UAH Technical Report No: 5-32639
Award No: NA58-36955
Delivery Order No: 119

A MULTIPLE POINTING-MOUNT CONTROL STRATEGY FOR SPACE PLATFORMS

Final Report

11 April, 1991 through 10 April, 1992

Prepared by
Dr. C. D. Johnson
Electrical and Computer Engineering Dept.
Univ. of Alabama in Huntsville
Huntsville, Alabama 35899

Prepared for
Pointing Control Systems Branch
NASA, Geo. C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

June. 1992

Report Documentation Page

NASA National Aeronautics and Space Administration

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
1		

4. Title and Subtitle
A Multiple Pointing-Mount Control Strategy for Space Platforms

5. Report Date
June 1992

6. Performing Organization Code

7. Author(s)
Dr. C. D. Johnson

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address
University of Alabama in Huntsville, ECE Dept.
Huntsville, Alabama 35899

11. Contract or Grant No.
NAS8-36955, D.O. # 119

13. Type of Report and Period Covered
FINAL
4/11/91 to 4/10/92

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546-0001
Marshall Space Flight Center, Huntsville, AL

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A new distrubance-adaptive control strategy for multiple pointing-mount space platforms is proposed and illustrated by consideration of a simplified 3-link dynamic model of a multiple pointing-mount space platform. Simulation results demonstrate the effectiveness of the new platform control strategy. The simulation results also reveal a system "destabilization phenomena" that can occur if the set of individual platform-mounted experiment controllers are "too responsive."

17. Key Words (Suggested by Author(s))
Platform control system
Multiple Pointing-Mount
GEO

18. Distribution Statement

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price

Table of Contents

A Multiple Pointing-Mount Control Strategy for Space Platforms

Chapter 1

An Overview of the Multiple Pointing-Mount Space Platform Control Problem

Chapter 2

A 3-Link Multi-Body Generic Model of a MPMSP in Planar Motion

Chapter 3

Design of a Disturbance-Adaptive MPMSP Control Strategy for the Planar Motion Generic Model

Chapter 4

An Exact Mathematical Model of the Generic MPMSP in Planar-Motion

Chapter 5

Performance Evaluation of the Disturbance-Adaptive Platform Controller Using an "Exact" Simulation of the 3-Link Generic MPMSP Model in Planar Motion

Chapter 6

Summary and Recommendations for Further Work

Chapter 1

AN OVERVIEW OF THE MULTIPLE POINTING-MOUNT SPACE PLATFORM CONTROL PROBLEM.

1.1 Space Experiments Involving Instrument Pointing and Slewing Motions

A number of space-based scientific experiments planned by NASA for the near future will involve instruments that are required to continually point accurately at various objects in space and/or locations on the earth's surface. Still other experiments will be required to perform carefully controlled back-and-forth slewing motions and thereby scan various regions of space and/or the earth's surface, looking for characteristic features of certain phenomena being studied.

In order to accommodate many such experiments, in a cost-effective manner, NASA has conceived the idea of a multiple pointing-mount space platform (hereafter referred to as MPMSP) on which a variety of such experiments would be mounted and operated simultaneously; Fig. 1.1. This MPMSP would act as a common chassis, or framework, to which the various moving and non-moving equipment modules associated with each experiment would be attached. In addition, solar power collector panels, and telemetry antennae that transmit scientific data and receive uplink commands associated with each experiment, will be attached to the MPMSP. The proposed Geostationary Earth Observatory (GEO) project, Figure 1.2, is one specific example of an MPMSP application.

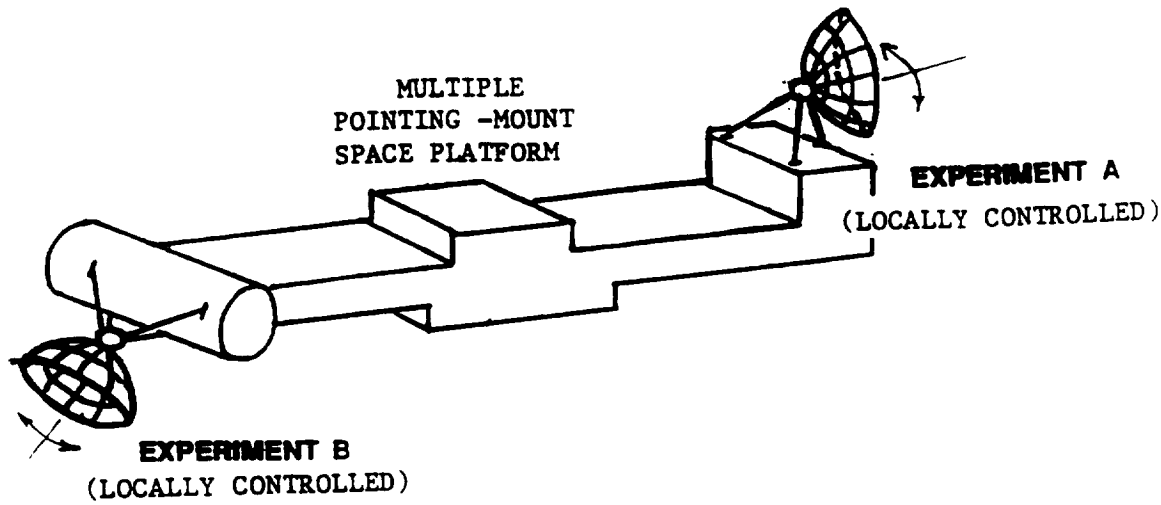


Figure 1.1 Concept of a Multiple Pointing-Mount Space Platform (MPMSP).

