	0.0494	49.40	0.0070	7.00
	3	0.1882	94.10	0.0101	5.07
	4	0.4698	70.47	0.0062	0.94
	5	0.2278	34.17	0.0010	0.16
	6	0.0726	10.89	0.0039	0.59
1/2 Max. Load	1	0.0239	23.90	0.0046	4.60
	2	0.0748	74.80	0.0099	9.90
	3	0.2435	121.75	0.0151	7.53
	4	0.7565	113.47	0.0096	1.44
	5	0.3056	45.84	0.0014	0.21
	6	0.1605	24.08	0.0041	0.61
Max. Load	1	0.0261	26.10	0.0048	4.80
	2	0.0954	95.54	0.0122	12.20
	3	0.2812	140.60	0.0193	9.65
	4	0.8752	131.28	0.0122	1.83
	5	0.3523	52.85	0.0016	2.33
	6	0.2131	31.97	0.0053	0.79

Table 3 Comparison of Control Method based on Two Dynamic Models

(1) Set $i = 0$ where $i = i^{th}$ sampling period.

(2) Determine $\vartheta^d[i]$, $\dot{\vartheta}^d[i]$ and $\ddot{\vartheta}^d[i]$ from preplanned trajectory.

(3) Compute position and velocity errors.

$$\mathbf{e}[i] = \vartheta^d[i] - \vartheta^a[i]$$

$$\dot{\mathbf{e}}[i] = \dot{\vartheta}^d[i] - \dot{\vartheta}^a[i]$$

(4) Determine error signals with K_v, K_p feedback gain matrices.

$$\ddot{\vartheta}^d[i] + K_v\dot{\mathbf{e}}[i] + K_p\mathbf{e}[i]$$

(5) Forward recursive equations in Newton-Euler formulation.

(6) Backward recursive equations in Newton-Euler formulation to determine the applied torques $\tau[i]$.

(7) Compute coefficients of robot arm model using Lagrange-Euler formulation as in Eq. (3.6).

(8) Integrate the dynamic equation of a PUMA arm derived from Lagrange-Euler formulation using the 4th order Runge-Kutta method. The outputs are $\vartheta_a[i]$ and $\vartheta_a[i]$.

(9) Set $i = i + 1$ next sampling period.

(10) Is $i = N$? (Total of N sampling periods). If yes, stop. Else go to step (2).

Figure 2 Flow-Chart of Computer Simulation for the Proposed Controller

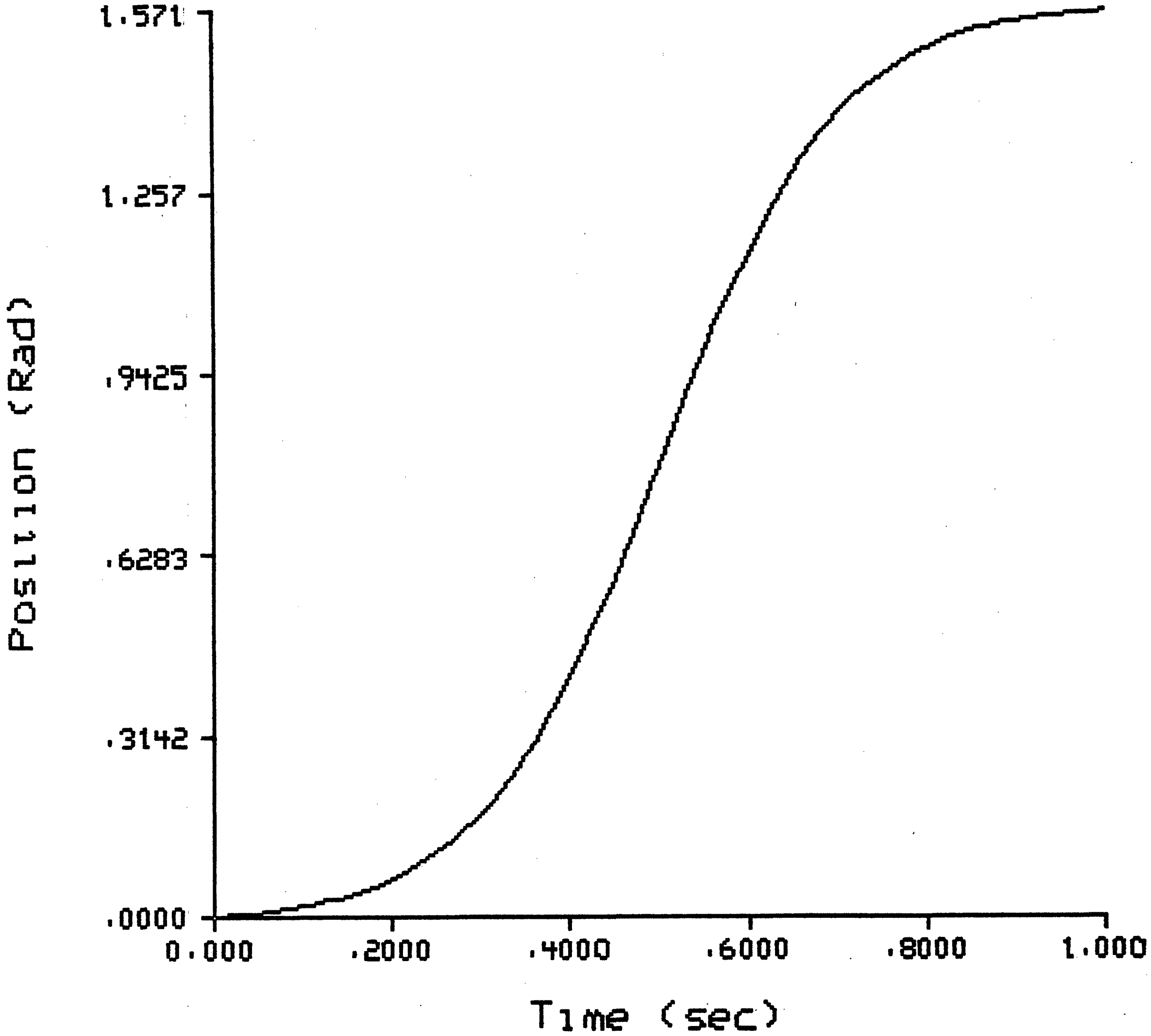

Figure 3 Position Trajectory (Joint 1)

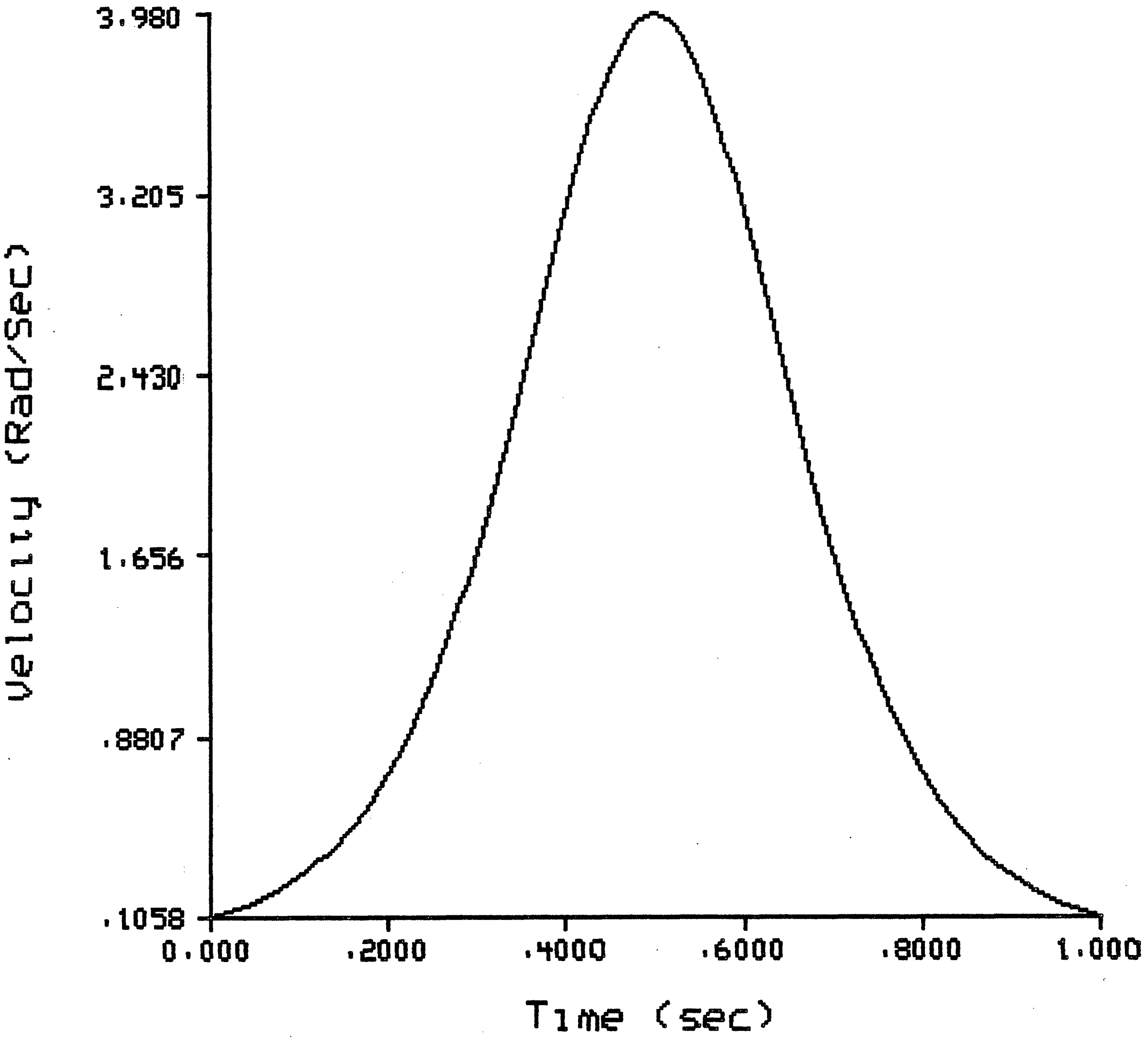

Figure 4 Velocity Trajectory (Joint 1)

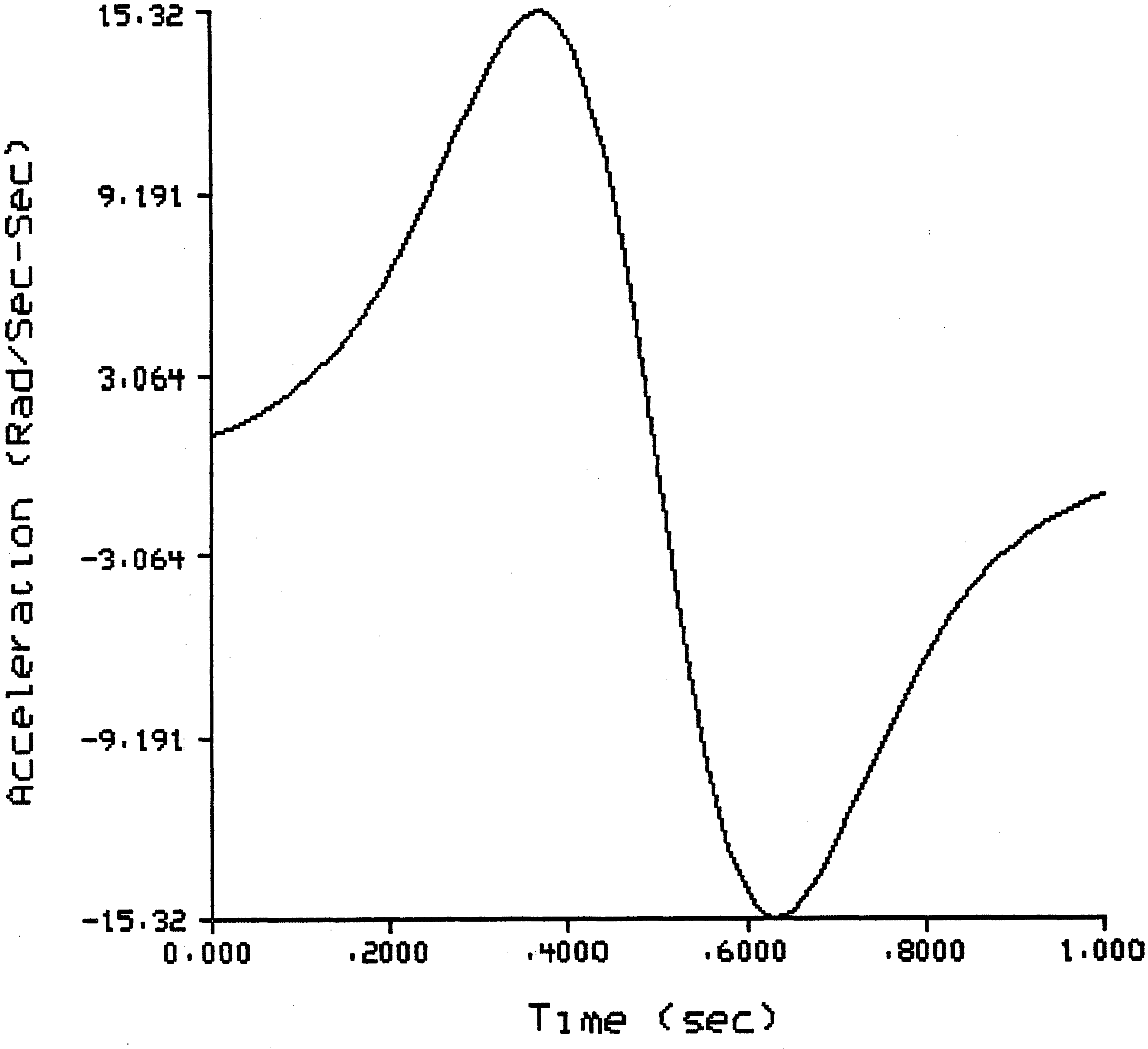

Figure 5 Acceleration Trajectory (Joint 1)

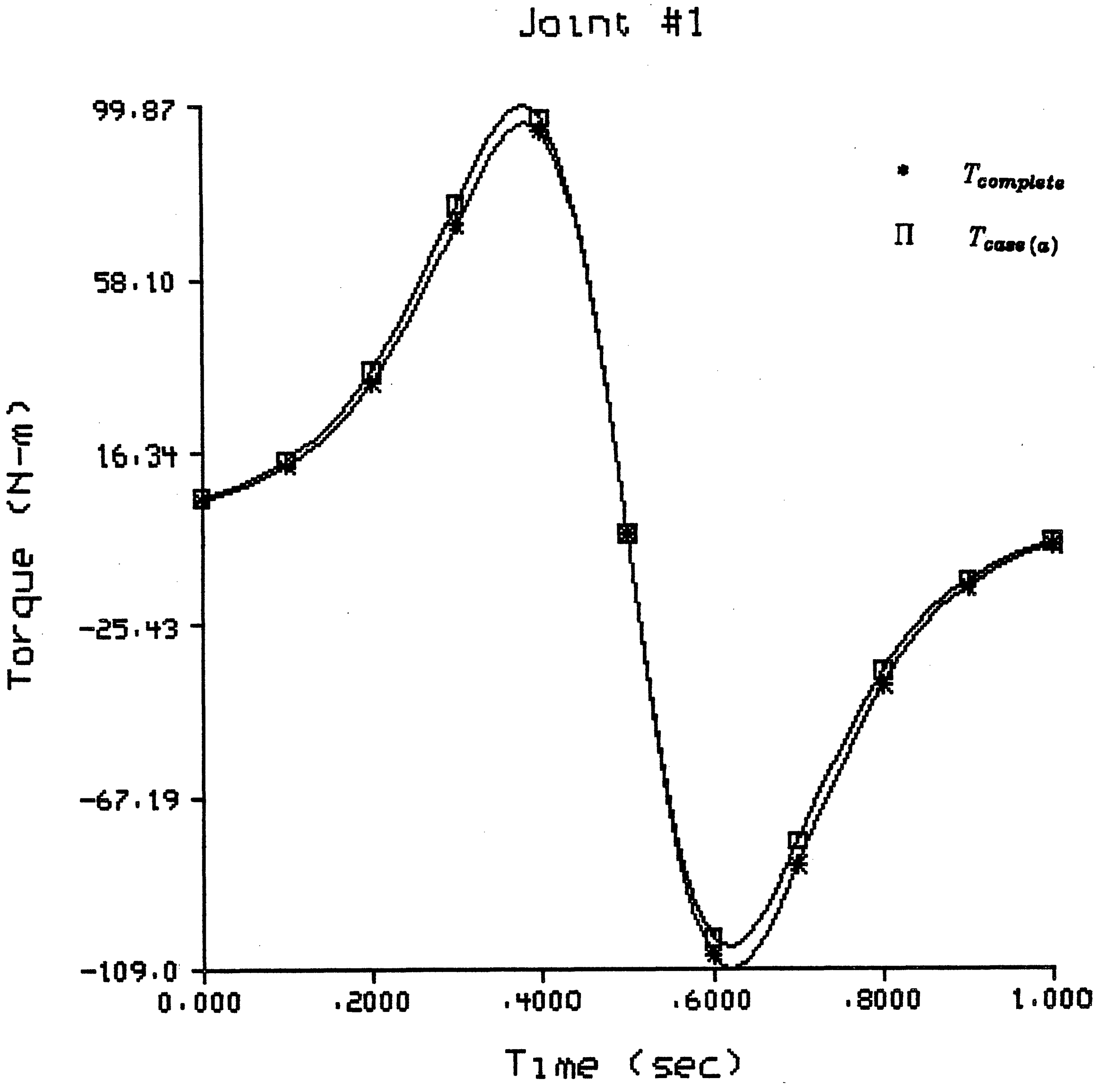

Figure 6 $T_{Complete}$ vs. Case (a)