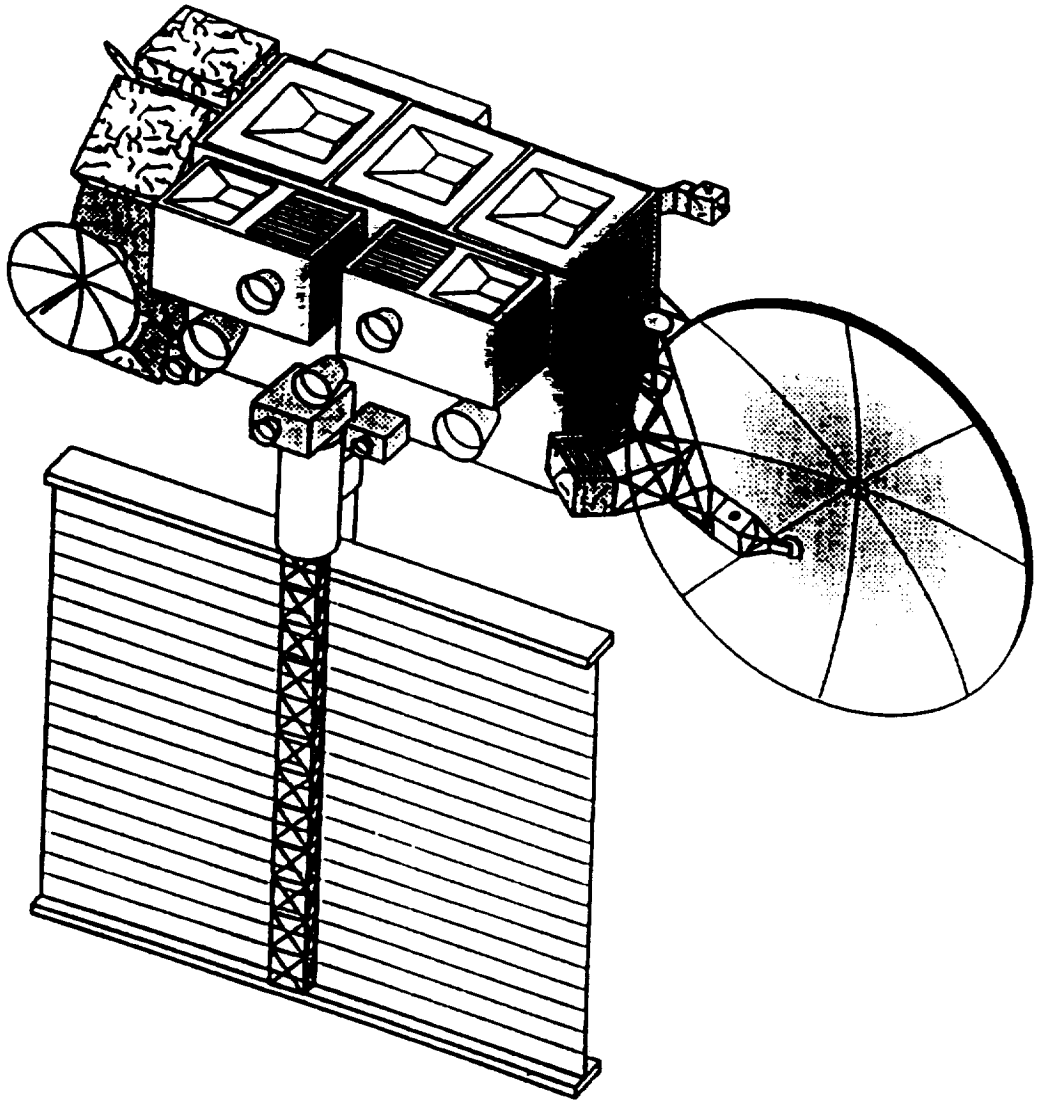


Figure 1.2 Specific Example of an MPMSP; the Proposed Geostationary Earth Observatory (GEO).

1.2 Conflict and Antagonism in Distributed Multi-Controller Systems

The individual pointing and slewing experiments mounted on an MPMSP will each be controlled by its own individual (local) controller, which will be designed to achieve and maintain the pointing or slewing requirements for that particular experiment. This family of local controllers comprises what is called a (spatially) *distributed multi-controller system*. The design of these local experiment controllers will probably be carried out by a variety of different vendors (design teams) working independently, and will involve consideration of the uncertain disturbance motions and/or vibrations of the common platform-framework induced by equipment movements associated with other experiments that are operating simultaneously.

A sometimes overlooked subtlety of such a multiple-controller arrangement mounted on a common platform is that *the platform base-motion disturbances felt by any one experiment are not conventional random-like exogenous inputs, but rather, are functionally related to the controller feedback actions of all the other individual experiment controllers mounted on the same platform.* As a consequence of this subtle fact, the otherwise well-designed experiment controllers can become antagonistic to one another in the sense that the control actions of any one controller become the "disturbances" that the other controllers must react to. The latter set of controller reactions, in turn, is reflected back as "disturbances" to the first controller, thereby triggering an action-reaction-action "vicious circle" which can , under appropriate conditions, *lead to the sudden, unexpected onset of chaos and instability of the whole platform system. This system instability behavior is <u>not</u> dependent*

on structural flexibility effects and can happen even though each equipment controller is, individually, quite stable. In such a situation, the otherwise desirable responsiveness of individual experiment controllers causes them to begin "fighting" each other, as if in conflict. A familiar example of this phenomena is the tragic consequences of seemingly innocuous pushing and shoving in a large, tightly-packed crowd of people, (here each person acts as an individually controlled element.)

1.3 Management of Multi-Controller Conflicts and Antagonisms

The early recognition of the inescapable controller conflicts and antagonisms, and the real possibilities of their suddenly triggering violent system instabilities, in the distributed multi-controller environment of a multiple pointing-mount space platform (MPMSP) should be considered as a major design and safety consideration for any MPMSP project. The importance of this consideration arises from the fact that the individual dynamic characteristics (i.e., settling-times, damping factors, etc.) of each individual experiment controller can have a critical influence on the stability of the overall platform system. A seemingly beneficial "re-tuning" of the controller "gains" associated with any one of those "individually stable" experiment controllers could, conceivably, trigger instability of the overall platform system, when in space. In fact, even a spatial relocation or reorientation of the mounting points for the moving equipment associated with an individual experiment module could, conceivably, trigger an instability of the overall platform system.

The unsettling aspect of this controller-induced destabilization phenomena is that,

owing to the inherent nonlinear nature of the overall system dynamics, the instability can, and usually will, be dependent on the occurrence of certain critical combinations of kinematic and kinetic conditions among the conflicting controllers and their respective experiments. Thus, the MPMSP system could, in fact, function quite well for some extended period of time until those critical dynamic conditions just happen to occur. Then, without warning, the whole platform system could suddenly become unstable.

There are primarily three approaches to managing multi-controller conflicts and antagonisms in MPMSP's. One approach consists of limiting the simultaneous operations of experiments to those that involve a negligible degree of conflicting control actions. This approach can lead to "one-at-a-time" operating scenarios. Another approach consists of re-designing the family of independent experiment controllers to work in a certain *strategically coordinated* manner that automatically mitigates excessive conflicts between controllers. This orchestration approach forces one to give up the individuality of the experiment activities, and imposes a hierarchical, centralized control-authority structure that can be quite complex and involve extensive communication links between the family of experiment controllers.

The third approach, which is the one considered in this study, consists of using a *platform controller* to impose a high-degree of "quietness" of the platform structure while the experiments are operating simultaneously. That is, the platform controller is designed to effectively suppress movements and vibrations of the platform due to the "disturbances" caused by experiment activities; see Figure 1.3. This maintenance of a "quiet platform," in the face of complex disturbance forces and moments induced by experiment pointing and