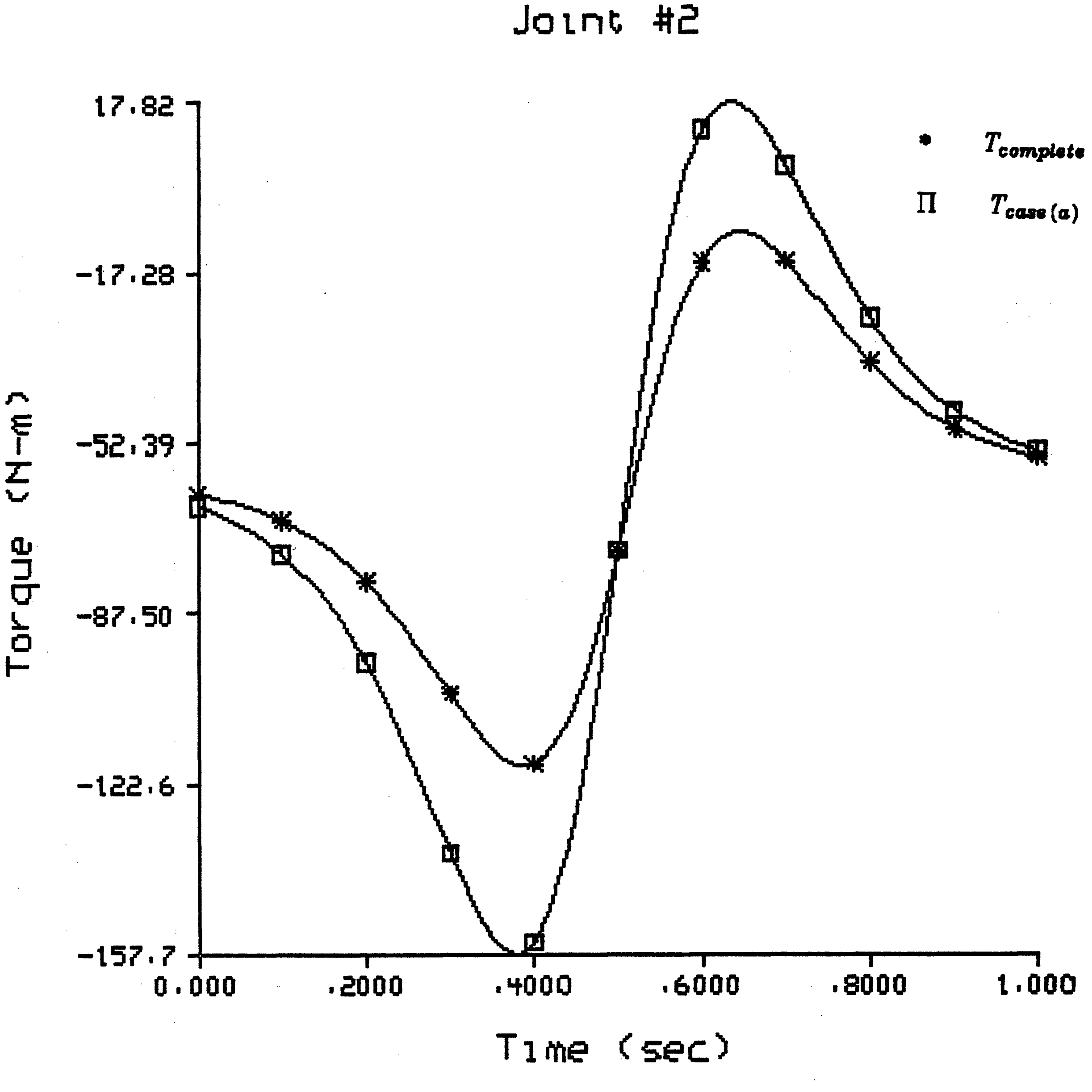

Figure 7 $T_{Complete}$ vs. Case (a)

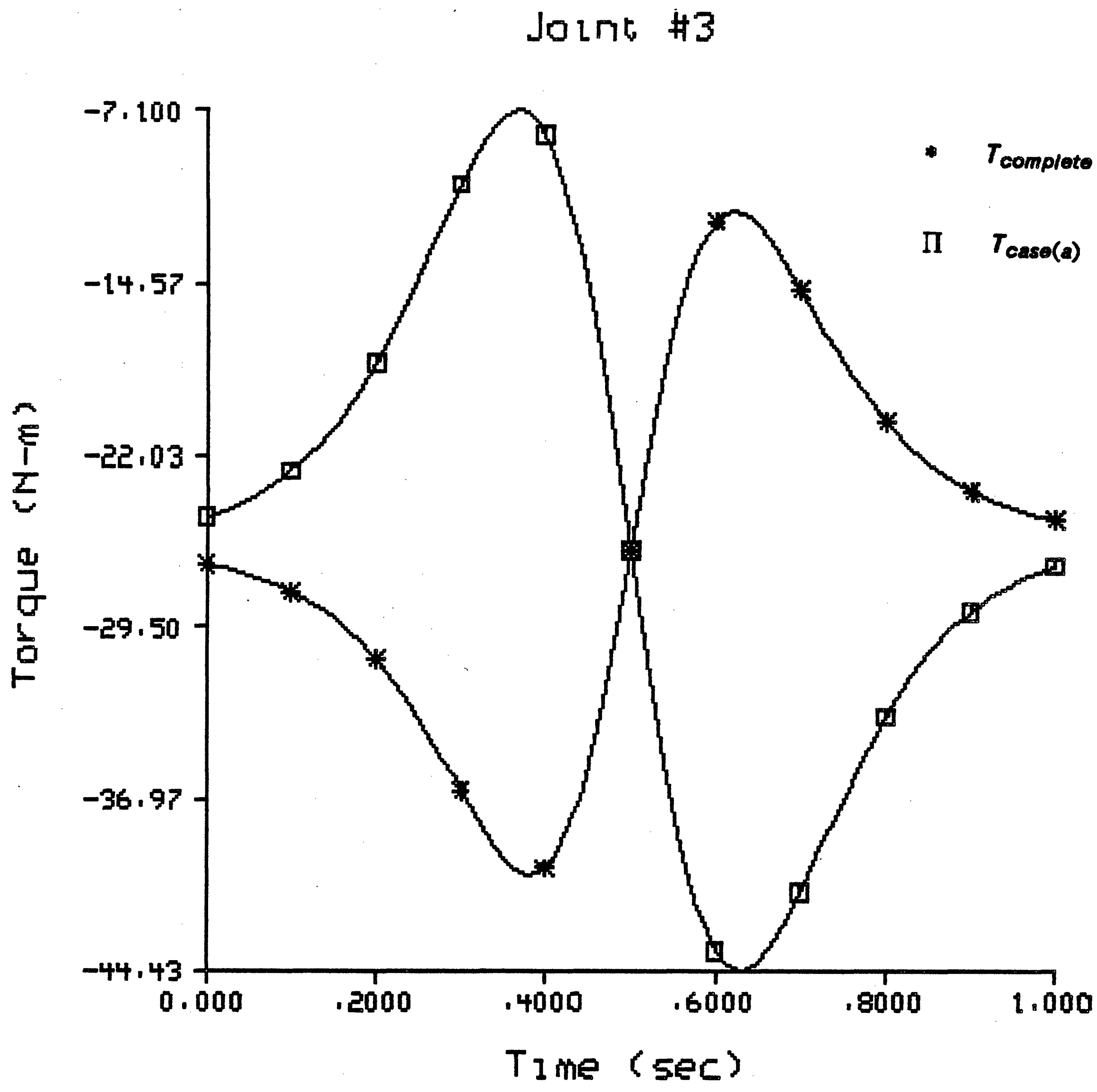

Figure 8 $T_{Complete}$ vs. Case (a)

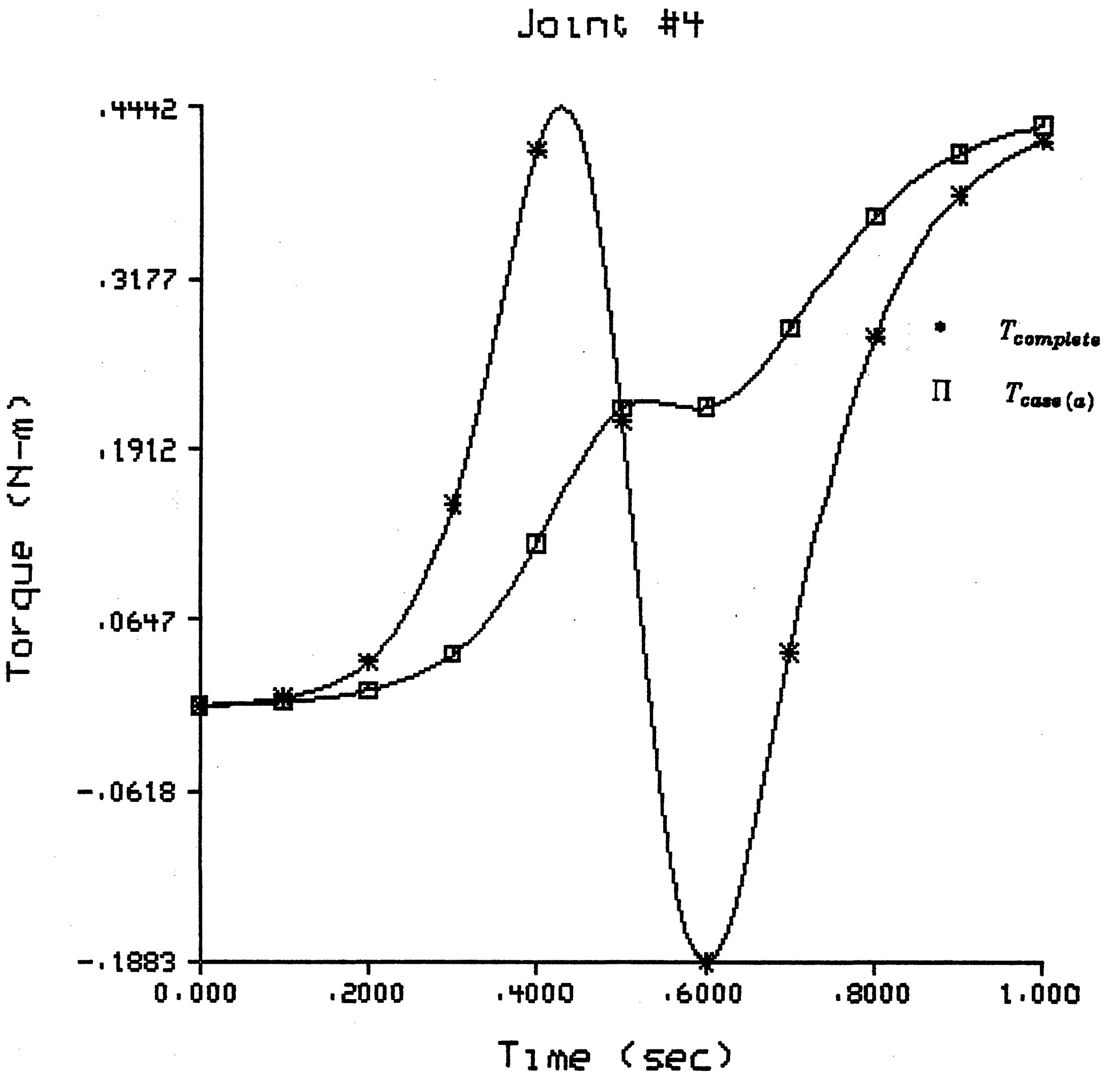

Figure 9 $T_{Complete}$ vs. Case (a)

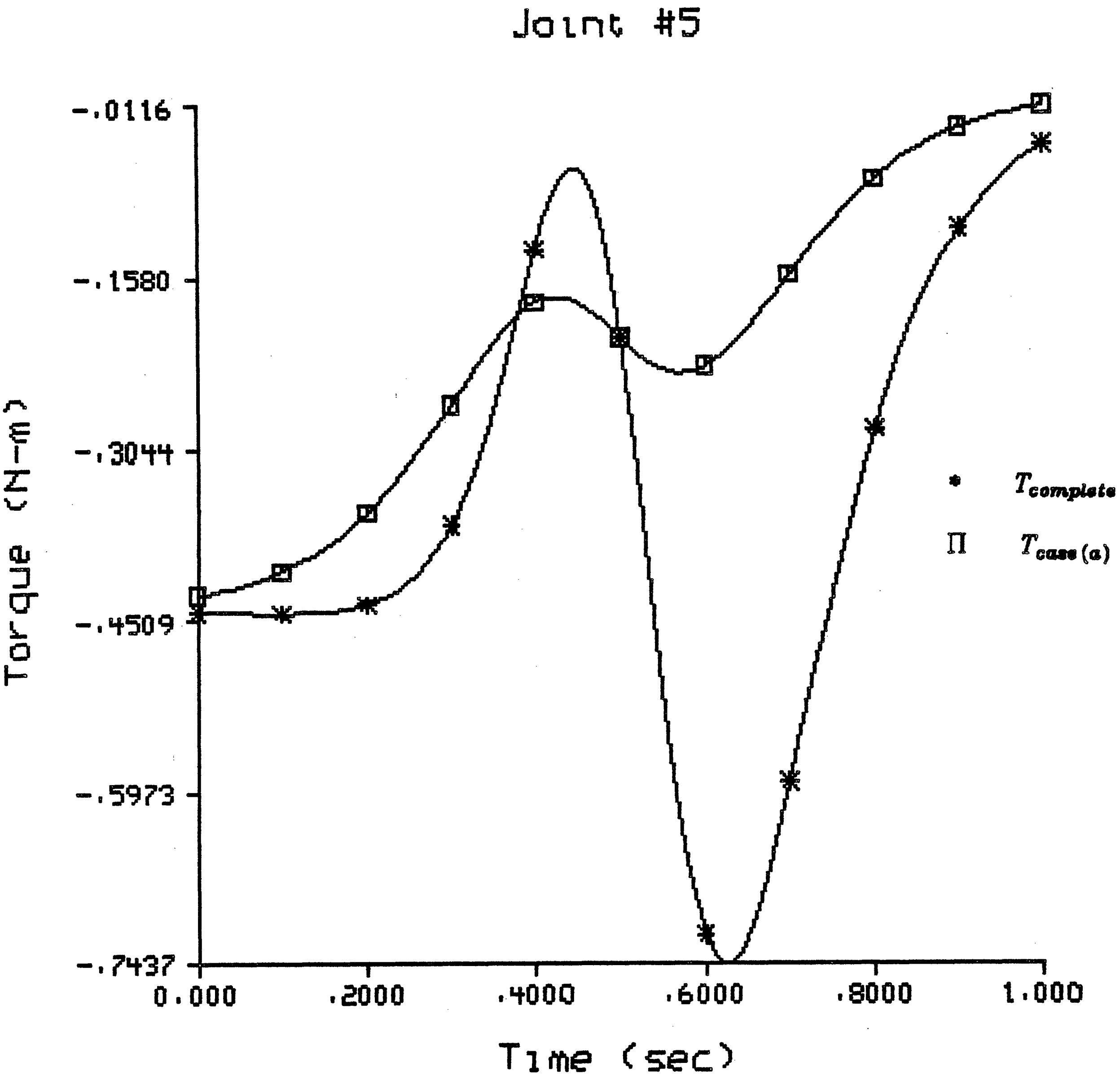

Figure 10 $T_{Complete}$ vs. Case (a)

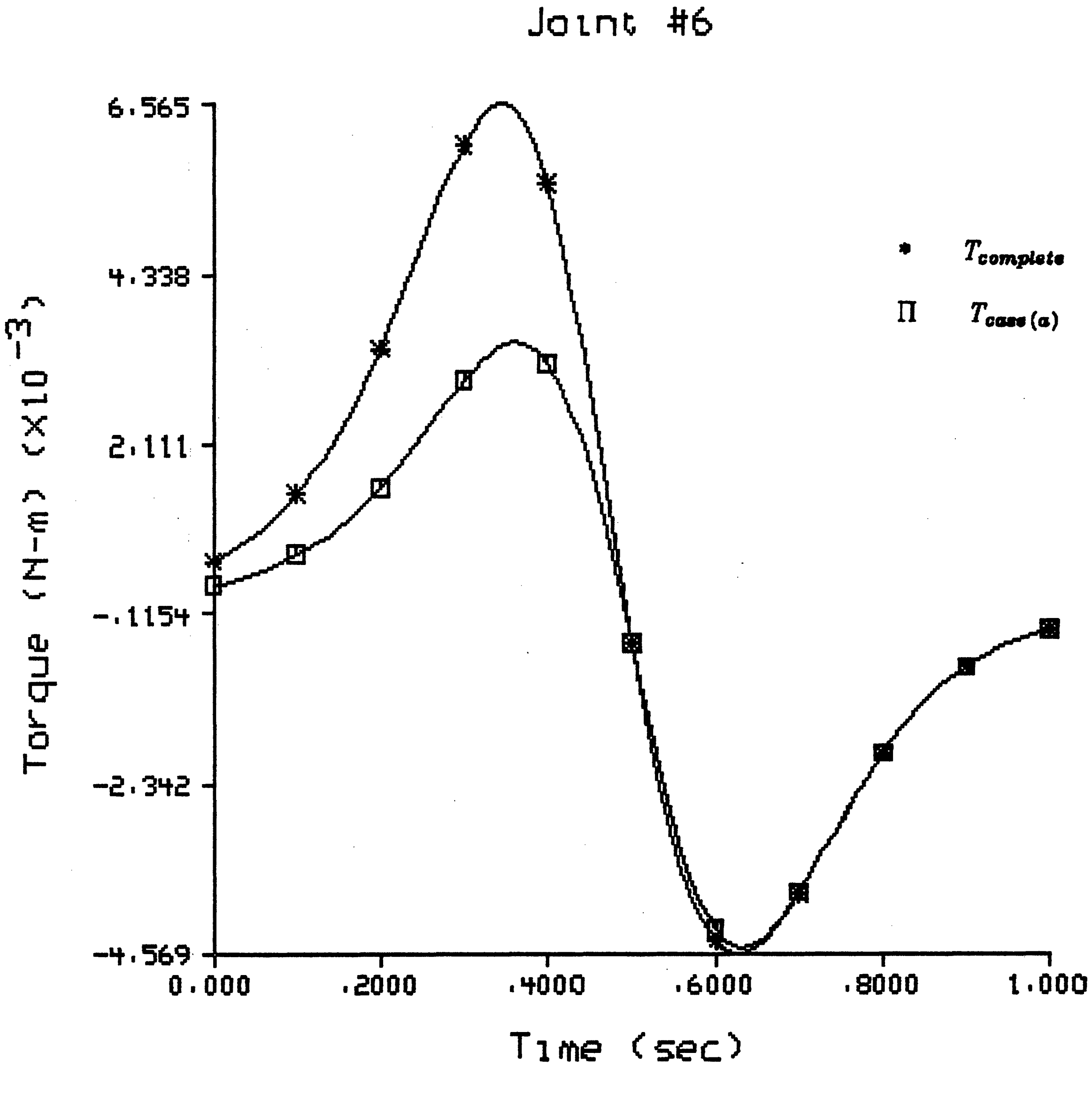

Figure 11 $T_{Complete}$ vs. Case (a)

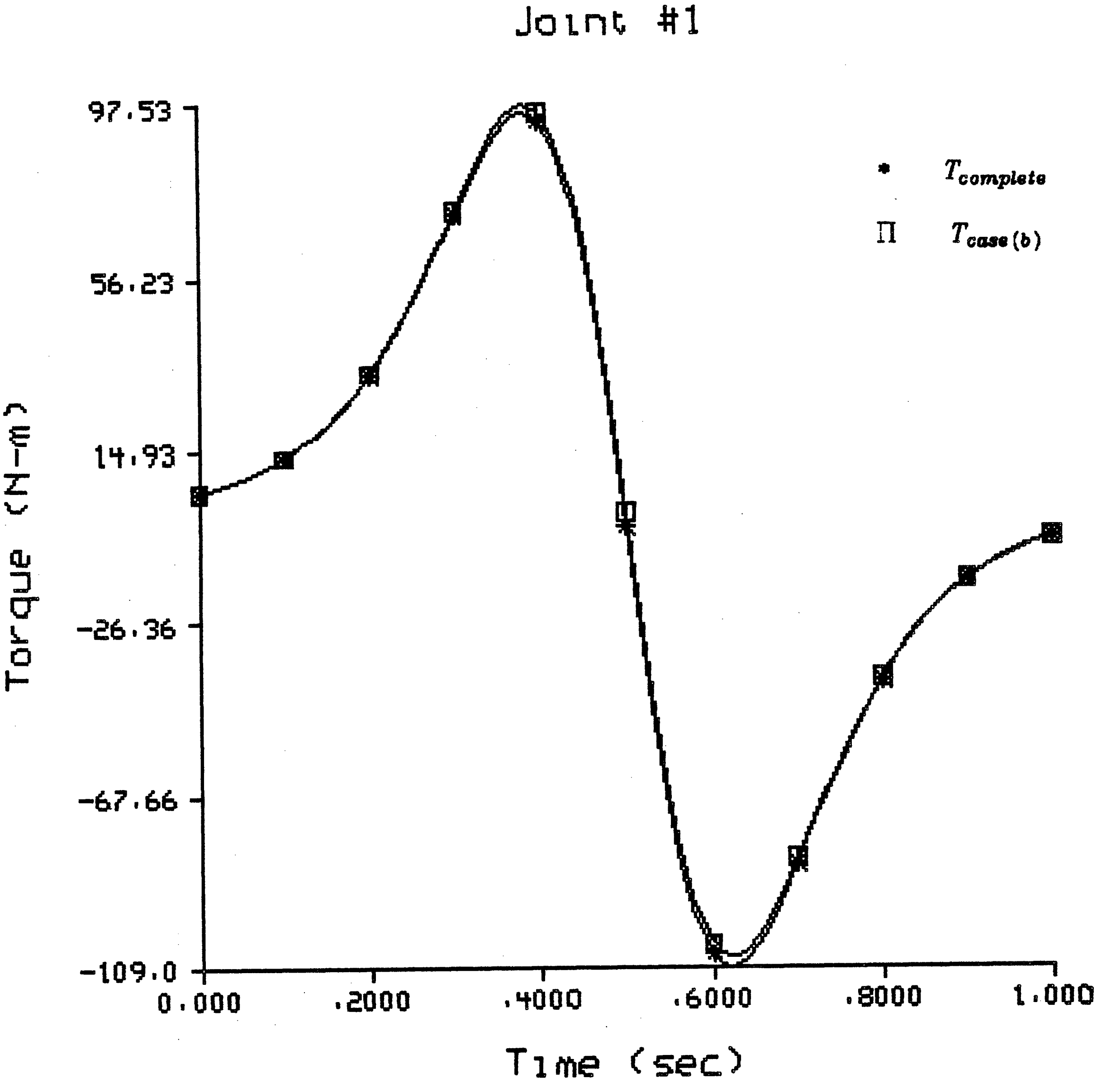

Figure 12 $T_{Complete}$ vs. Case (b)

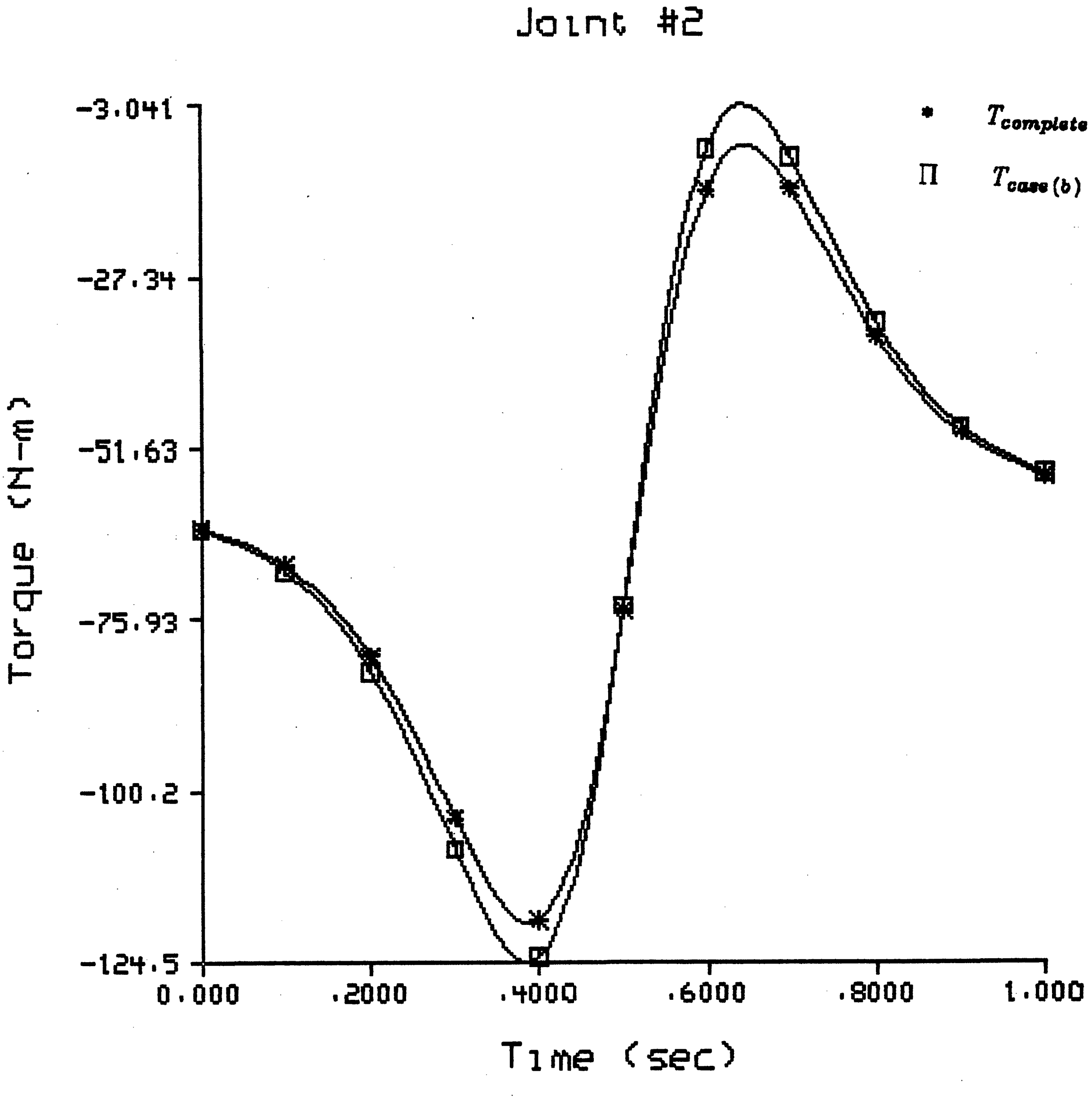

Figure 13 $T_{Complete}$ vs. Case (b)

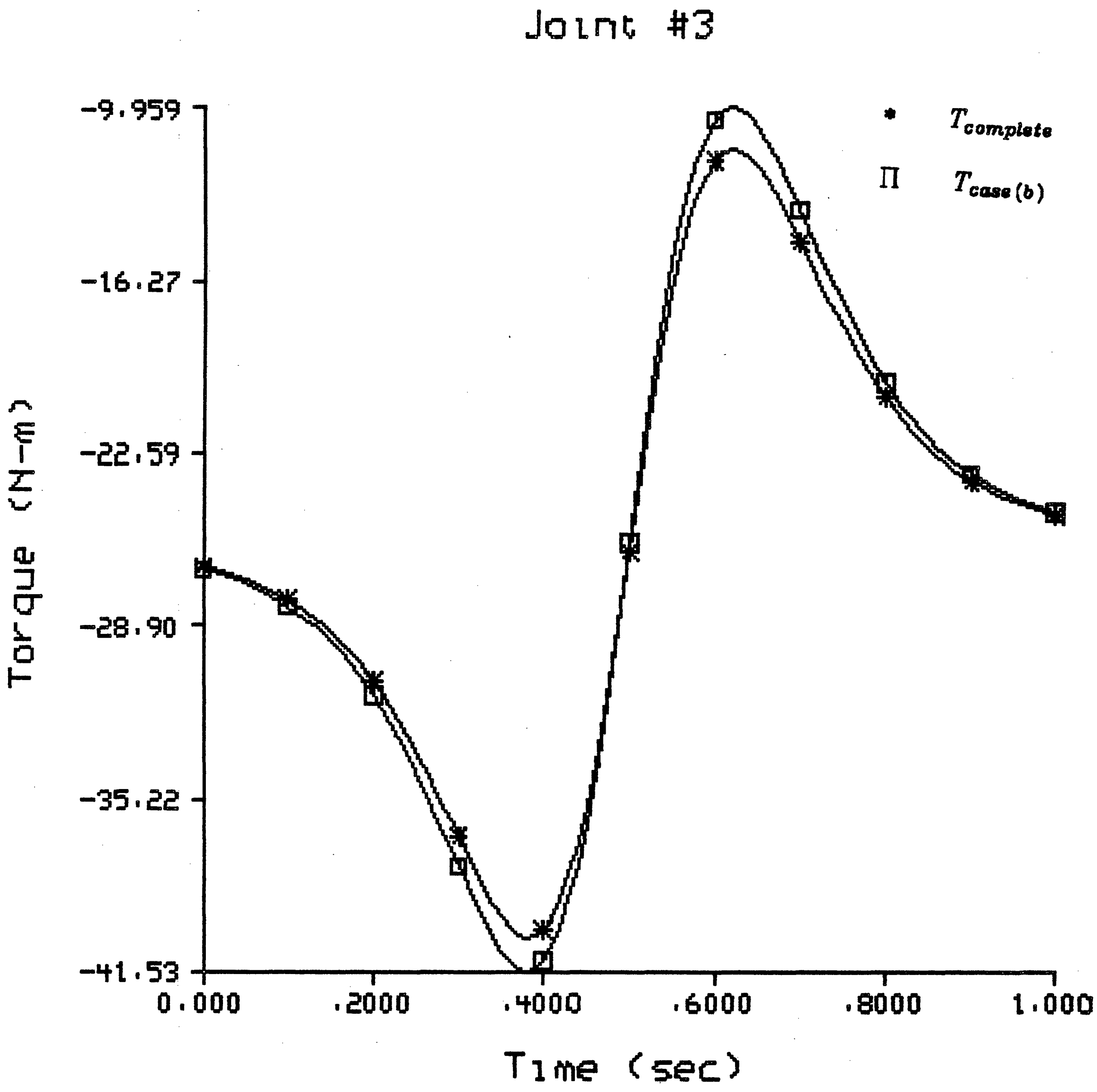

Figure 14 $T_{Complete}$ vs. Case (b)

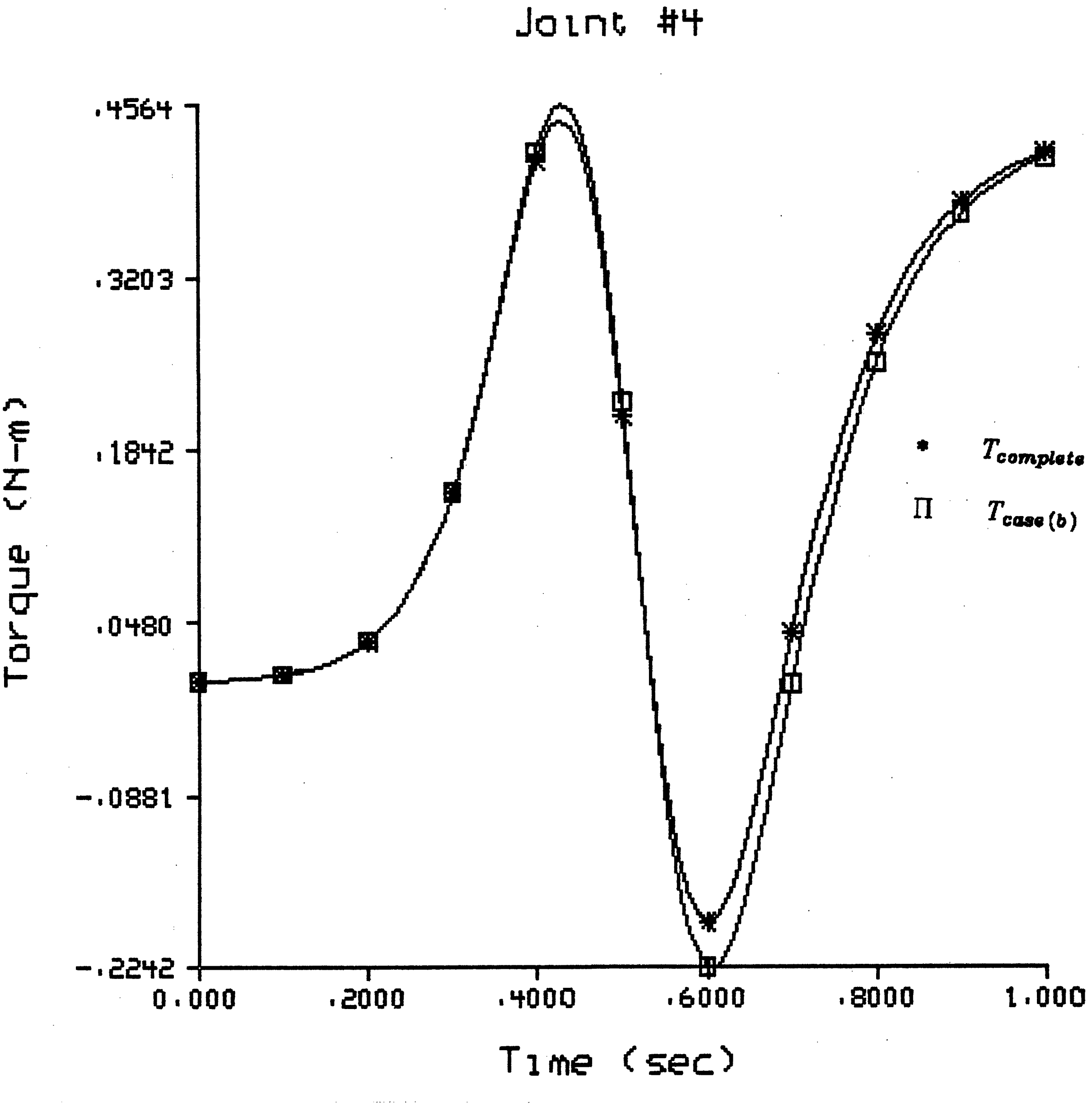

Figure 15 $T_{Complete}$ vs. Case (b)

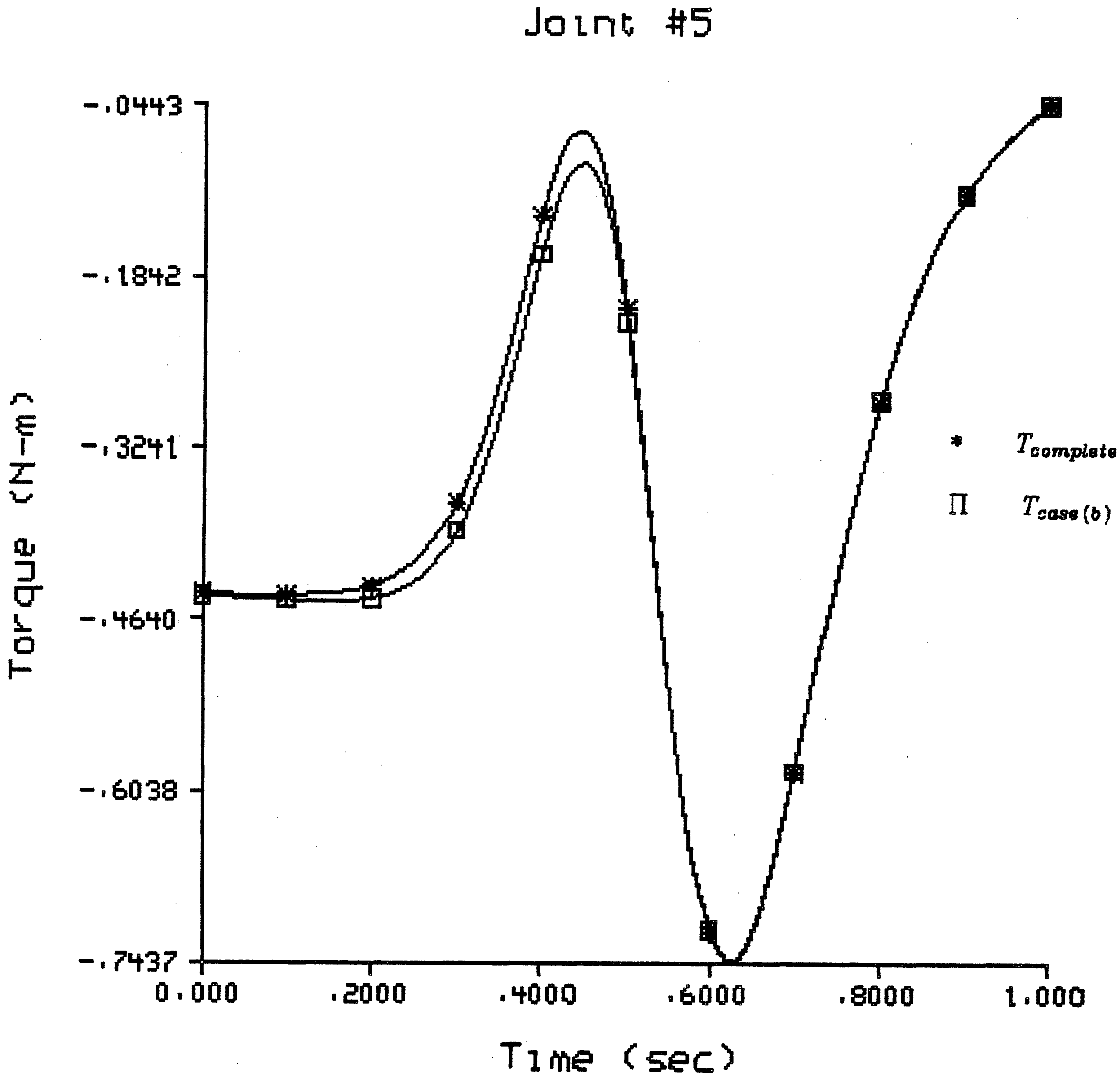

Figure 16 $T_{Complete}$ vs. Case (b)