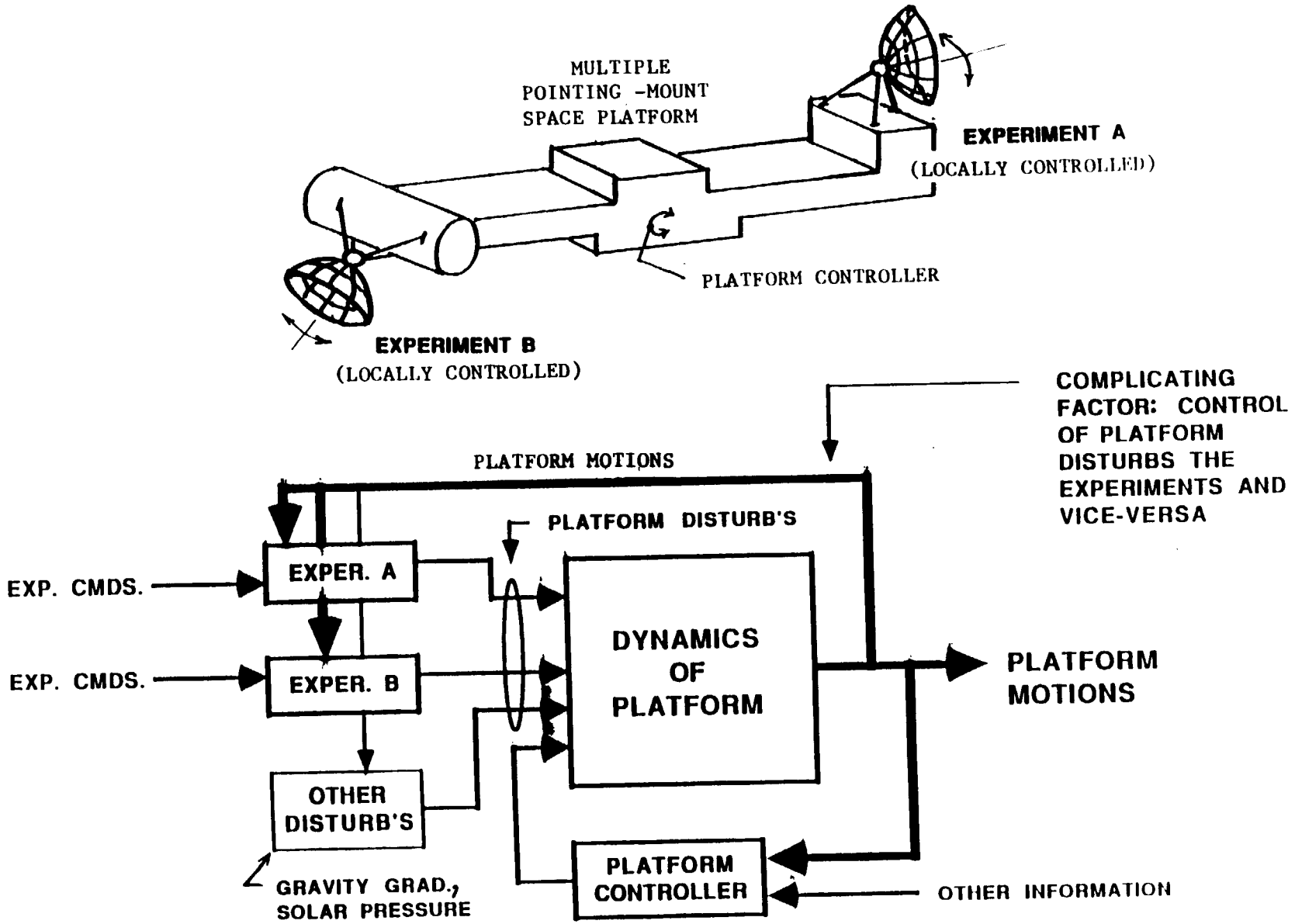


Figure 1.3 The "Quiet Platform" Approach to Multi-Controller Conflict Management as Used in This Study.

slewing motions, essentially breaks the "vicious circle" that allows one controller's action to be felt as a disturbance by the other controllers, and vice-versa. This mediating action clearly requires an exceptionally fast-acting and versatile "disturbance-adaptive" platform controller. Modern control theory has provided an extensive design methodology, known as Disturbance-Accommodating Control (DAC) Theory, for designing such disturbance-adaptive controllers and that theory will form the basis for the platform controller design developed in the present study.

1.4 Scope of This Research Effort

This research effort is directed at developing a new control concept that could form the technology basis for designing a high-performance platform controller for MPMSP-type projects. Because the present effort is directed at developing a control *concept*, and demonstrating the *credibility* of that control concept, a highly simplified, planar-motion 3(rigid)-element, multi-body model of a generic MPMSP has been adopted as the basic test-bed model for designing and demonstrating the proposed platform controller concept. Of course, any attempt to use the concepts and methodologies described herein on a real MPMSP project would necessitate consideration of the inevitable flexibility and out-of-plane motions of any real space platform, as well as the torque limitations of any realistic platform controller actuator. Nevertheless, the new platform control concept presented here is considered to be an innovative and viable candidate for consideration in any realistic design of a MPMSP.

Chapter 2

A 3-LINK MULTI-BODY GENERIC MODEL OF A MPMSP IN PLANAR MOTION

2.1 The Idea of a Concept-Demonstration Model

In Control Engineering, the initial development and effectiveness demonstration of a new control concept is typically conducted using simplified, low-order plant models (called "concept-demonstration" models) that permit one to focus attention on, and understand, the basic features of the control system behavior without being overwhelmed by dynamic complexities of the plant model. In this chapter we propose a concept-demonstration model of a generic multiple pointing-mount space platform. This model is a highly simplified, 3 (rigid, pinjointed)-link representation of the dynamic features of an MPMSP, moving in planar motion. Although it is highly simplified, this model embodies the essential features that make the platform control of an MPMSP a challenging problem. In Chapter 5 of this report, the effectiveness of the proposed platform controller concept will be demonstrated by computer simulation exercises, using mathematical models of the concept-demonstration model and the platform controller.

2.2 A 3-Link Generic Model of a MPMSP

The five essential features of an MPMSP platform control problem, from the control theoretic point-of-view, are:

1. A platform upon which two or more independent experiments are mounted.
2. Each experiment involves the pointing or slewing of equipment (telescopes,sensors, antennae, etc.) that has significant mass and/or rotational inertia.
3. The pointing/slewing motion of each experiment's equipment is controlled by its own individual (local) control system, designed for that particular experiment, and operated autonomously in accordance with the individual needs of that particular experiment.
4. The controlled equipment motions associated with each experiment induce reaction forces and/or moments on the platform.
5. The local control systems for pointing and/or slewing each experiment are designed to cope with uncertain-type "base motion" disturbances that arise from vibrations and transient motions of the platform mount.

Feature #4 implies that the angular (and linear) momentum vectors associated with equipment motions do not remain invariant during such motions. This feature occurs naturally in most equipment, except in those special cases where the equipment incorporates specially designed, "counter-inertia," mechanical assemblies that contain controlled, motor driven, counter-rotating inertia disks, which effectively cancel-out the angular momentum changes that would otherwise occur when, say, a heavy instrument is rotated in its mount. Indeed, if all equipment motions on an MPMSP were accompanied by such "counter-inertia" devices, there would be no platform disturbances or controller conflicts to contend with...but

there would then be a significant increase in cost, weight, equipment complexity, and system power consumption. Feature #5 is a common control system requirement for any precision pointing/slewing experiment mounted on a space platform, and typically leads to the use of some form of "integral-feedback" in the controller.

Any space platform big enough to be used as a mounting base for multiple experiments will involve some structural flexibility. This latter feature has not been included in the above list. The reason for this omission is the quantum increase in modeling complications and model complexity that such flexibility considerations would entail. Specifically, consideration of torsional flexibility would force consideration of coupled out-of-plane motions, which would lead to the modeling of 3-dimensional dynamics---a complication that would exceed the resources budgeted for this project. Moreover, inclusion of the platform's in-plane, lateral flexibility would add enormous complications to the already difficult task of developing the exact equations of in-plane motion of the "simplified" concept demonstration model. (See Chapter 4 of this report.)

The simplest configuration of mechanical elements that embody the five essential MPMSP features listed above is shown in Figure 2.1 and consists of three co-planar rigid links, pin-jointed together as shown. The center link represents the space platform, together with the non-moving experiment equipment, while the two unsymmetrical, but co-planar, end-links represent respectively, the pointing/slewing equipment associated with two independent experiments, which are presumed mounted at either end of the platform. The (co-planer) rotational movements of each end-link, with respect to the platform, are assumed