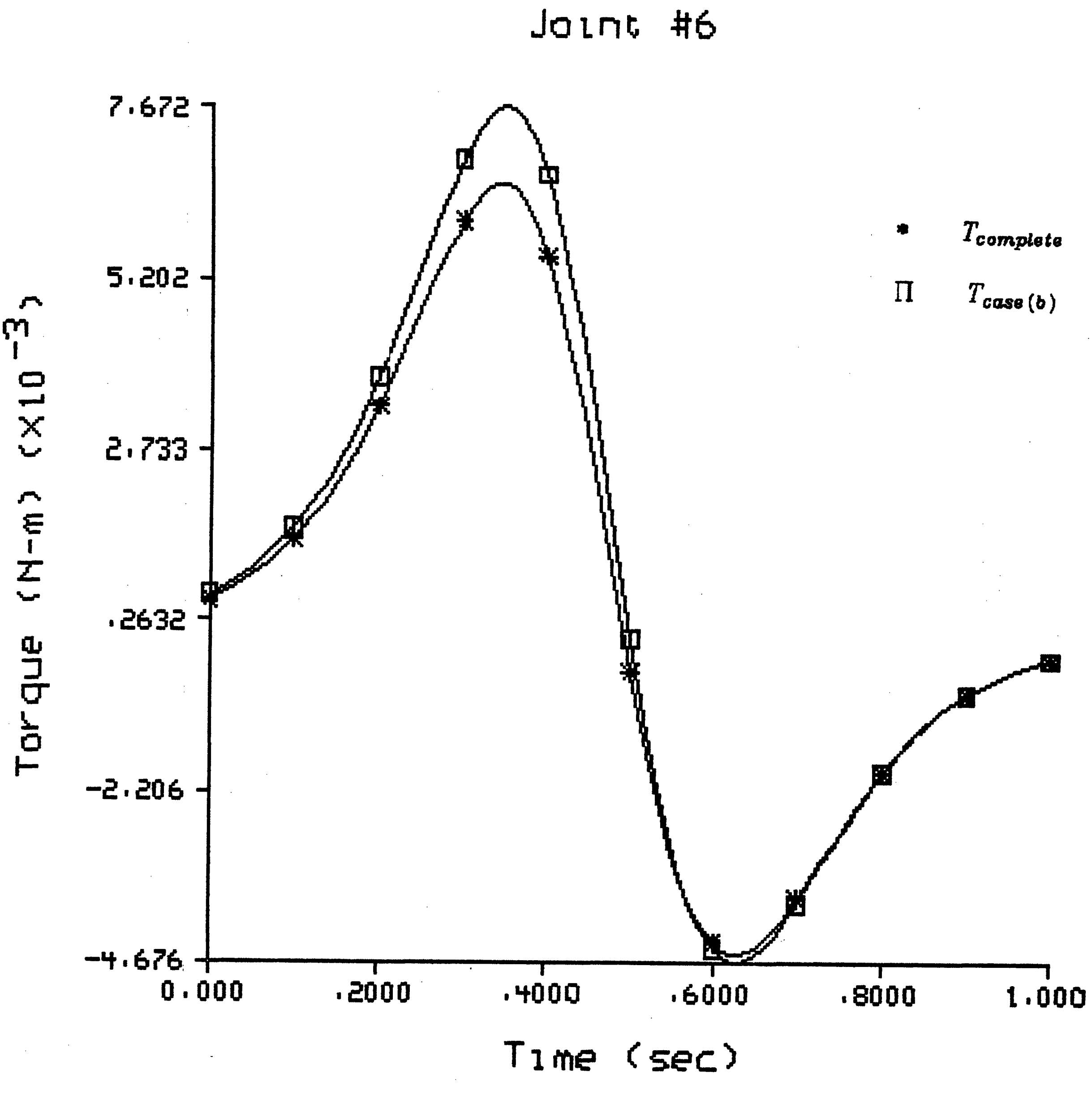

Figure 17 $T_{Complete}$ vs. Case (b)

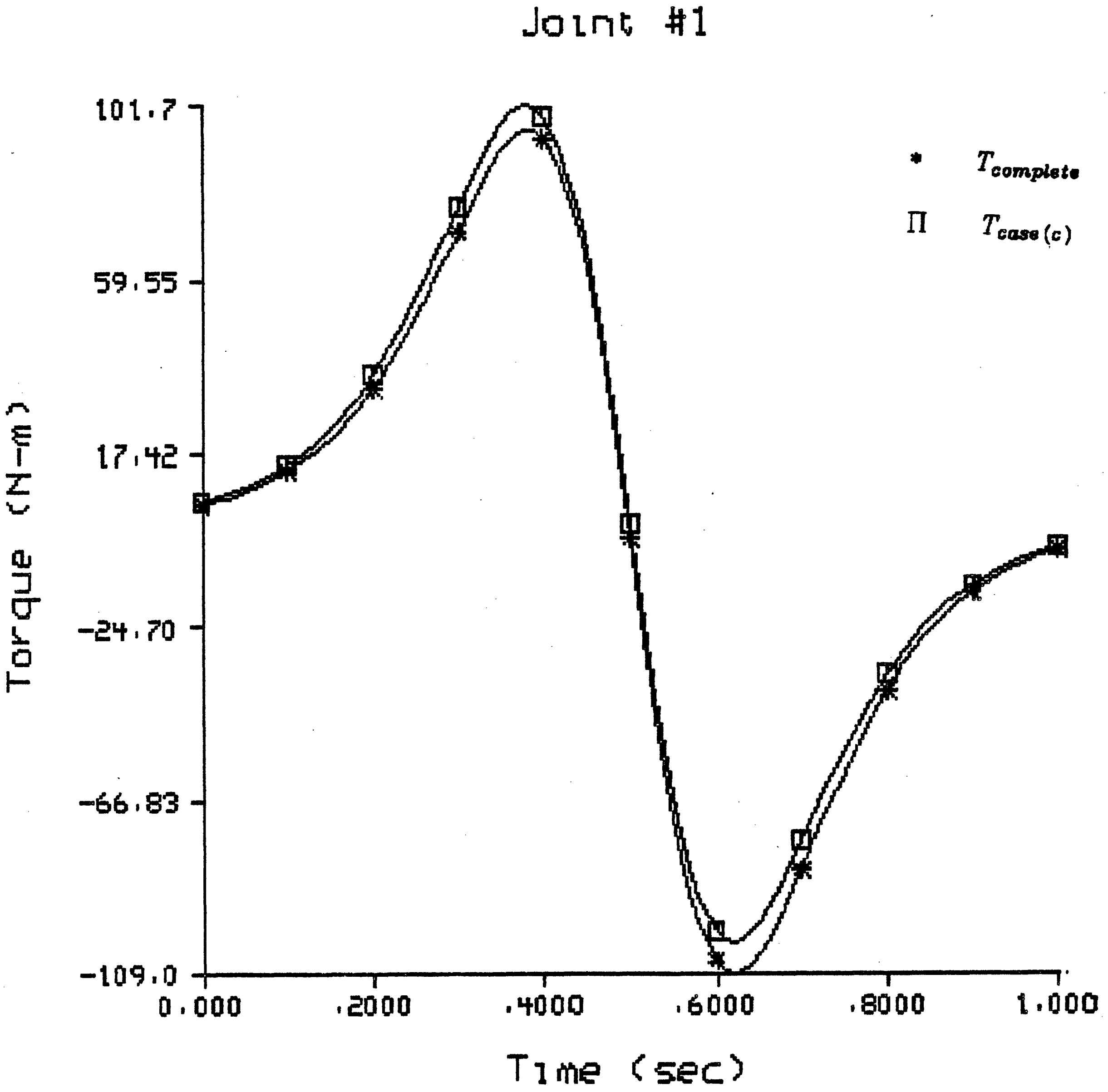

Figure 18 $T_{Complete}$ vs. Case (c)

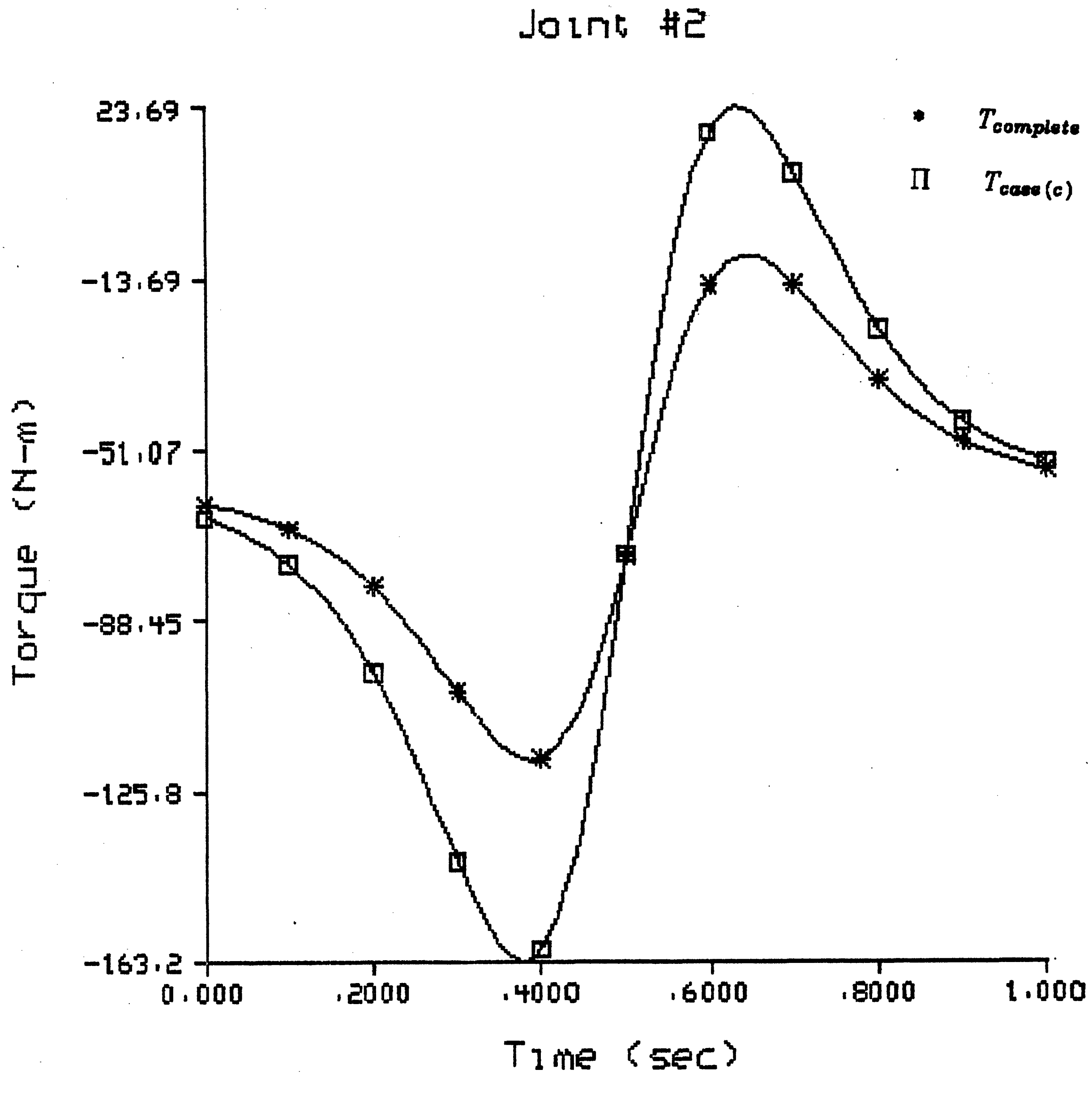

Figure 19 $T_{Complete}$ vs. Case (c)

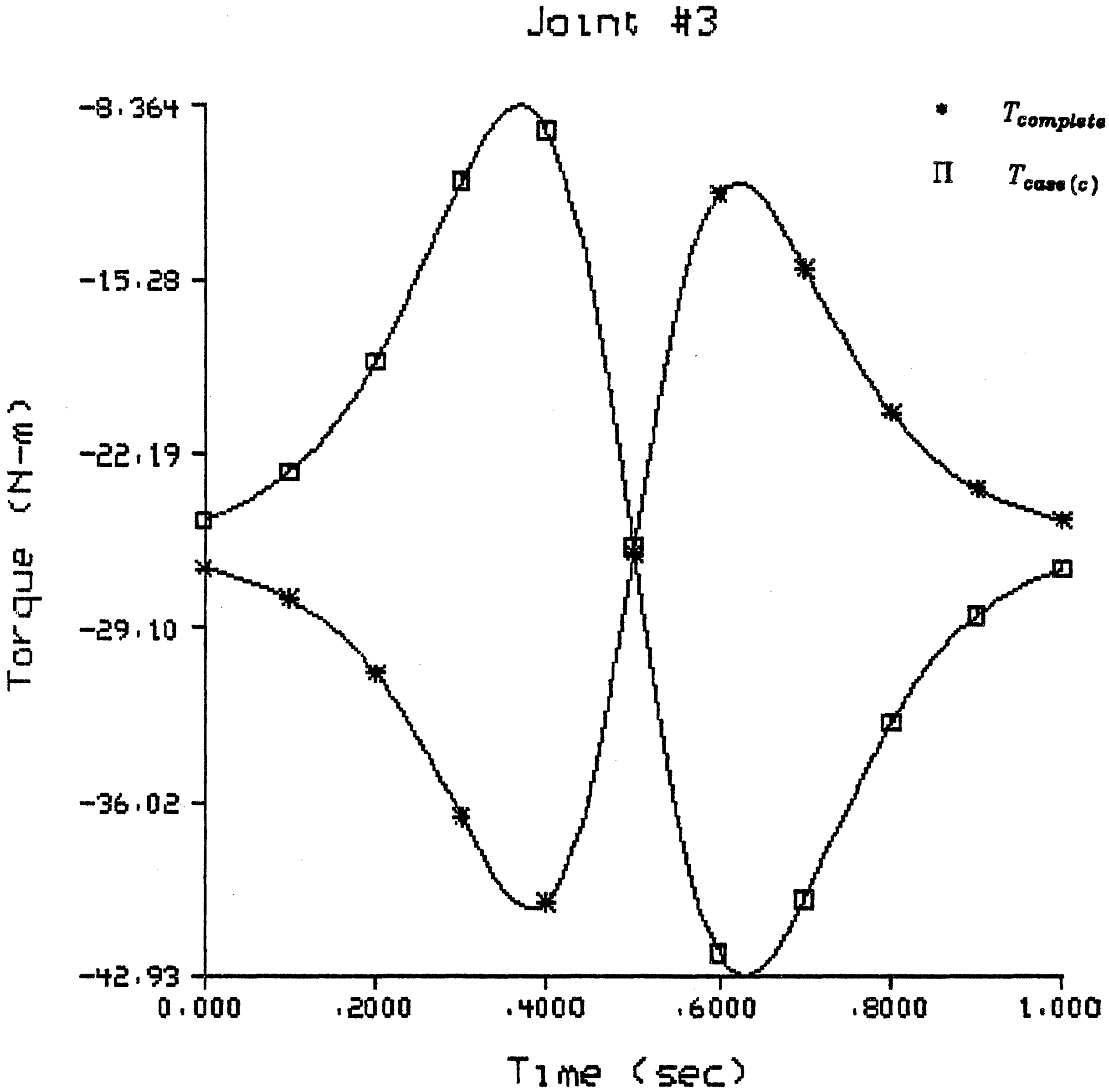

Figure 20 $T_{Complete}$ vs. Case (c)

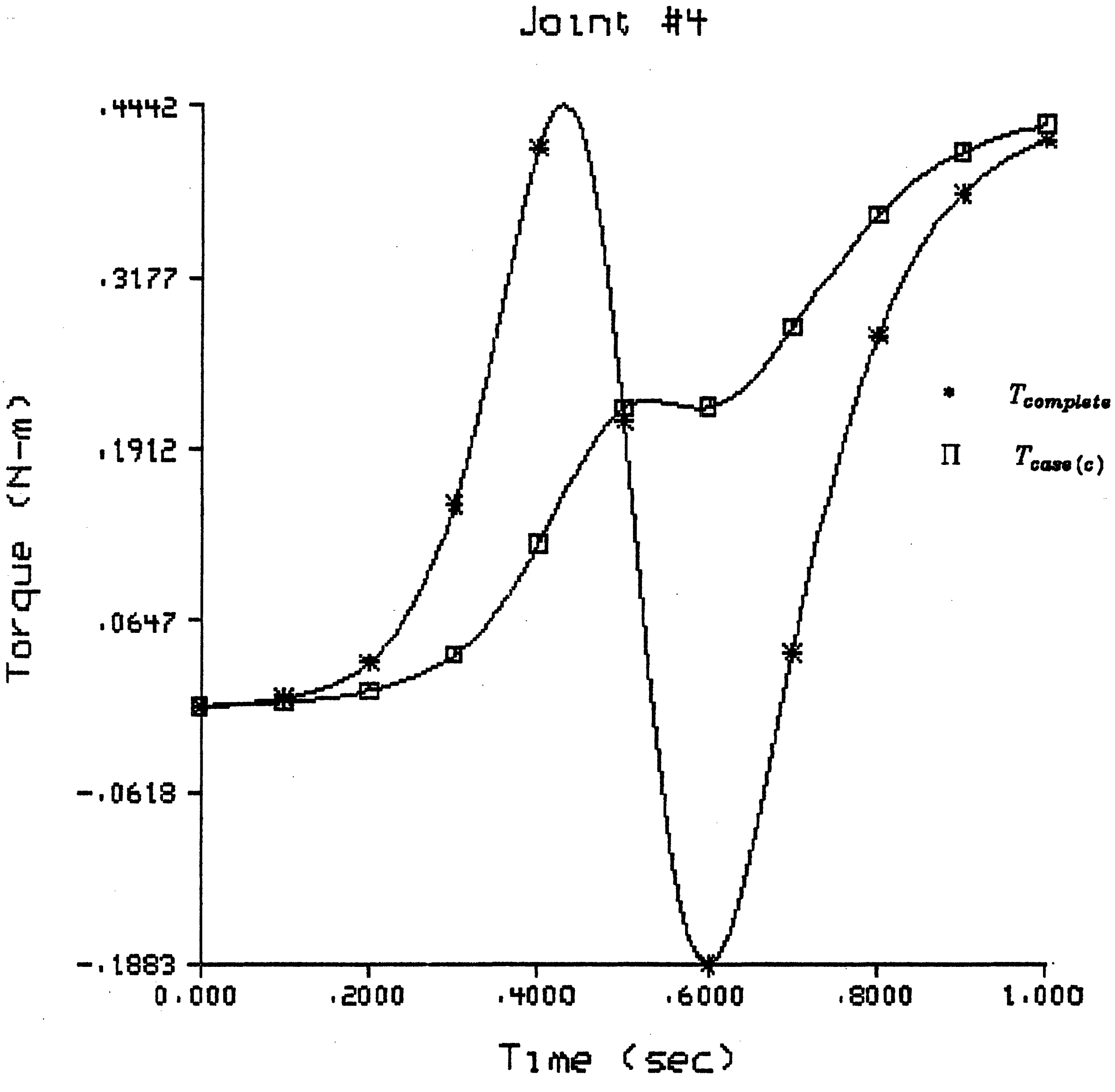

Figure 21 $T_{Complete}$ vs. Case (c)