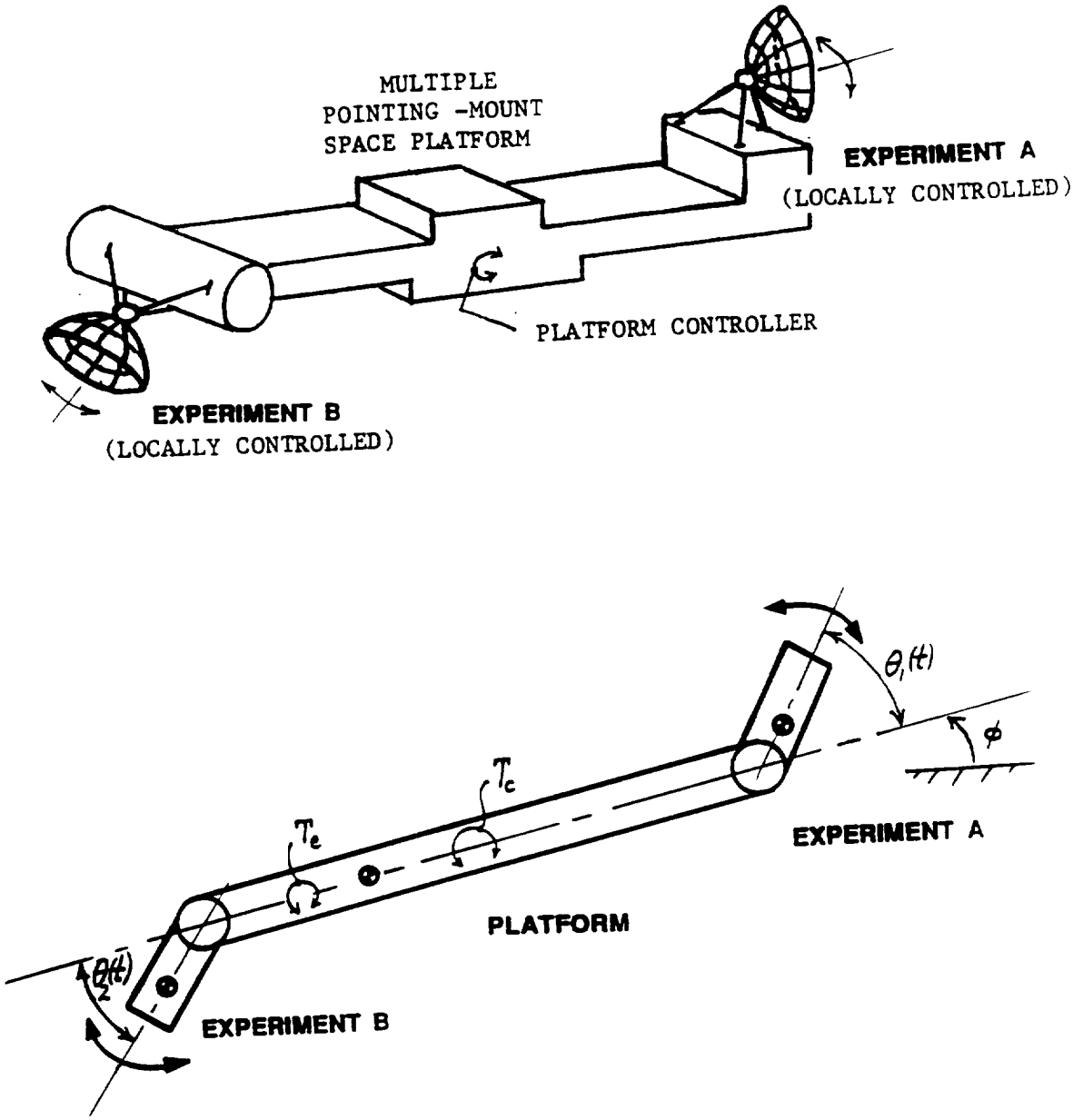


Figure 2.1 A 3-Link Generic Model of an MPMSP in Planar Motion (All links are assumed rigid.).

to be controlled by an independent motor, or toque device (torquer), located at each pin-joint connection, and such that the controlled torques exerted on each end-link (on each experiment equipment) results in an equal-but-opposite reaction torque exerted on the center link (on the platform). The entire assembly shown in Figure 2.1 is assumed to be positioned in space, moving in (locally) planar-motion, in an essentially zero-gravity environment (e.g. in orbit around the earth). Thus, in the course of mathematically modeling the dynamics of the "configuration model" in Figure 2.1, one can ignore the usual gravity forces that would be imagined, say, as acting through the centers of gravity of the respective links. However, it may be necessary to include in that mathematical model the small, but persistently acting, "gravity-gradient torques" that will act on such an assembly in orbit. Our exact dynamic mathematical model of Figure 2.1, developed in Chapter 4, will include a generic "external disturbance torque" term T_e to account for such gravity gradient torques, as well as solar-pressure effects, etc.

2.3 The Platform Control Problem for the 3-Link Model

The platform, or center-link, in Figure 2.1 is assumed to contain its own "platform torquer device" (e.g. a C.M.G. device) that can be controlled to exert precision, quick-acting, in-plane torques T_c on the platform as determined by the platform control algorithm. (The design of this platform control algorithm is the main task of this research effort.) The locations of the centers of gravity, as well as the mass and rotational inertia-values of each link shown in Figure 2.1, are considered to be completely arbitrary in this study.

In long-range pointing tasks, such as envisioned for experiments on an MPMSP, angular pointing errors due to platform angular base-motion disturbances are far more serious than errors associated with platform rectilinear base-motion disturbances. Thus, in accordance with the "quiet platform" approach to be used in this study for controller conflict mediation (the third approach cited in Chapter 2), the task of the platform controller is to achieve and maintain the "quiet" angular condition

$$\phi\,(t) \approx o \tag{2.a}$$

$$\dot{\phi}\,(t) \approx 0 \tag{2.b}$$

in the face of arbitrary angular motions $\theta_1(t)$, $\theta_2(t)$ of the two end-links shown in Figure 2.1. Moreover, the motions $\theta_1(t)$, $\theta_2(t)$ are assumed to be controlled by independently acting controllers in accordance with the real-time requirements of each experiment (each end-link); see Feature 3 in Section 2.2.

Chapter 3

DESIGN OF A DISTURBANCE-ADAPTIVE MPMSP CONTROL STRATEGY FOR THE PLANAR MOTION GENERIC MODEL

The controller design methodology used in this chapter is based on the theory of Disturbance-Accommodating Control (DAC). We begin with a brief tutorial review of the principles of DAC theory.

3.1 DAC Waveform Models of Uncertain Disturbances

The theory of Disturbance-Accommodating Control [1]-[4] is concerned with non-statistical modeling and controller design techniques for systems subjected to uncertain, unmeasurable, time-varying, multi-variable disturbances $w(t) = (w_1(t), w_2(t), ..., w_p(t))$.

In the remainder of this chapter we will focus on the special case $p=1$ of an equivalent *scalar* (single-input) disturbance $w(t)$; see [2] for details of the theory for vector disturbances $p>1$. The central idea in DAC disturbance modeling is the concept of a ***waveform model***, which is simply a representation of $w(t)$ as an unknown weighted linear combination of completely known ***basis-functions*** $\{f_1(t), f_2(t), ...,f_m(t)\}$ of the form

$$w(t) = C_1 f_1(t) + C_2 f_2(t) + ... + C_m f_m(t) \qquad (3.1)$$

where the $\{C_1, C_2, ..., C_m\}$ are scalar weighting coefficients that are completely unknown "constants", which may occasionally jump in value in a once-in-awhile manner. This sparse jumping behavior of the "constants" C_i is referred to as "stepwise-constant" behavior and

is illustrated in Figure 3.1. Disturbances that can be effectively represented by an expression of the form (3.1) are said to have *waveform-structure*; those that cannot are referred to as *noise* disturbances. The waveform-model representation (3.1) is a generalized spline-function model as used in *approximation theory* , and can be viewed as an extension of the idea of a Fourier-series representation.