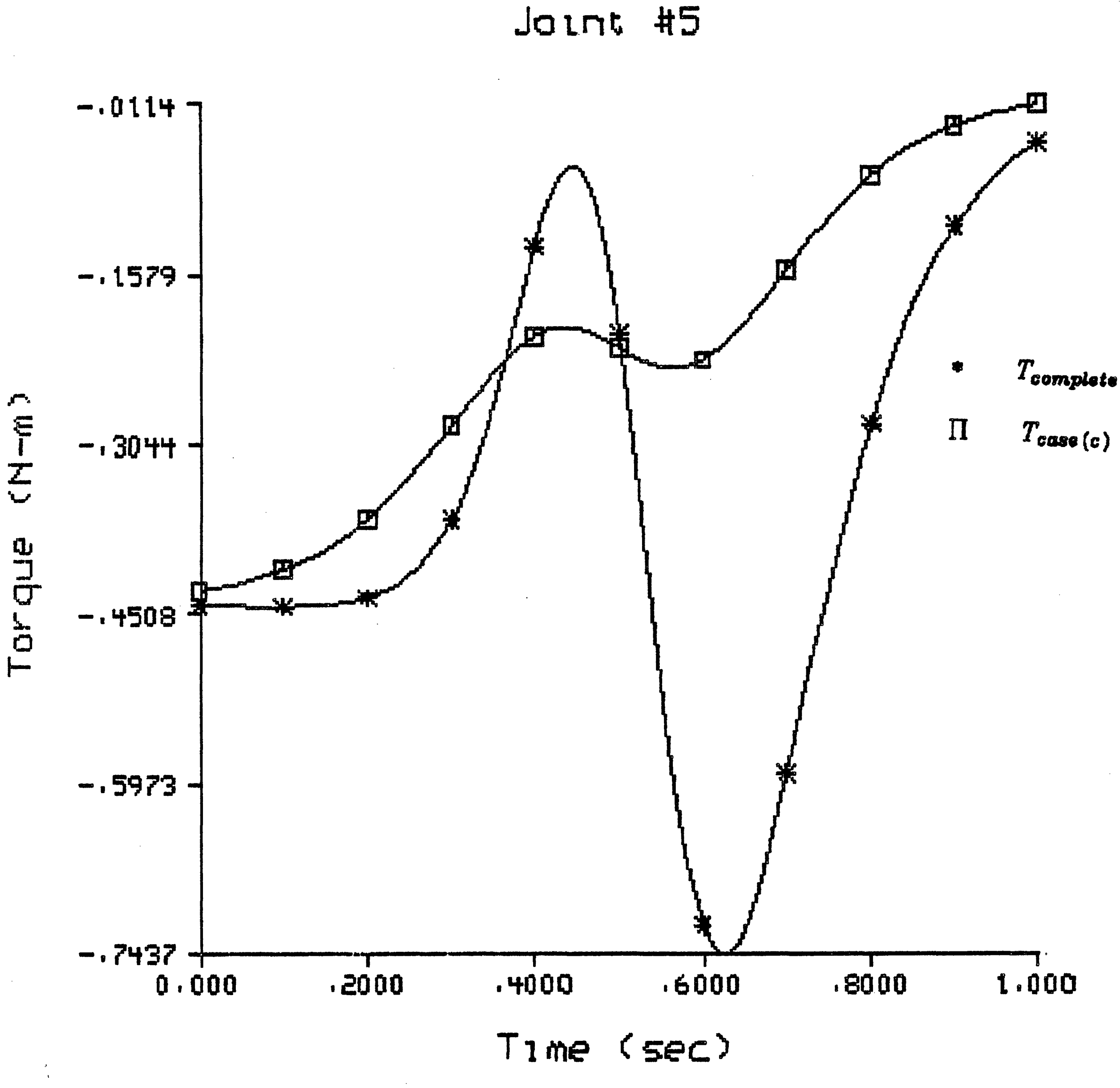

Figure 22 $T_{Complete}$ vs. Case (c)

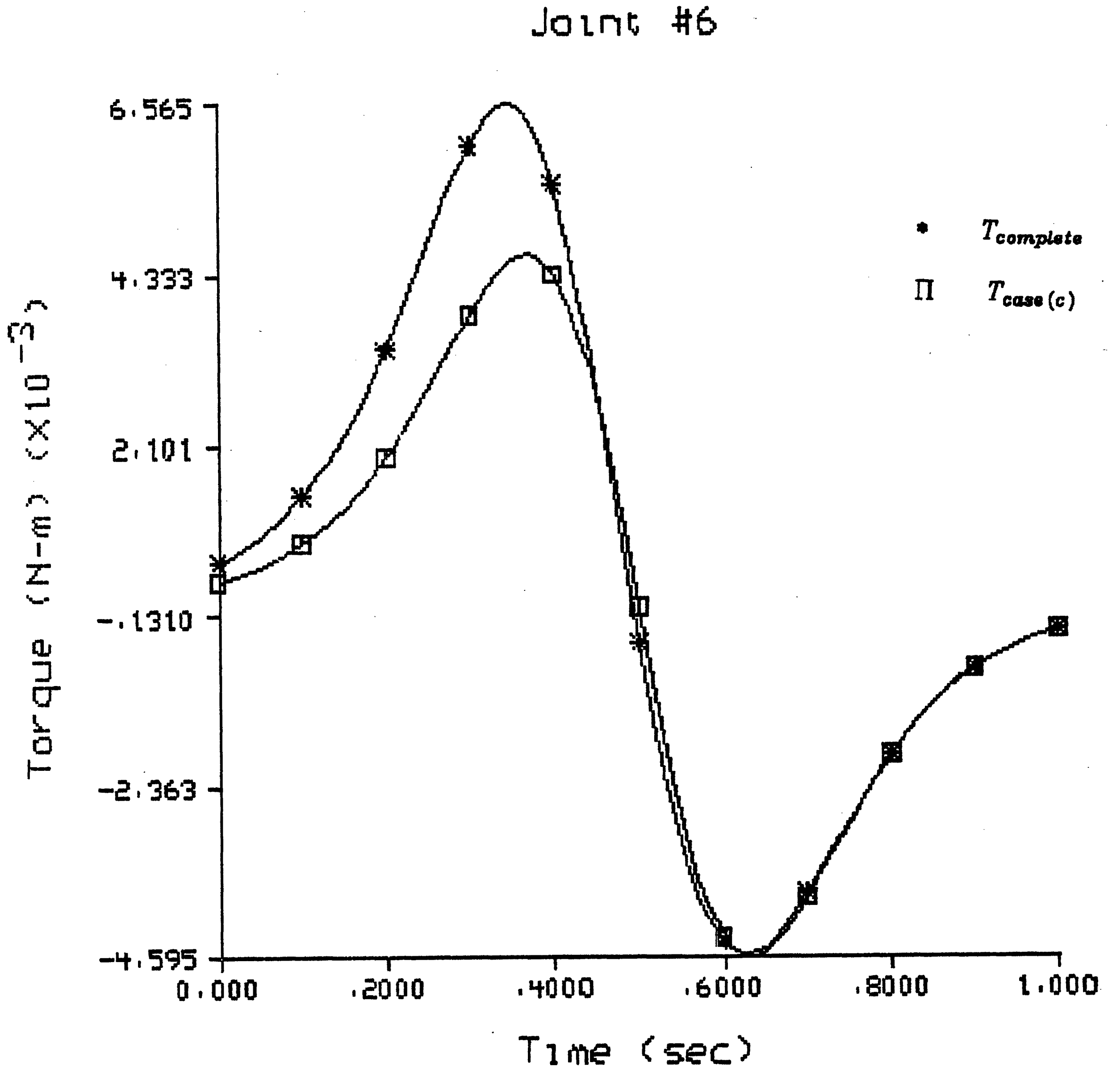

Figure 23 $T_{Complete}$ vs. Case (c)

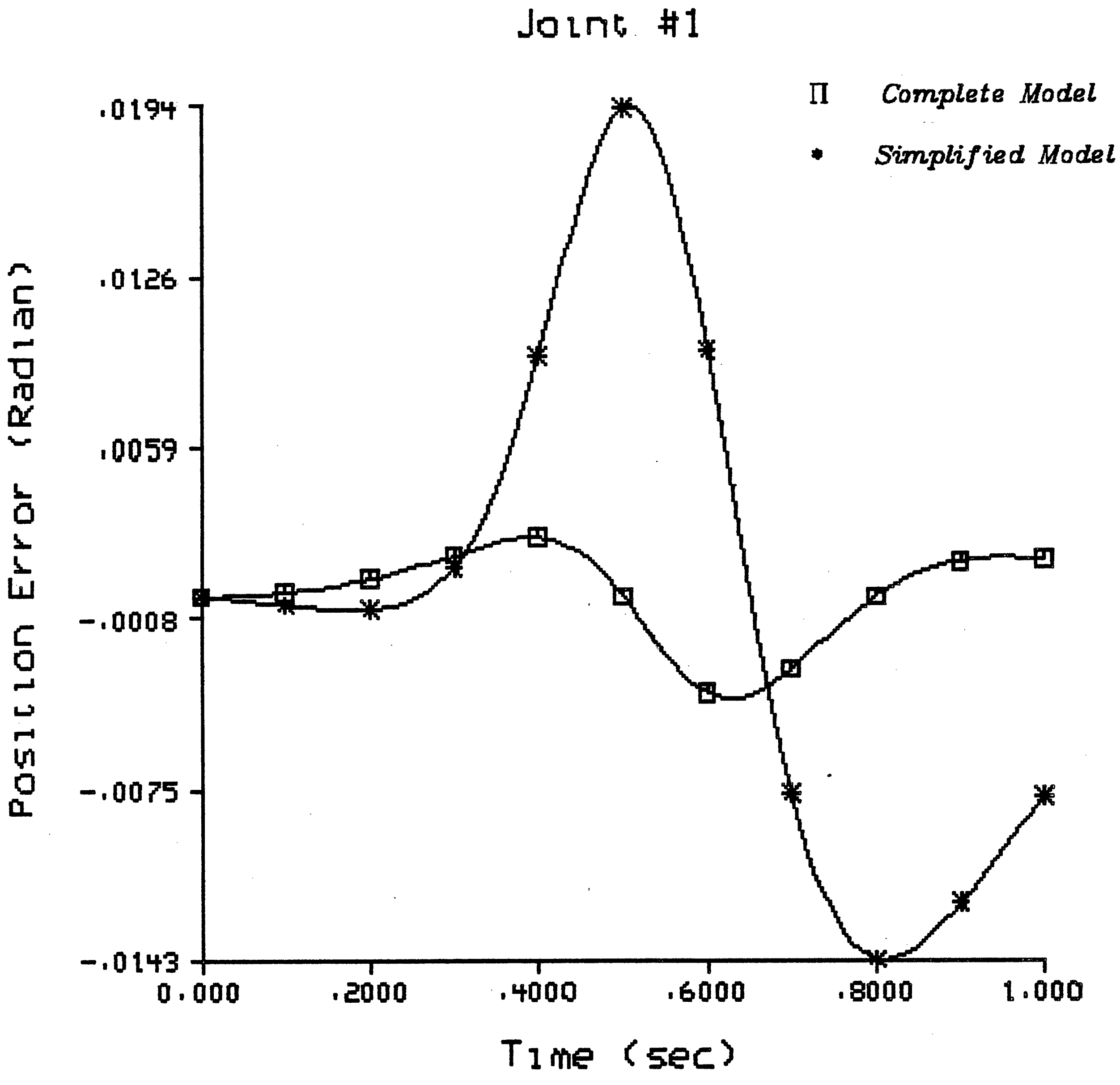

Figure 24 No Load

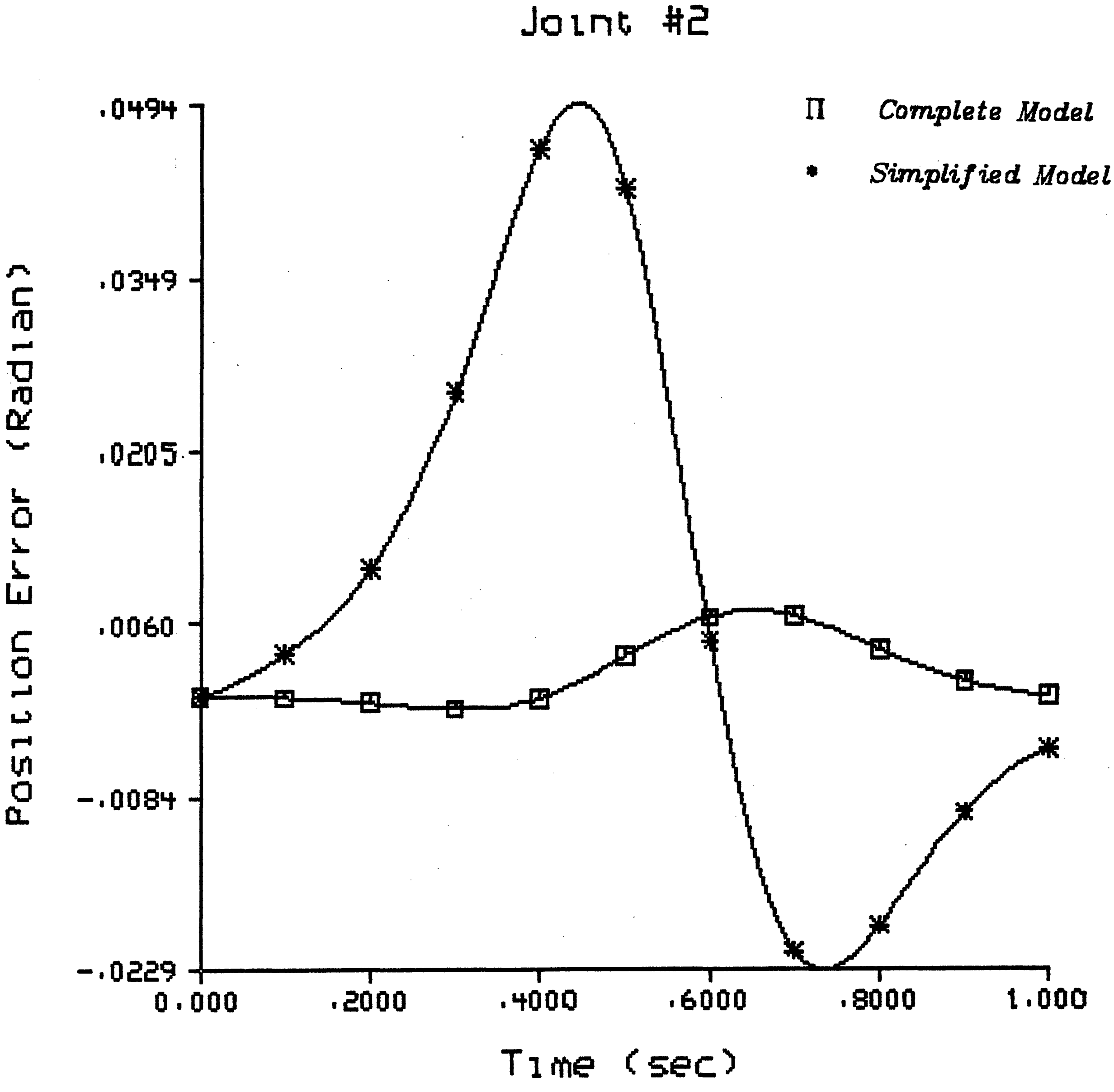

Figure 25 No Load

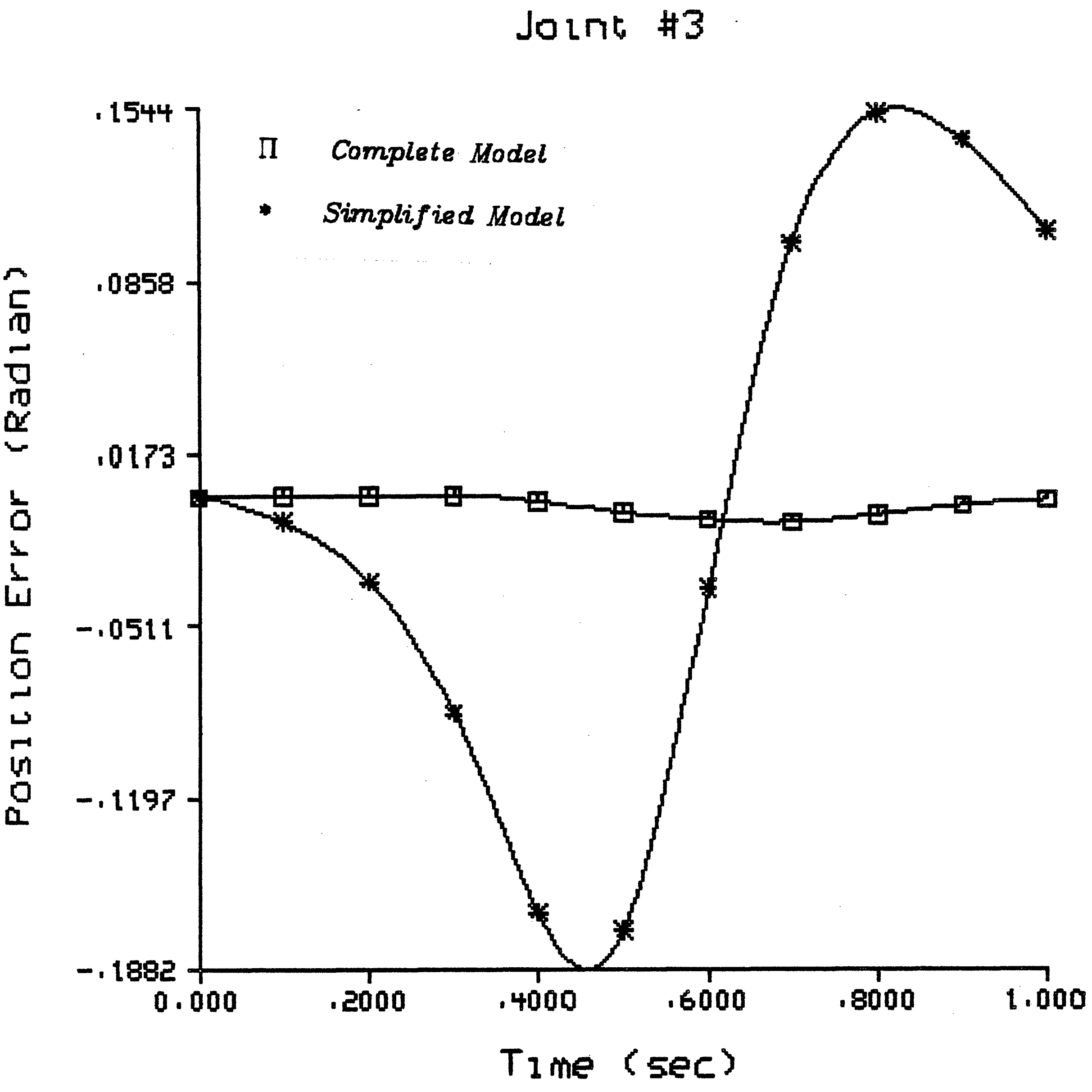

Figure 26 No Load

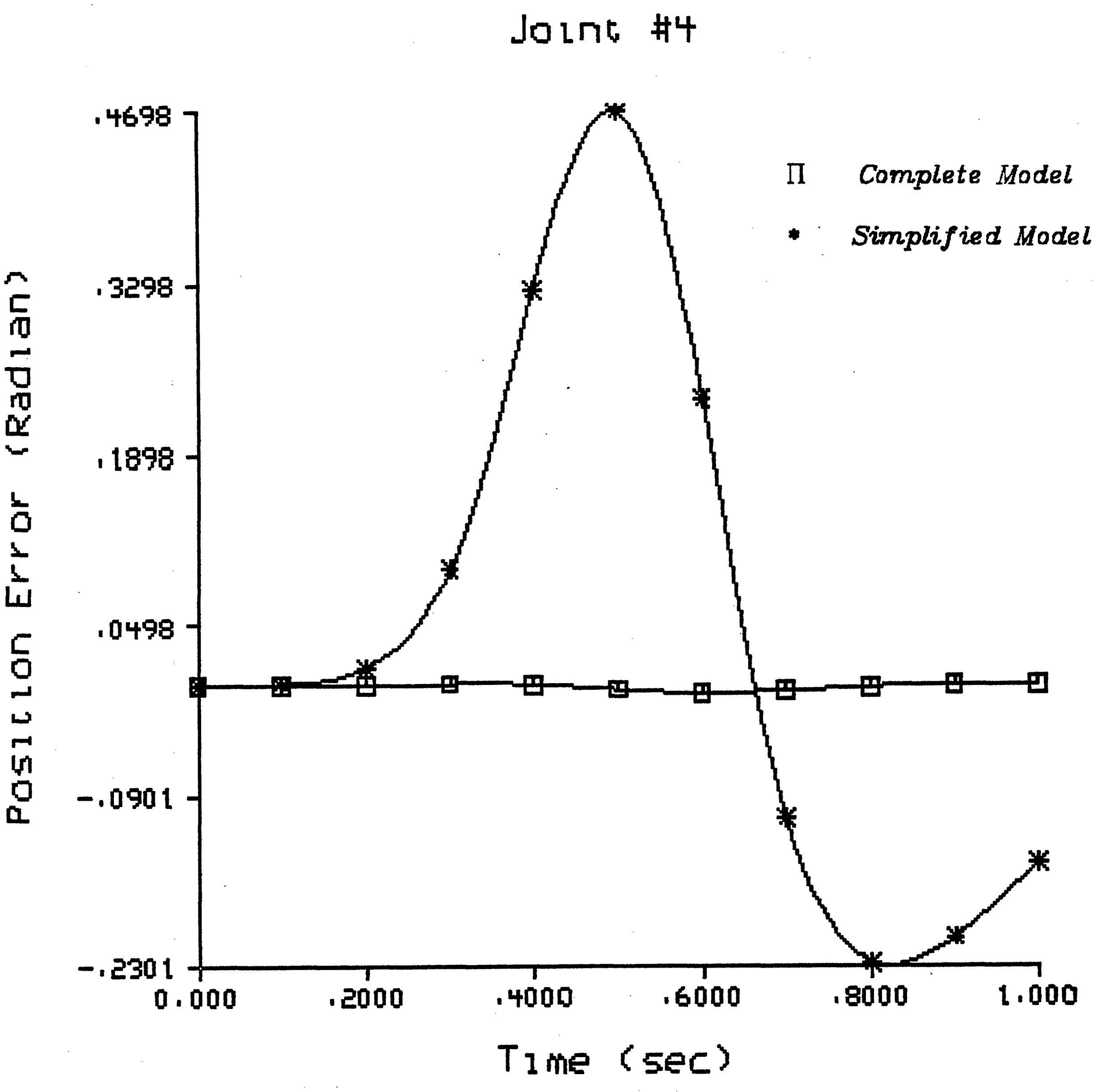

Figure 27 No Load

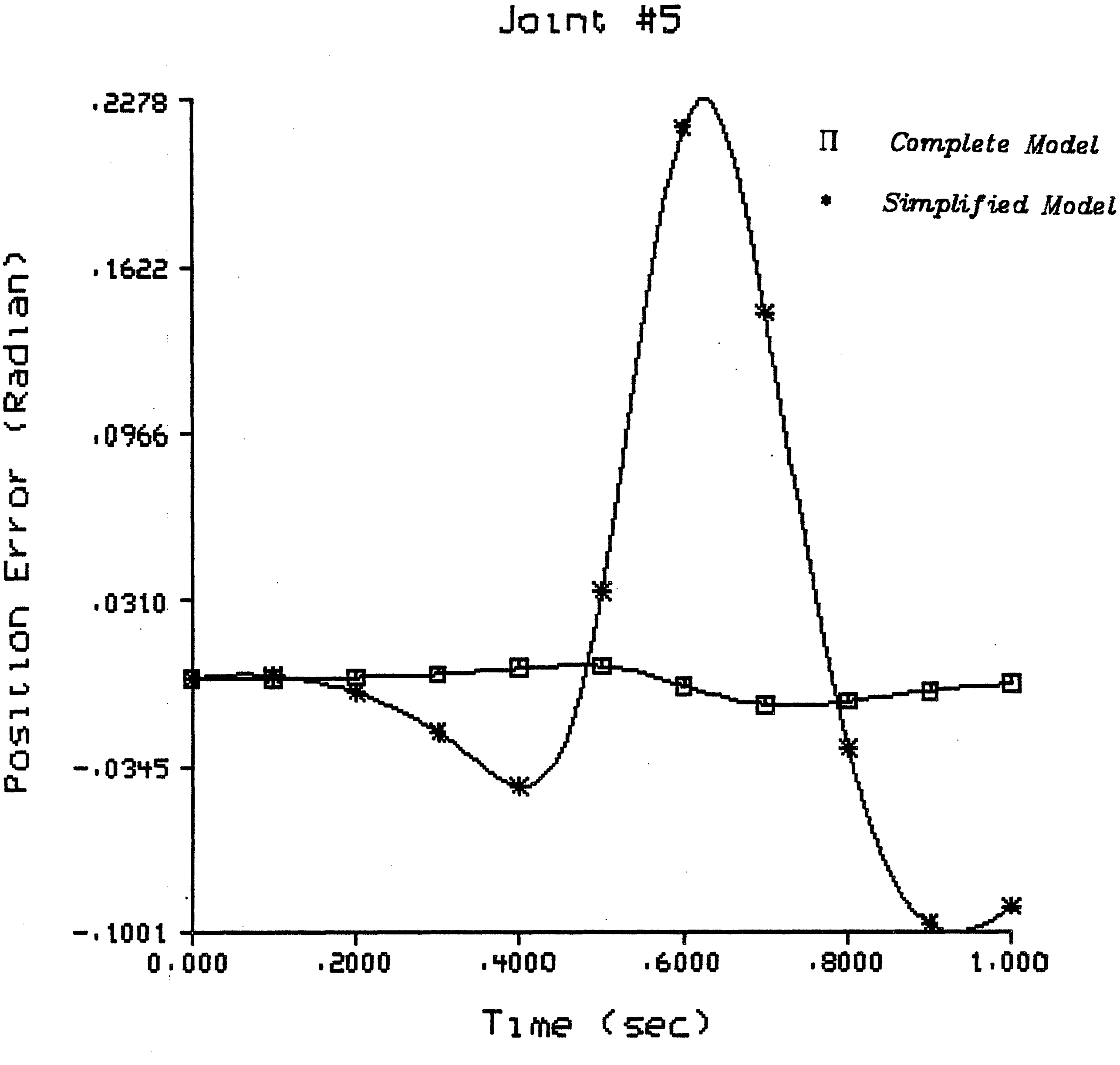

Figure 28 No Load

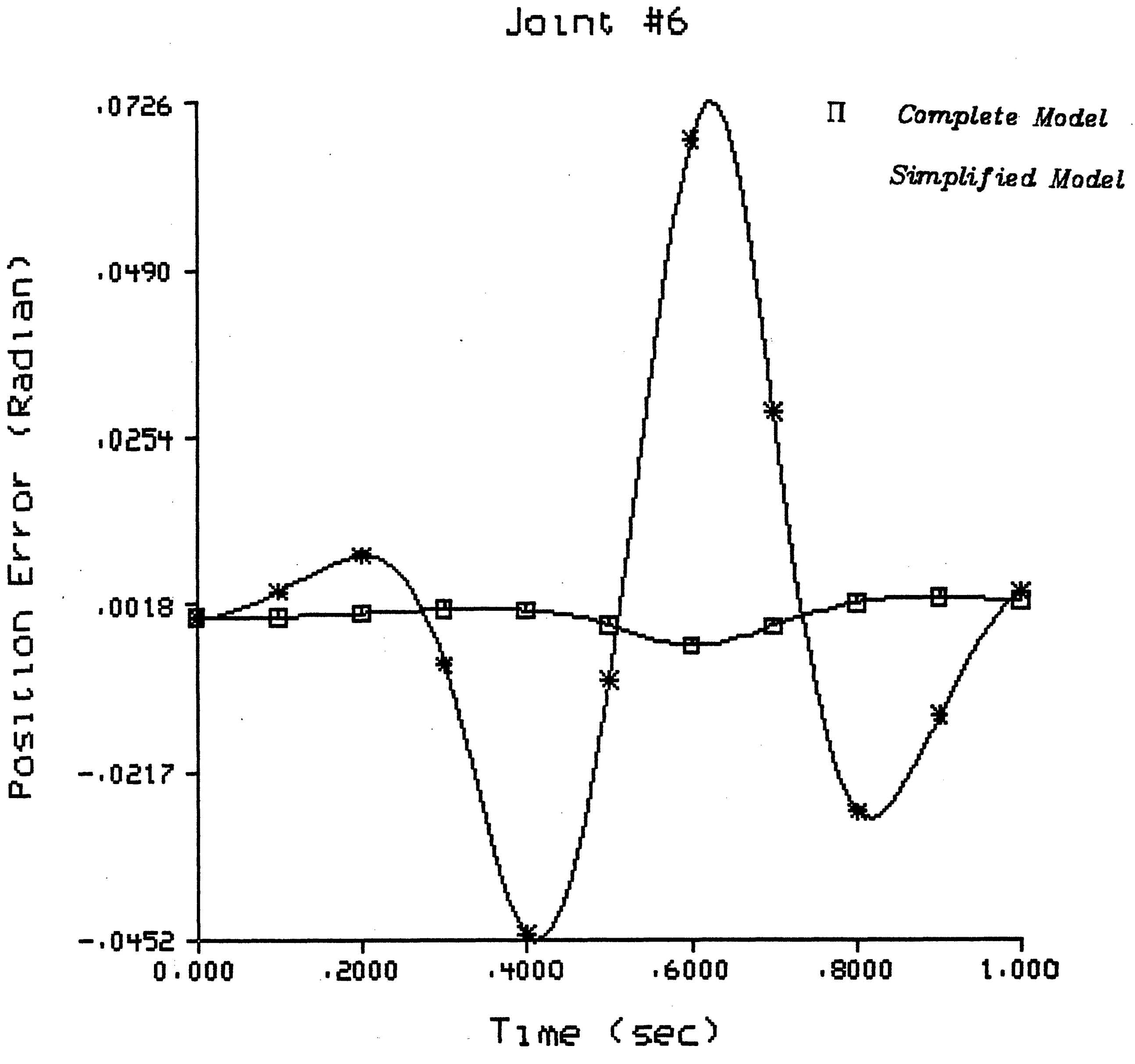

Figure 29 No Load

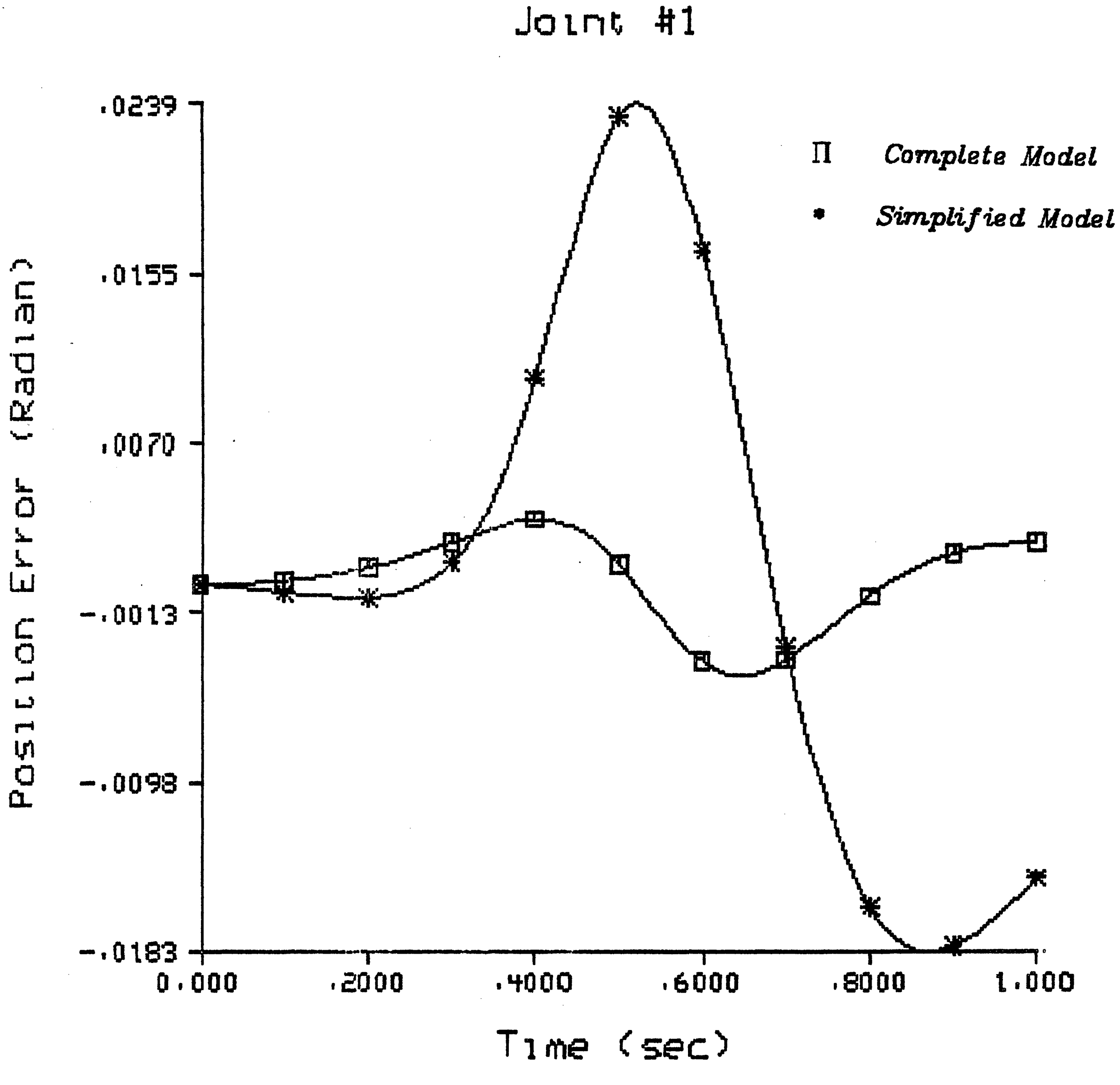

Figure 30 Half Load

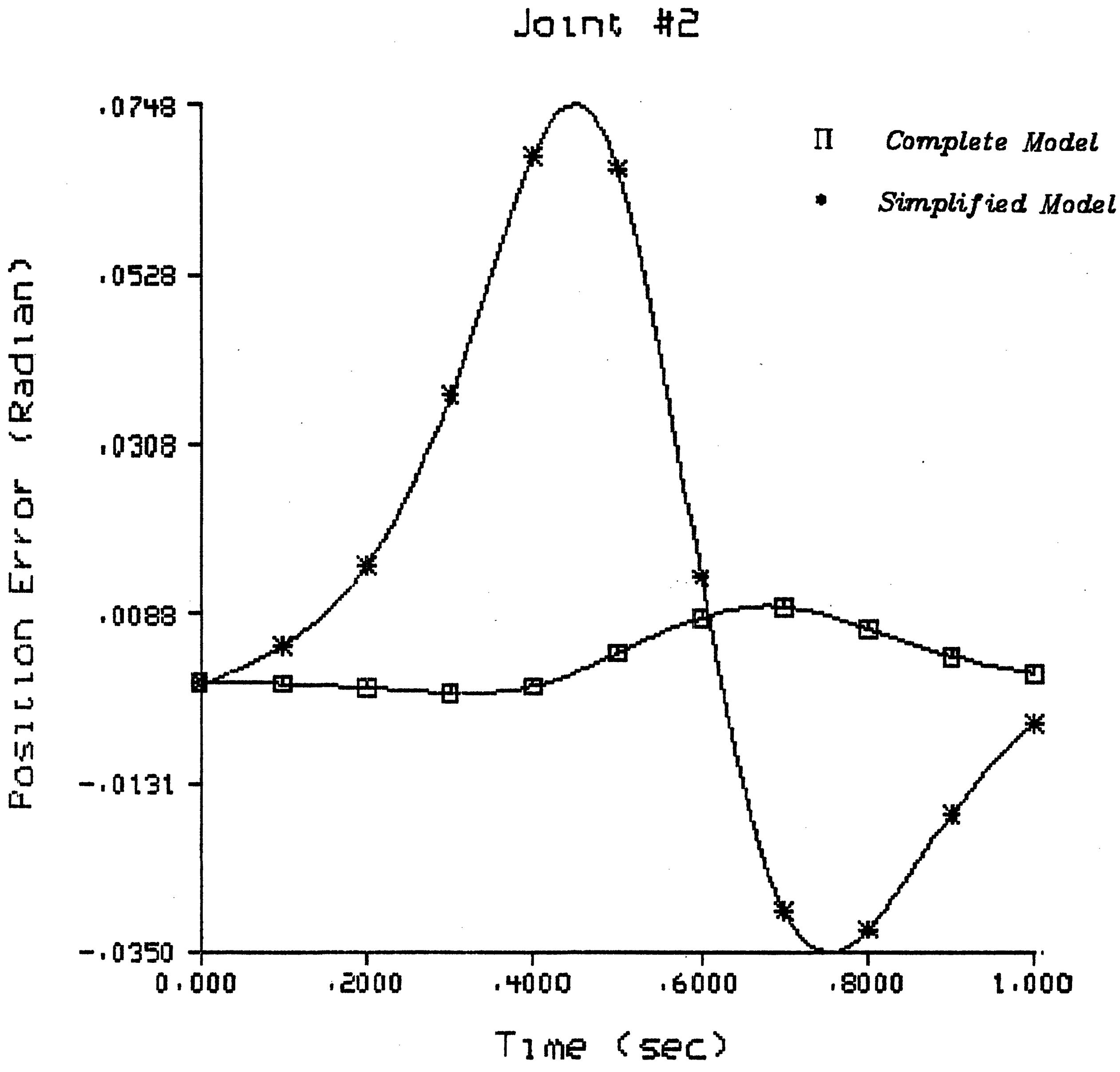

Figure 31 Half Load

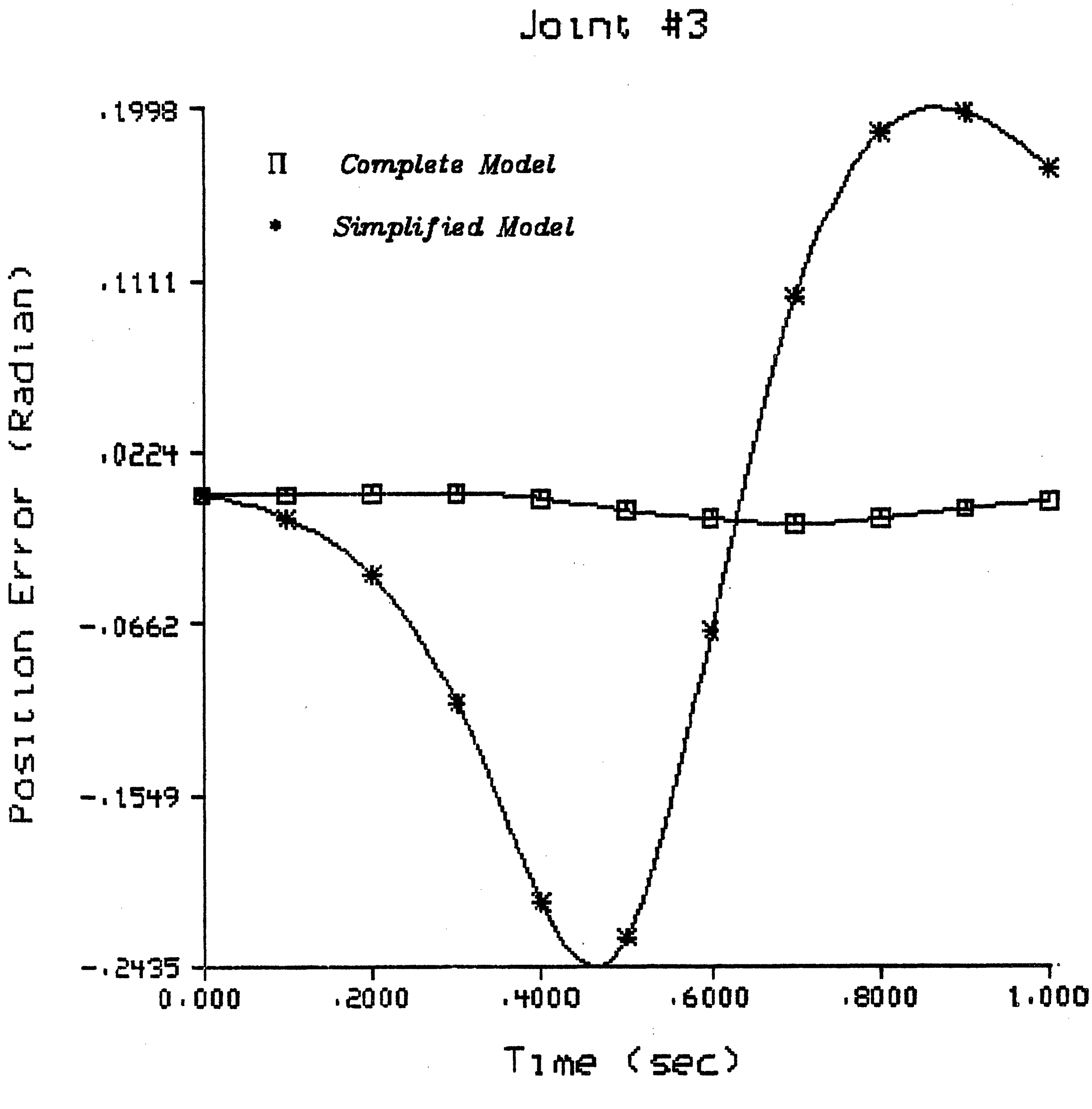

Figure 32 Half Load

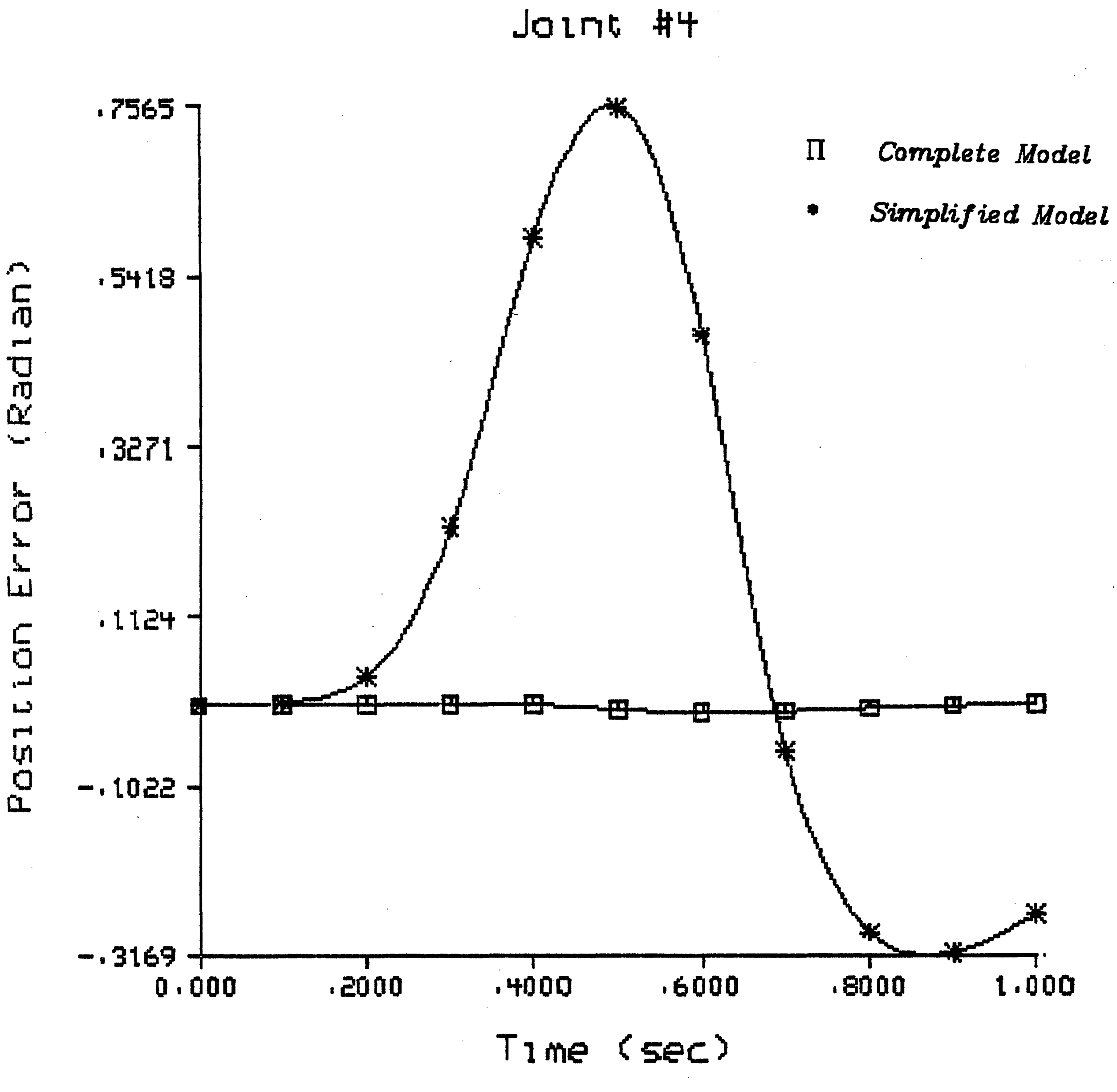

Figure 33 Half Load

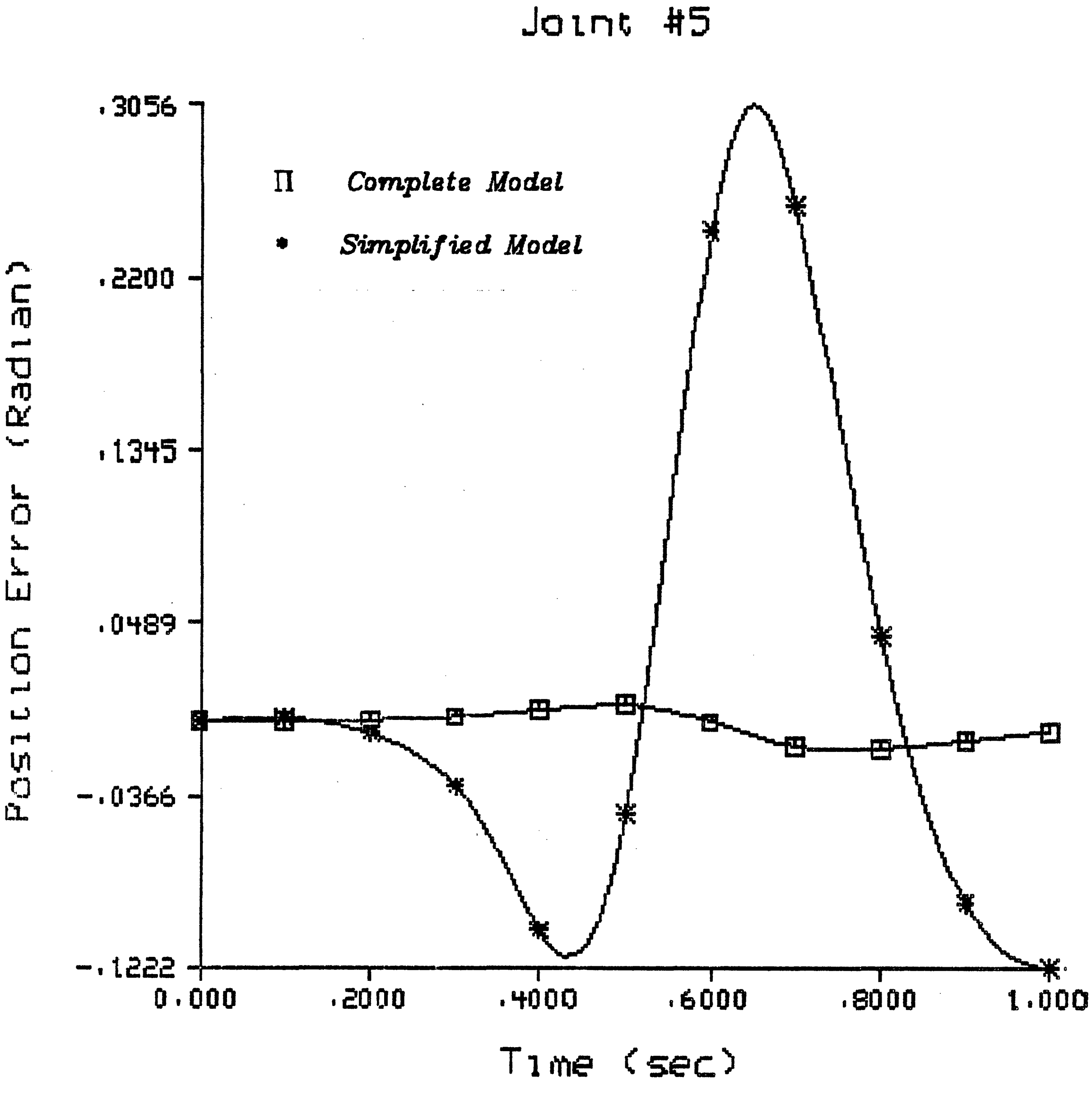

Figure 34 Half Load

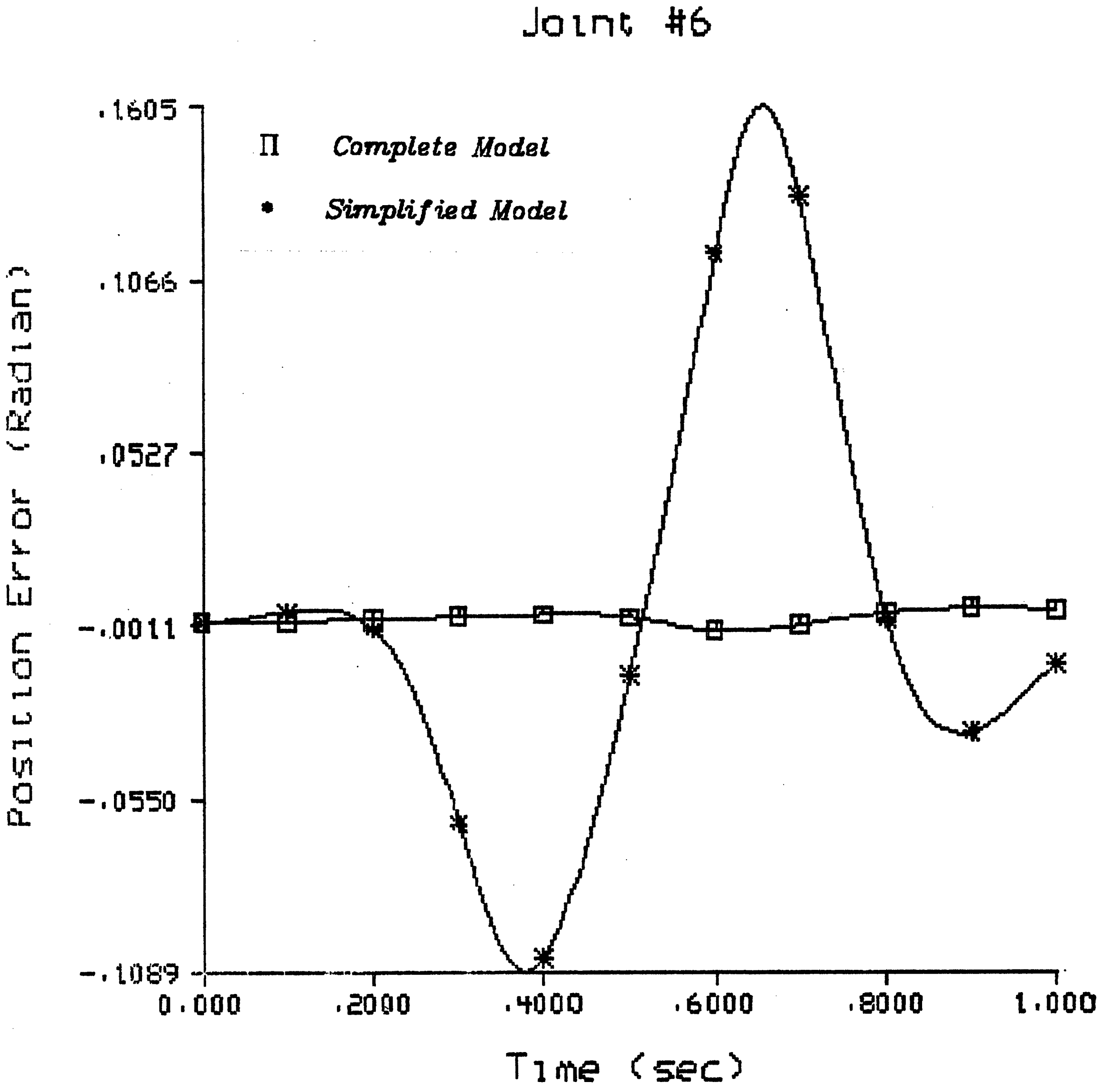

Figure 35 Half Load

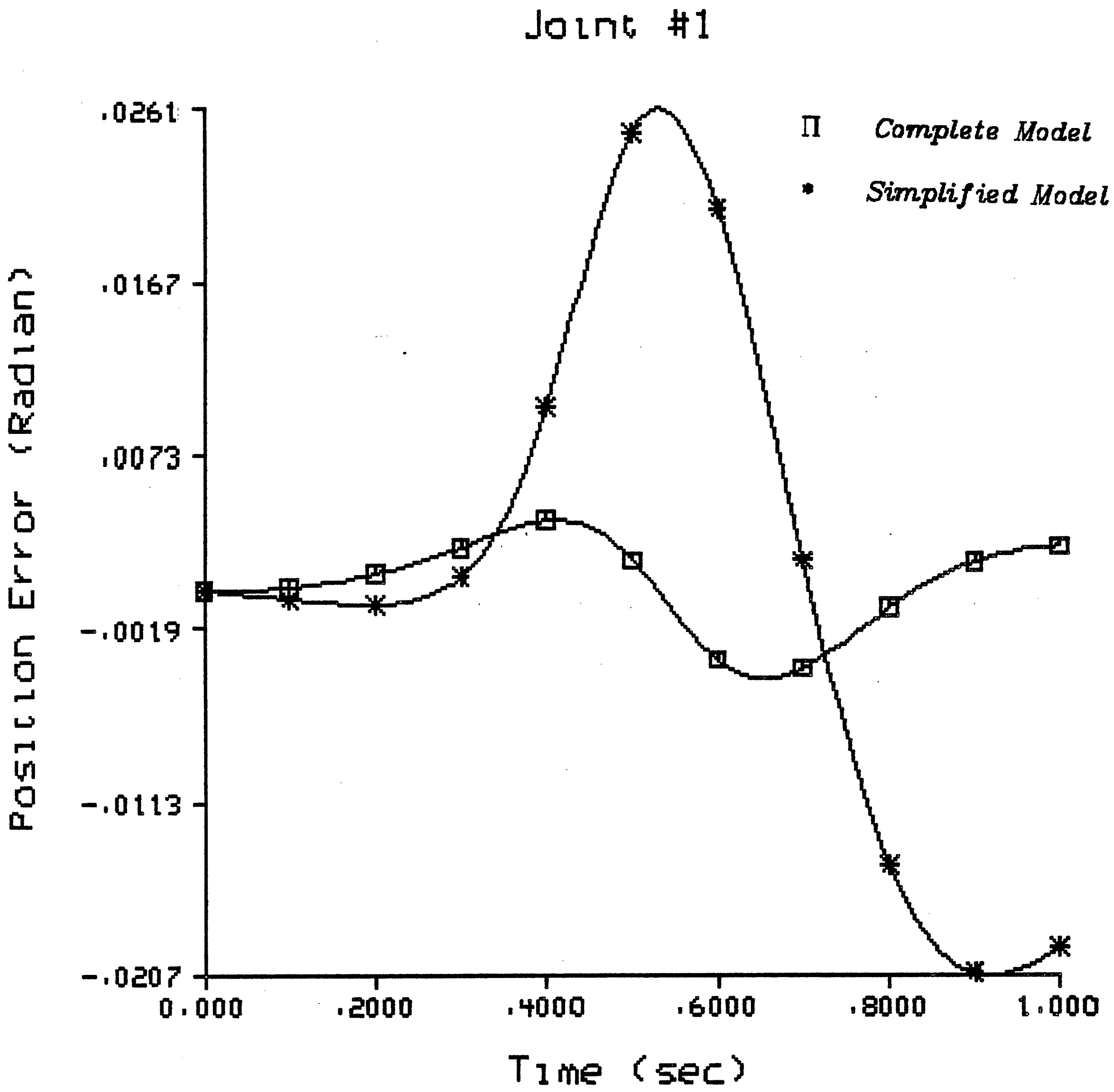

Figure 36 Maximum Load

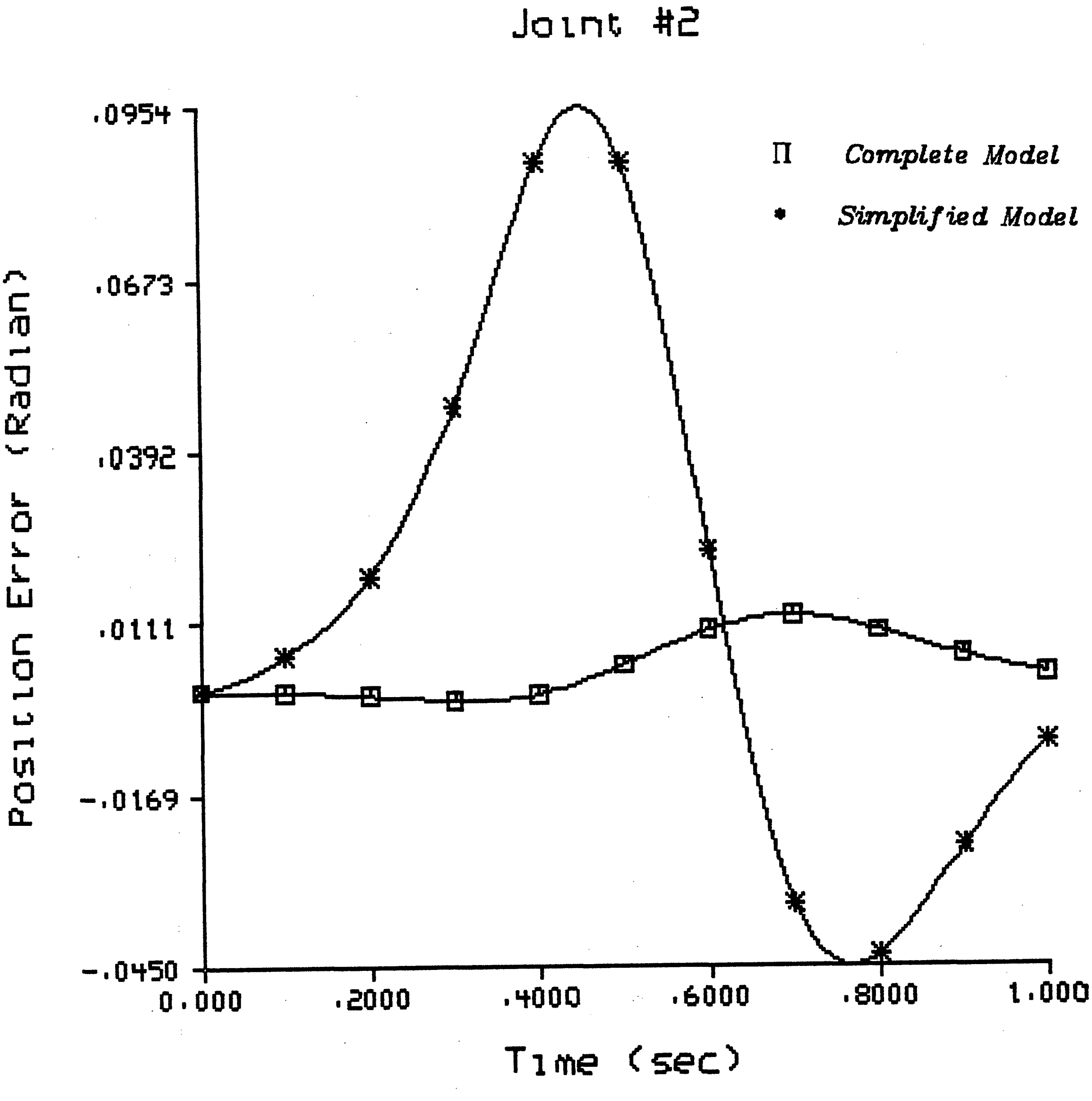

Figure 37 Maximum Load

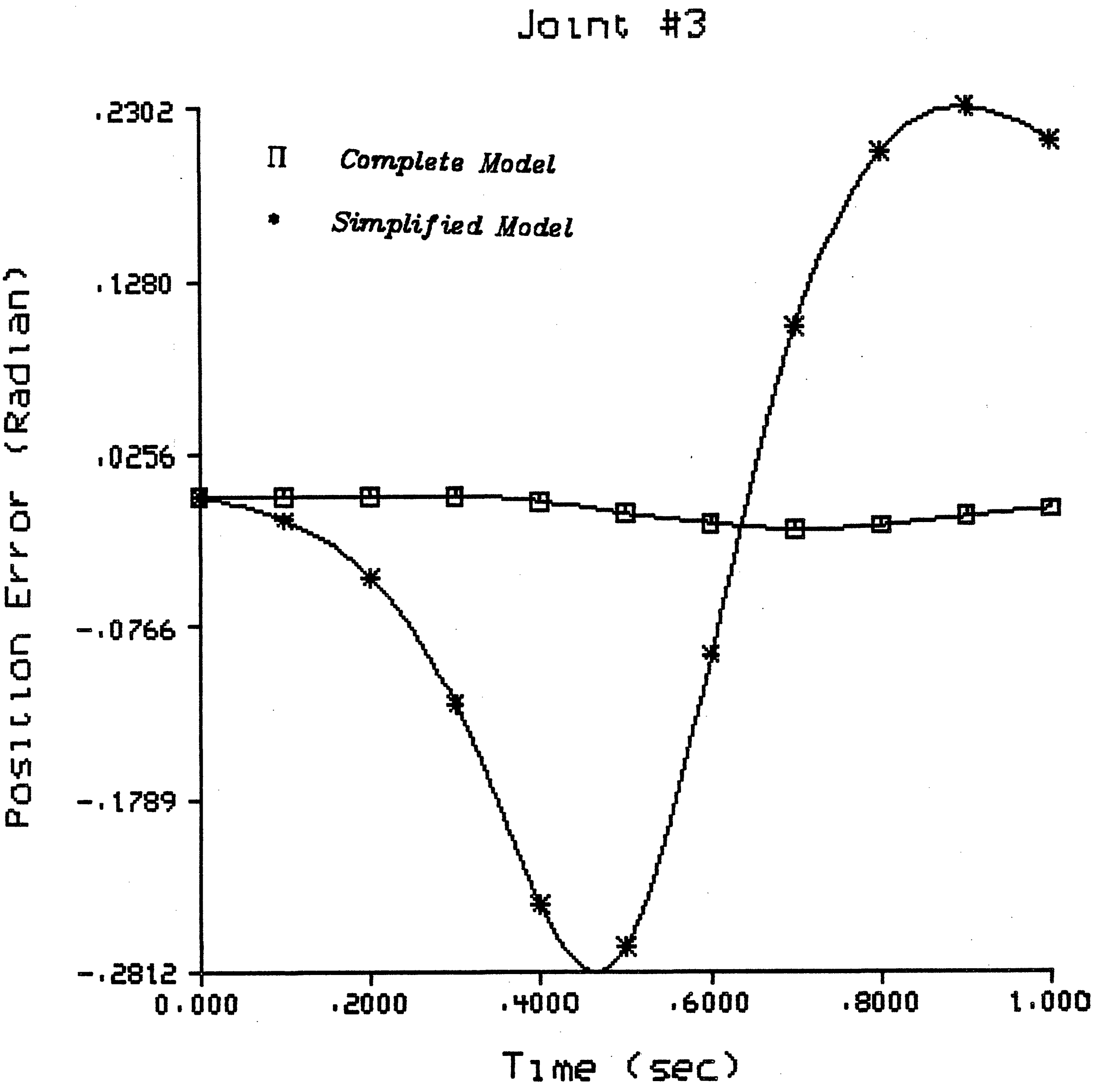

Figure 38 Maximum Load

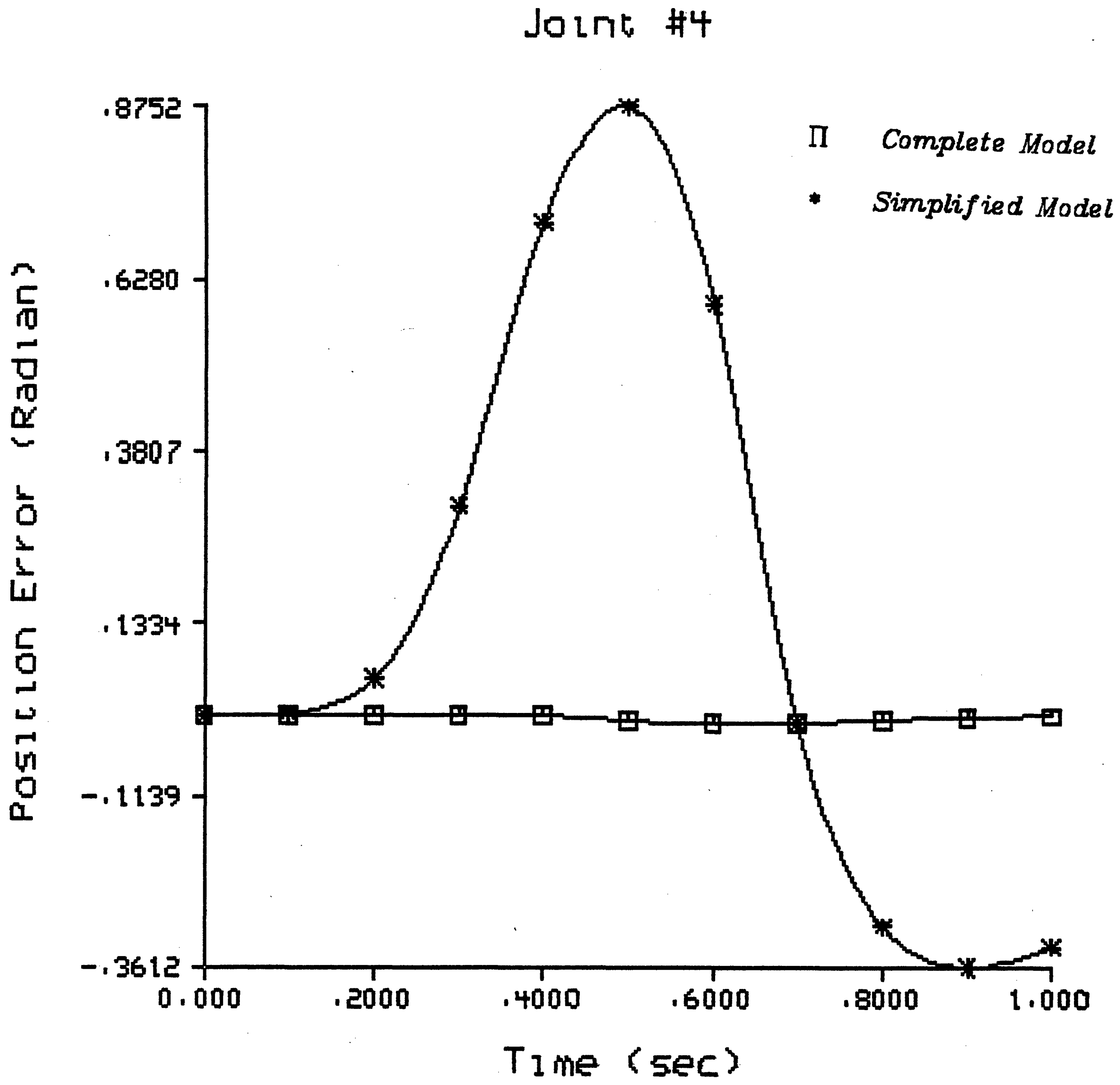

Figure 39 Maximum Load

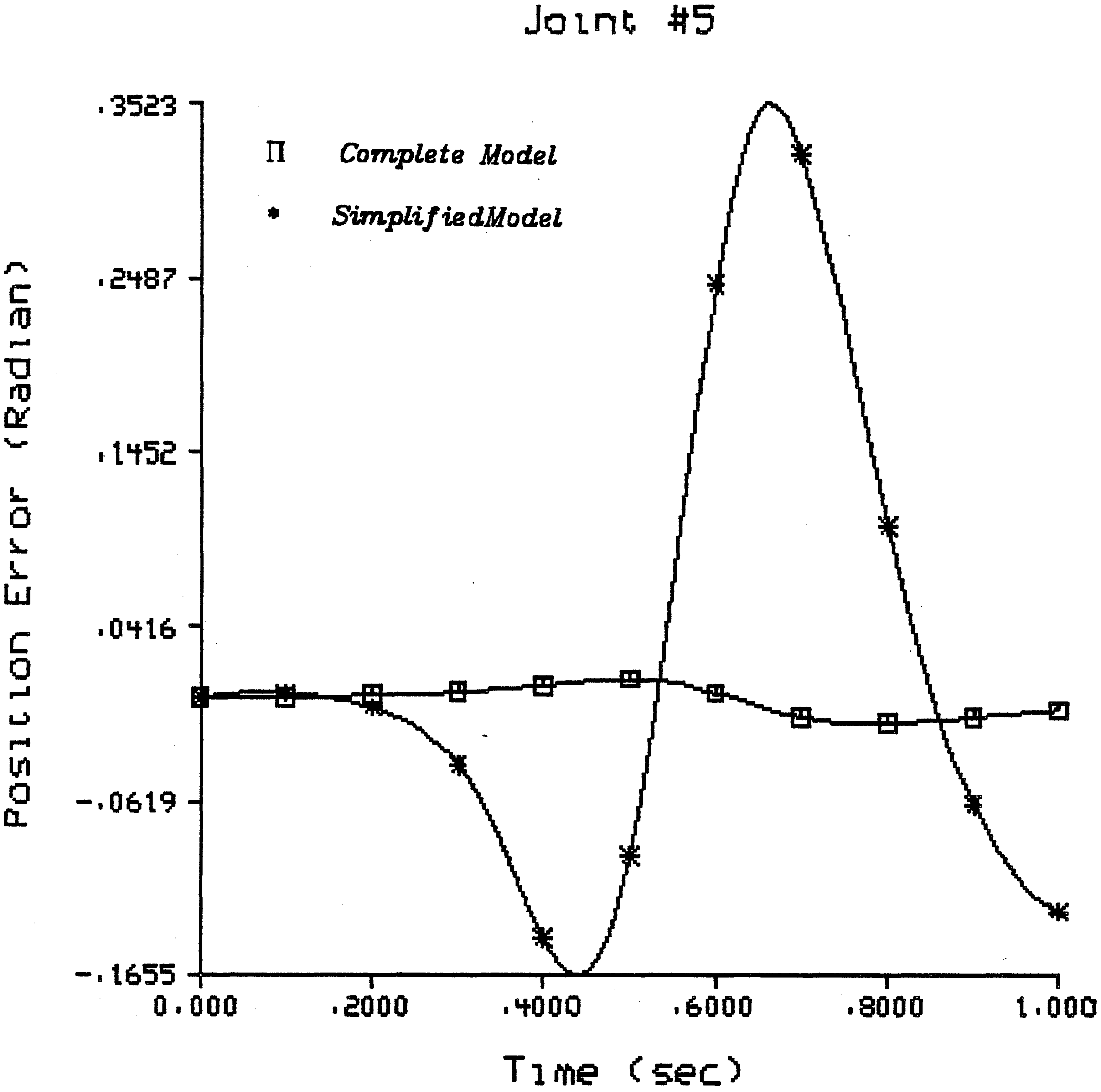

Figure 40 Maximum Load

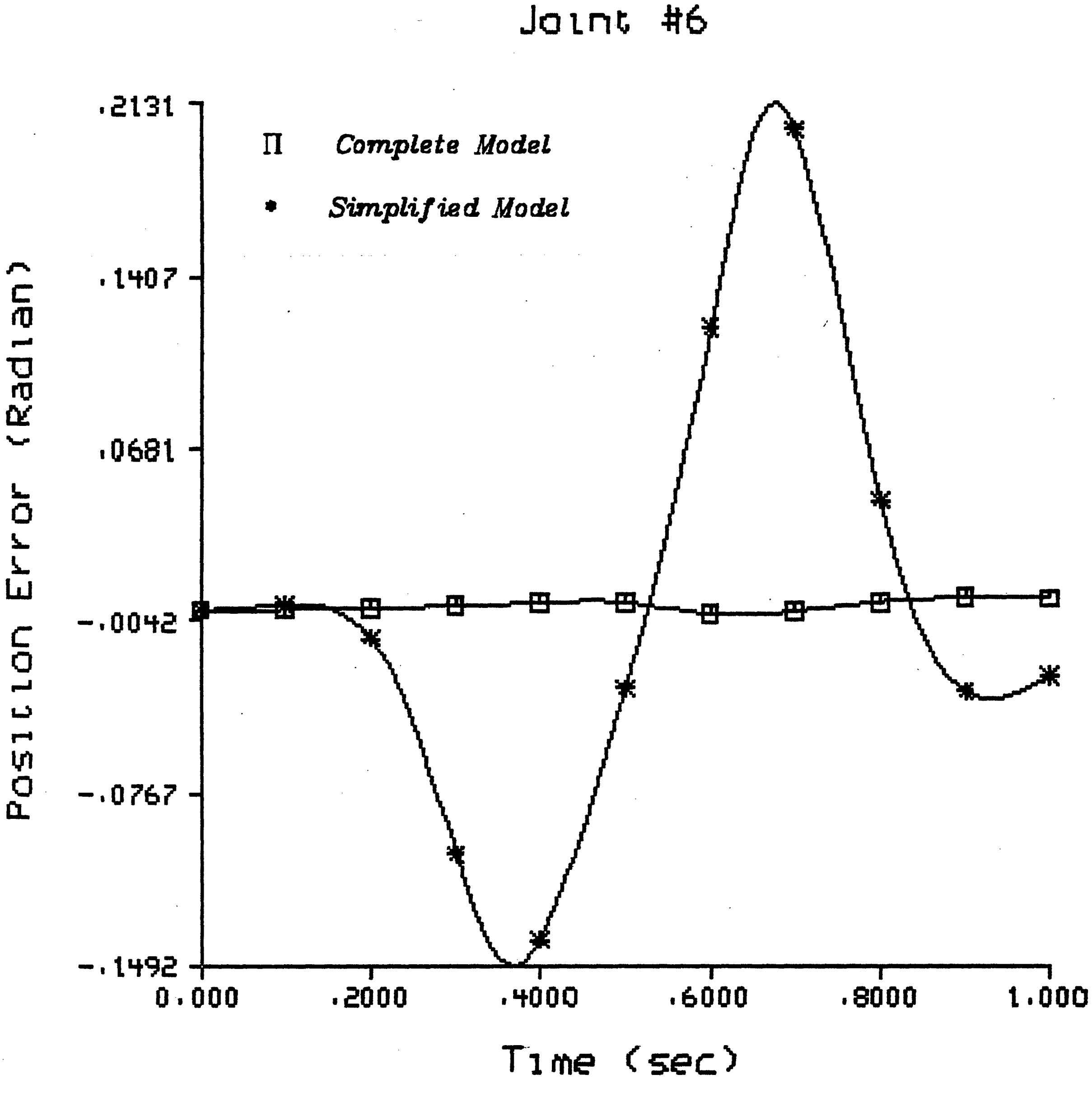

Figure 41 Maximum Load

6. References

[Alb75] Albus, J. S., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control*, Series G, Vol. 97, No. 3, September 1975, pp. 220-227.