The collection $\{f_i(t)\}$ of presumed known basis functions in (3.1) is chosen by the user to reflect the actual patterns of w(t) time-behavior as seen in experimental data, etc. Thus, if w(t) characteristically exhibits a sinusoidal pattern of behavior, with *known* frequency ω and *unknown* stepwise-constant amplitude and phase, one would write (3.1) as

$$w(t) = C_1 \sin \omega t + C_2 \cos \omega t \tag{3.2}$$

Likewise, if the uncertain time-behavior of w(t) has the generic piecewise-linear (step plus ramp) characteristic as shown in Figure 3.2, one would write (3.1) as

$$w(t) = C_1 1 + C_2 t \tag{3.3}$$

In some industrial applications, the characteristic time-behavior of w(t) is rather varied and undistinguished, as shown in Figure 3.3. In such cases, an effective choice for the basis-functions $\{f_i(t)\}$ is the *polynomial* basis-set $\{1,t,t^2, ..., t^{(m-1)}\}$. The corresponding representation (3.1) then becomes the *polynomial-spline* waveform model

$$w(t) = C_1 1 + C_2 t + C_3 t^2 + ... + C_m t^{(m-1)}, \tag{3.4}$$

which is usually quite effective in modeling slow, meandering-type functions w(t), even when one chooses the relatively small value m=3, (the so-called *quadratic-spline* model).

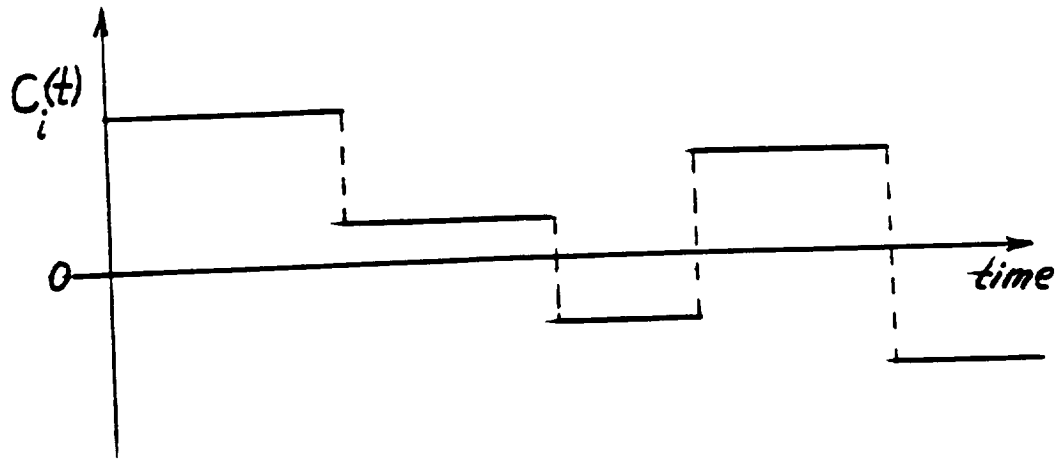


Figure 3.1 Stepwise-Constant Time-Behavior of the Weighting "Constants" C_i in (3.1).

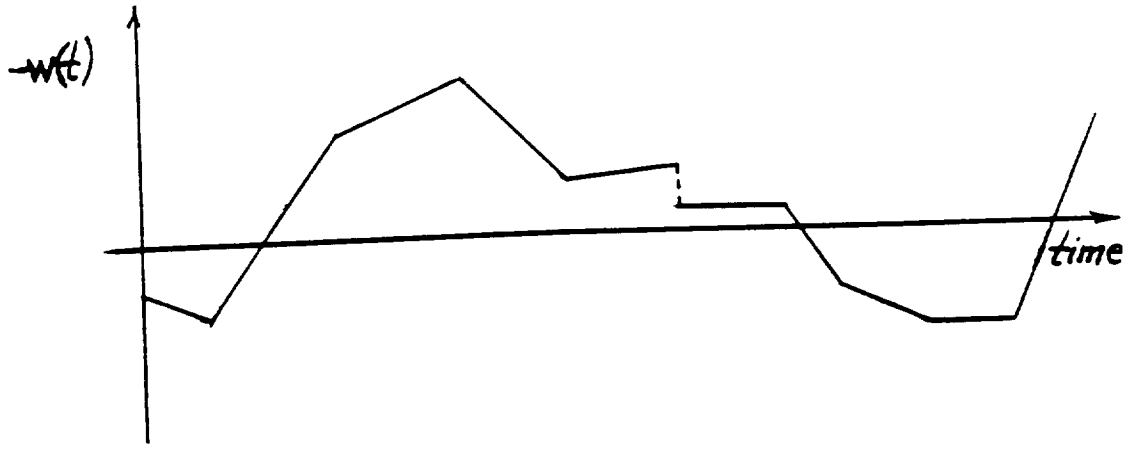


Figure 3.2 An Uncertain Disturbance w(t) Have Piecewise-Linear (step-plus-ramp) Time-Behavior.

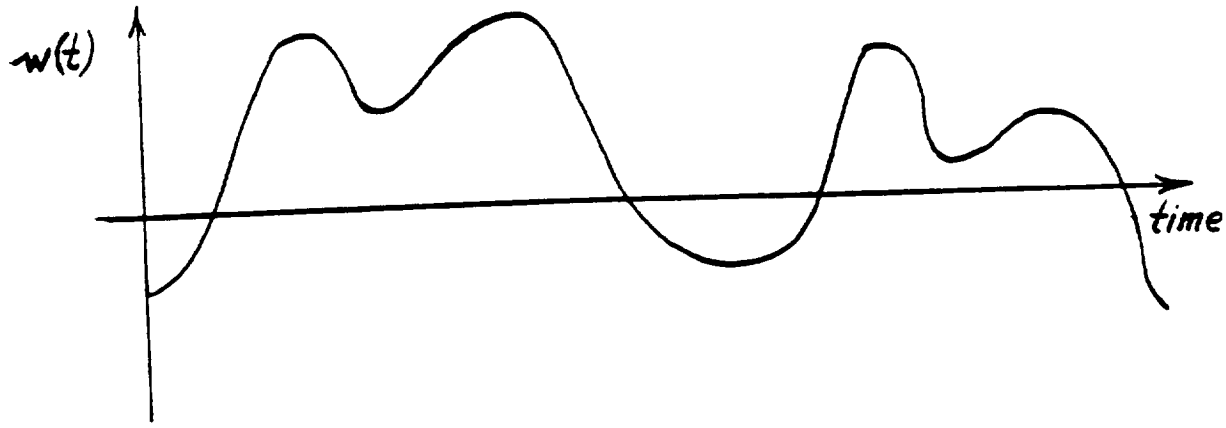


Figure 3.3 An Uncertain Disturbance w(t) Having Undistinguished, Meandering Time-Behavior.

3.2 Conversion of a Disturbance Waveform Model to a State Model

The design of a DAC controller is based on the technique of on-line identification of uncertain disturbances having waveform-structure [2]. This on-line identification of uncertain disturbances w(t) is accomplished by a conventional state-observer or Kalman filter, which processes the system control input u(t) and output measurement y(t) data to produce an accurate real-time estimate $\hat{w}$ (t) of the actual disturbance input w(t). To do this, it is necessary to have a differential equation model of the waveform behavior of w(t). That is, one must find a differential equation for which (3.1), with constant C_i, is the (a) solution. In practice an effective choice of basis-functions $\{f_i(t)\}$ can usually be found in the class of functions that satisfy some *linear time-invariant* differential equation; expressions (3.2), (3.3), (3.4) are common examples of such cases. In such linear cases, the differential equation corresponding to (3.1) will have the form

$$\frac{d^{\beta}w}{dt^{\rho}} + \beta_{\rho}\frac{d^{\rho-1}w}{dt^{\rho-1}} + \ldots + \beta_2\frac{dw}{dt} + \beta_1 w = 0 \qquad (3.5)$$

where ρ and the $\{\beta_1, \beta_2, \ldots, \beta_{\rho}\}$ are *completely determined* by the (known) basis functions $\{f_1(t), \ldots f_m(t)\}$. For instance, in the case (3.2) the corresponding differential equation model (3.5) is