[Bej74] Bejczy, A. K., "Robot Arm Dynamics and Control," Technical Memo 33-669. Jet Propulsion Laboratory, February 1974.

[DeH55] Denavit, J., Hartenberg R. S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, June 1955, pp.215-221.

[Hol80] Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 11, November 1980 , pp. 730-736,

[HoT80] Horowitz, R., Tomizuka M., "An Adaptive Control Scheme for Mechanical Manipulators- Compensation of Nonlinearity and Decoupling Control," Dynamic System and Control Division of the A.S.M.E., Winter Annual Meeting, Chicago, Ill., November 1980.

[KaB71] Kahn, M. E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, September 1971, pp 164-172.

[LeC82] Lee, C. S. G.,Chung, M. J. "An Adaptive Control Strategy for Computer-Based Manipulators," Proceedings of the 21st Conference on Decision and Control Orlando, Fla., December 8-10, 1982.

[Lew74] Lewis, R. A., "Autonomous Manipulation on a Robot: Summary of

Manipulator Software Functions," Technical Memo 33-679, Jet Propulsion Lab, March, 1974.

[LLN82] Lee, C. S. G., Lee, B. H., Nigam, R., "An Efficient Formulation of Robot Arm Dynamics for Control and Computer Simulation," CRIM Technical Report RSD-TR-8-82, University of Michigan, August, 1982.

[LWP80] Luh, J. Y. S., Walker M. W., and Paul R. P. C., "On-Line Computational Scheme for Mechanical Manipulators," *Transaction of ASME, Journal of Dynamic Systems, Measurement, and Control*, Vol. 120, June 1980 pp. 69-76.

[Mar73] Markiewicz, B.R., "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator," Technical Memorandum 33-601, Jet Propulsion Laboratory, March 15, 1973.

[OMV79] Orin, D. E., McGhee, R. B., and Vukobratovic, M., and Hartoch, G., "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods," *Mathematical Biosciences*, Vol. 43, No.1/2, February 1979, pp 107-130.

[Pau72] Paul, R. "Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm," Stanford Artificial Intelligence Laboratory Memo AM-177, November 1972.

[Pie68] Pieper, D. L., "The Kinematics of Manipulators Under Computer Control," Computer Science Department, Stanford University, Artificial Intelligence Project Memo No. 72, October 1968.

[Sym71] Symon, K. R., *Mechanics*, 2nd addition, Addison-Wesley, 1971.

[TML80] Turney, J., Mudge, T. N., Lee, C. S. G., "Equivalence of Two Formulations for Robot Arm Dynamics," SEL Report 142, ECE Department, University

UNIVERSITY OF MICHIGAN
3 9015 03023 0323

of Michigan, December 1980.

[Whi69] Whitney, D. E., "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transaction on Man-Machine Systems*, Vol. MMS-10, no. 2, June 1969 pp 47-53