$$\frac{d^2w}{dt^2} + \omega^2 w = 0 \qquad (3.6)$$

Likewise in the case (3.3) expression (3.5) becomes

$$\frac{d^2w}{dt^2} = 0, \qquad (3.7)$$

and for (3.4) the counterpart of (3.7) is

$$\frac{d^m w}{dt^m} = 0. \tag{3.8}$$

The observer, or Kalman filter, used to generate the disturbance estimate $\hat{w}(t)$ requires that (3.5) be re-written in the form of a state-variable model. The general form of such a linear disturbance state-model for a *scalar* disturbance w(t) is

$$w = h^T z; \qquad h,z = \rho - \text{vectors}\ ;\ (\cdot)^T - \text{denotes transpose} \tag{3.9a}$$

$$\dot{z} = Dz + \sigma(t)\ ; \quad D = \rho \times \rho \text{ matrix} \tag{3.9b}$$

where (h,D) can be chosen as any one of the canonical completely-observable pairs [2; pg.418]. The term $\sigma(t) = (\sigma_1(t), \ldots, \sigma_\rho(t))$ in (3.9) denotes a vector of *totally unknown*, sparse sequences $\sigma_i(t)$ of Dirac impulses, which arrive in a random, once-in-a-while manner with unknown intensities. The unknown impulses of $\sigma_i(t)$ represent the "cause" of the once-in-a-while jumping of the C_i as shown in Figure 3.1. Note that $\sigma(t)$ in (3.9) is <u>not</u> "white noise" as is commonly used in stochastic control theories. The constant matrix D in (3.9) embodies the collection of basis functions $\{f_i(t)\}$ as characteristic eigenfunctions of D. Thus. D may be an <u>unstable</u> matrix, even though w(t) in (3.9) always remains bounded. In DAC theory, the ρ-vector $z=(z_1, \ldots, z_\rho)$ is called the *state* of the disturbance w(t).

To illustrate the procedure for developing a state-model (3.9), consider the differential equation model (3.6). If we define the disturbance state-variables (z_1, z_2) for (3.6) as

$$z_1 = w \ ; z_2 = \dot{w} \tag{3.9c}$$

then, clearly

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} \sigma_1(t) \\ \sigma_2(t) \end{pmatrix}; \; w = (1,0) \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \tag{3.9d}$$

where the sparse impulses of $\sigma_1(t)$, $\sigma_2(t)$ account for the unknown, random-like once-in-a-while jumps in w(t) and/or $\dot{w}(t)$, corresponding to jumps in C_1 and C_2 in (3.2).

3.3 On-Line Identification of w(t) via a Composite State-Observer/Kalman Filter

The next step in DAC controller design is to incorporate (endogenize) the disturbance state-model (3.9) with the plant state-model. For this purpose, suppose, for simplicity, that the plant with scalar disturbance is represented by a linear, time-invariant state-model of the form

$$\dot{x} = Ax + Bu + fw \; ; \; u = \text{r-vector control input,} \tag{3.10a}$$

$$y = Cx \qquad ; \; y = \text{m-vector output measurement,} \tag{3.10b}$$

The disturbance state-model (3.9) may now be incorporated into (3.10) to obtain the following *composite* plant/disturbance state-model

$$\begin{pmatrix} \dot{x} \\ \dot{z} \end{pmatrix} = \left[\begin{array}{c|c} A & fh^T \\ \hline O & D \end{array}\right] \begin{pmatrix} x \\ z \end{pmatrix} + \begin{bmatrix} B \\ \hline 0 \end{bmatrix} u + \begin{pmatrix} 0 \\ \sigma(t) \end{pmatrix} \tag{3.11}$$

which can be written in the compact form

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u + \tilde{\sigma} \tag{3.12a}$$

$$y = \tilde{C}\tilde{x}$$

where

$$\tilde{x} = \begin{pmatrix} x \\ \overline{z} \end{pmatrix} \; ; \; \tilde{A} = \left[\begin{array}{c|c} A & fh^T \\ \hline O & D \end{array}\right] \; ; \; \tilde{B} = \left[\begin{array}{c} B \\ \hline 0 \end{array}\right] ; \; \tilde{C} = [C \,|\, 0] \; ; \; \tilde{\sigma} = \begin{pmatrix} 0 \\ \overline{\sigma(t)} \end{pmatrix} . \tag{3.12b}$$

A standard full-order, or reduced-order, *observer* can now be designed to generate real-time estimates $\hat{\tilde{x}}$ (t)in (3.12). Alternatively, if the plant measurement y(t) in (3.10b) contains additive "measurement noise" of the form

$$y = Cx + \eta(t) \; , \; \eta(t) = \text{measurement noise} \tag{3.13}$$

one then can use a standard *Kalman filter* to generate the real-time, minimum-square-error estimate $\hat{\tilde{x}}$ (t) . In either case, the corresponding disturbance estimate $\hat{w}(t)$ is obtained as

$$\hat{w}(t) = h^T \hat{z} \; (t) = [0 \,|\, h] \begin{pmatrix} \hat{x} \\ \overline{\hat{z}} \end{pmatrix} = \tilde{h}^T \hat{\tilde{x}} \quad . \tag{3.14}$$

To demonstrate this observer design methodology, recall that a full-order observer for (3.12a) has the well-known form [2; pg. 432]

$$\dot{\hat{\tilde{x}}} = \tilde{A} \; \hat{\tilde{x}} + \tilde{B} u - \tilde{K}_0 \left[y \; (t) - \tilde{C} \; \hat{\tilde{x}} \right] \tag{3.15}$$

where the observer gain matrix $\tilde{K}_0$ is designed to make the estimation error $\epsilon = \left(\tilde{x} - \hat{\tilde{x}} \right)$ rapidly approach zero between arrivals of the sparse, unknown impulses of $\sigma(t)$. It is easy to show from (3.12), (3.15) that, between impulses of $\sigma(t)$, $\epsilon(t)$ obeys the vector-matrix homogenous differential equation

$$\dot{\epsilon} = \left[\tilde{A} + \tilde{K}_0 \; \tilde{C} \right] \epsilon \tag{3.16}$$

Thus, to make $\epsilon(t) \rightarrow 0$ promptly, $\tilde{K}_0$ should be chosen to place the eigenvalues of

$\left[\tilde{A} + \tilde{K}_0 \tilde{C} \right]$ sufficiently deep in the left-half of the complex plane.

3.4 Design of the DAC Control Law

In DAC theory, there are a variety of ways a control system can "accommodate" the disturbances w(t) that act on a given plant. The most common method of accommodation is to design the controller to exactly *counteract* (reject) the total effect of w(t) on the plant state x(t). To accomplish this, we first agree to split the total control effort u(t) into two parts

$$u = u_p + u_d \tag{3.17}$$

where u_d will be designed to exactly cancel the disturbance effects on x(t) and where u_p is then designed to accomplish the primary control task (set-point regulation, servo-tracking, etc.) for the *undisturbed* plant. If the plant state model is given by (3.10), the incorporation of (3.17) yields

$$\dot{x} = Ax + Bu_p + Bu_d + fw(t) \tag{3.18}$$

Thus, to completely cancel w(t) in (3.18), one should design u_d to satisfy

$$Bu_d = -fw(t) = -fh^T z(t) \quad , z \epsilon E^\rho \tag{3.19}$$

The necessary and sufficient condition for existence of a solution u_d to (3.19) is the *total cancellation* condition [2].

$$\text{rank } [B \mid f] = \text{rank } [B] \tag{3.20a}$$

which implies

$$f = B\gamma \tag{3.20b}$$

for some (possibly non-unique) vector γ. Assuming (3.20) holds a control u_d satisfying

(3.19) is given (ideally) by

$$u_d = - \gamma z \tag{3.21}$$

In practical applications, the term z(t) in (3.21) would be replaced by the real-time estimate $\hat{z}(t)$ obtained from a real-time observer or Kalman filter (3.15). If condition (3.20) fails to be satisfied, it is impossible for the control action to cancel-out all the effects of w(t) on x(t). In that case, there are a variety of alternative modes of disturbance accommodation one can consider, such as "disturbance minimization;" see [2].

Assuming u_d can be designed as in (3.21), the remaining part u_p of u can be designed to accomplish the primary control task, using conventional design methods and setting $Bu_d + fw(t) \equiv 0$ in (3.18). This is a well-known standard procedure in modern control theory. This completes our brief tutorial review of DAC principles. We will now apply those principles to the platform controller design for the generic MPMSP model developed in Chapter 2.

3.5 Design of a Disturbance Estimator for the MPMSP Generic Model

The uncertain disturbances associated with the platform control of an MPMSP are ideally suited for representation by a waveform-model (3.1). To see this, recall the arrangement in Figure 2.1 and consider the corresponding platform "free-body diagram" shown in Figure 3.4.

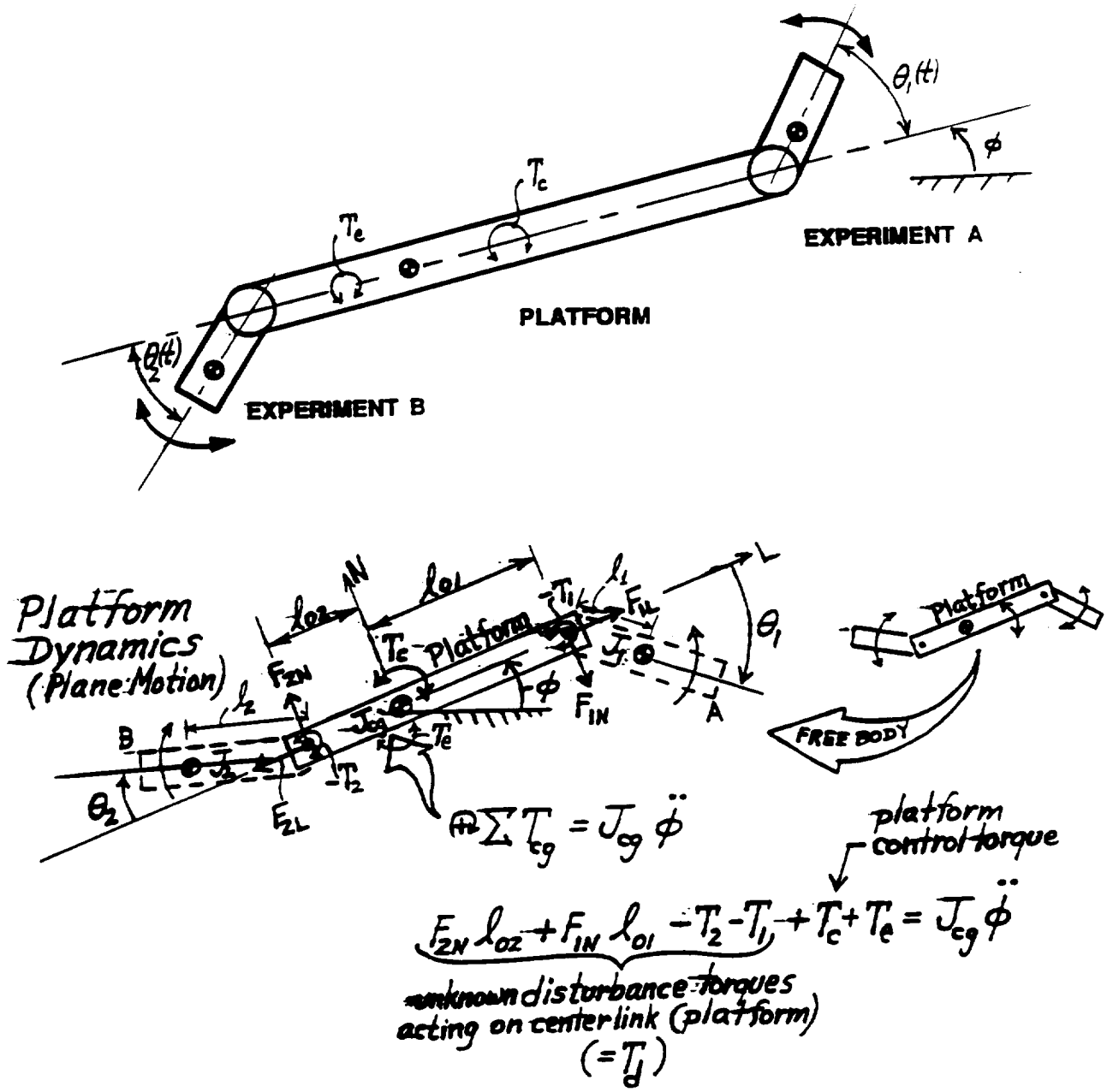


Figure 3.4 "Free-Body Diagram" Analysis of the Platform Dynamics for the 3-Link Generic MPMSP Model.

The disturbing forces and moments that act on the platform consist of:

(i) the pin-joint reaction forces associated with the movements of each end-link.

(ii) the torque reactions associated with local controller torques that act on each end-link, at the respective pin-joints, and

(iii) the external torques that act on the platform through the effects of gravity-gradients, solar pressures on the solar collectors, etc.

Referring to the geometry of Figure 3.4, it can be seen that the basic Newtonian equation governing platform angular dynamics

$$\circlearrowright \sum (\text{ torques})_{cg} = J_{cg}\,\ddot{\phi} \tag{3.22}$$

has the specific form

$$-T_1-T_2 + F_{2N}\,\ell_{02} + F_{1N}\ell_{01} + T_e + T_c = J_{cg}\,\ddot{\phi} \tag{3.22b}$$

where

T_1, T_2 = local controller torques that act on each end link to control the experiment equipment motions $\theta_1(t)$, $\theta_2(t)$

T_c = platform controller torque

T_e = net external torque on platform due to combined effects of gravity gradients, solar pressures on the solar panels, etc.

F_{1N}, F_{2N} = the components of the pin-joint reaction forces that act on the platform and are *normal* to the longitudinal axis of the platform

ℓ_{01}, ℓ_{02} = distances from the pin-joints to the platform center of gravity.

In order to design an effective, disturbance-adaptive platform controller, $T_c = T_c(?)$, it is important to decide which torque-terms on the left side of (3.22b) are likely to behave as "uncertain disturbances"; i.e., torques that are *not* reliably known and/or *not* directly and

reliably measurable (in real-time), in a realistic MPMSP project. As will be shown in Chapter 4 [see Eqs. (4.1), (4.3)], the exact mathematical expressions for the normal components F_{1N}, F_{2N} of the pin-joint reaction forces are incredibly long and complicated functions of $\{\phi, \dot{\phi}, \ddot{\phi}, \theta_i, \dot{\theta}_i, \ddot{\theta}_i, J_i, M_i, \text{etc.}\}$ thereby rendering the accurate on-line computation or measurement of $F_{1N}(t)$, $F_{2N}(t)$ rather impractical. The local controller torques $T_1(t)$, $T_2(t)$ should be relatable to the electrical currents in the respective torque motors. However, there may be some degree of liability (or excessive risk) in using such a scheme to measure $\{T_1(t), T_2(t)\}$ in real-time. Thus, in this study, we elected to consider $\{-T_1(t), -T_2(t)\}$ as uncertain, unmeasurable reaction torques acting on the platform.

The inherent uncertainty as to the exact kinematic configuration of all the moving parts of all the experiments, solar-panels, antennae, etc., mounted on the MPMSP, suggests that the time-behavior of the gravity gradient and solar pressure torques would be difficult to compute on-line or predict *a priori*, and should, therefore, be viewed as uncertain, unmeasurable disturbance torques.

In summary, for this study, we will adopt the "worst case" in regards to the ability to measure disturbances; namely, the total net uncertain,unmeasurable disturbance torque $T_d(t)$ acting on the platform in Figure 3.4 will be *defined* as:

$$T_d(t) \triangleq - T_1(t) - T_2(t) + F_{2N}(t)\, \ell_{02} + F_{1N}(t)\ell_{01} + T_e \tag{3.23}$$

In view of (3.23), expression (3.22b) can be written as

$$T_d + T_c = J_{cg}\, \ddot{\phi} \tag{3.24a}$$

3.5.1 A Subtle Fact About (3.24a):

Owing to the fact that the reaction forces $F_{1N}(\cdot)$, $F_{2N}(\cdot)$ in (3.23) contain terms that are *explicit* functions of the platform angular acceleration $\ddot{\phi}$, it is necessary, for controller design purposes, to write (3.24) in the modified form,

$$\tilde{T}_d + T_c = \tilde{J}_{cg}\ddot{\phi} \qquad (3.24b)$$

where $\tilde{T}_d$ denotes the *remainder* of T_d after the $\ddot{\phi}$-related terms of T_d have been removed from T_d and combined with the right side of (3.24a) to augment the "effective" J_{cg} and form the new expression $\tilde{J}_{cg}\ddot{\phi}$. (Note that $\tilde{J}_{cg}$ in (3.24b) is typically a nonlinear function of $\{\phi, \dot{\phi}, \theta_i, \dot{\theta}_i\}$.) If this step is not invoked, the design of a disturbance estimator as described below becomes extremely difficult. This is a rather unusual and potentially confusing consideration in DAC theory that has not been heretofore discussed in the literature. With respect to the augmented inertia model (3.24b) the platform control problem is to estimate and cancel the term $\tilde{T}_d(t)$ and regulate $\phi(t) \to 0$.

3.5.2 Waveform Characterization of $\tilde{T}_d$

The various physical sources which create or originate the components of $\tilde{T}_d(t)$, as indicated in (3.23b), are all characterized as producing essentially smoothly evolving, dynamic torques with the possibility of simple jump-behavior occurring once-in-awhile, (due, for instance, to sudden reversals of the local controller's torque motors, etc.). Thus it can be anticipated that a typical time-plot of $\tilde{T}_d(t)$ would be as shown in Figure 3.5. The generic meandering behavior of $\tilde{T}_d(t)$ shown in Figure 3.5 suggests in general that $\tilde{T}_d(t)$ may have a waveform structure in which no distinguishing periodic, or other specific basis functions.

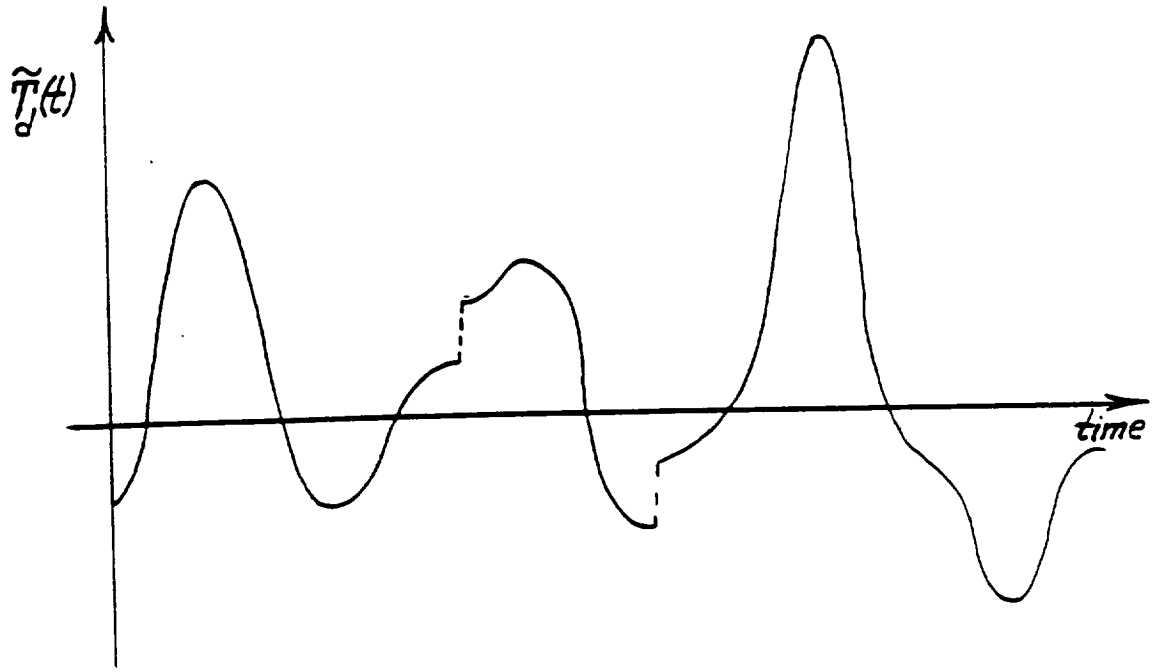


Figure 3.5 Generic Meandering Time-Behavior of $\tilde{T}_d(t)$.

are apparent. This case is one in which a polynomial-spline waveform model of the type (3.4) is appropriate. Thus, in the interest of keeping things simple, our first trial design of the platform controller will be based on modeling the uncertain time-behavior of $\tilde{T}_d(t)$ by the *quadratic-spline* waveform model

$$\tilde{T}_d(t) = C_1 + C_2\, t + C_3 t^2 \tag{3.25}$$

where the "constant" coefficients $\{C_1, C_2, C_3\}$ in (3.25) are allowed to jump in value in a sparse, once-in-a-while manner (= "stepwise 3.6 - constant"); see Figure 3.1.

The model (3.25) is capable of emulating a rich variety of meandering uncertain disturbance behaviors such as shown in Figure 3.5. In fact, the waveform model (3.25) can effectively emulate $\tilde{T}_d(t)$ behavior containing "slow," unknown *sinusoidal* components, unknown *exponential* components, etc., provided the associated unknown frequencies, time-constants, etc. are sufficiently "small;" see the "disturbance estimator" performance plots in Section 3.7.

A disturbance state-model (3.9) corresponding to (3.25) is developed as follows. First, define the three disturbance state-variables z_1, z_2, z_3 as

$$z_1 = \tilde{T}_d \;\; ; \;\; z_2 = \dot{\tilde{T}}_d \;\; ; \;\; z_3 = \ddot{\tilde{T}}_d \tag{3.26}$$

Then, note that, for *constant* C:

$$\dot{z}_1 = z_2\left(= C_1 + 2C_3 t\right) \; ; \;\; \dot{z}_2 = z_3\left(= 2C_3\right) \; ; \; \dot{z}_3 = 0 \; , \tag{3.27}$$

Thus, letting the random-like $\sigma_i(t)$-impulses account for the sparse jumps that occur in $(C_1$.

C_2, C_3), one can write (3.27) in the form of (3.9) as

$$\tilde{T}_d = (1,0,0)\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \tag{3.28a}$$

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} \sigma_1(t) \\ \sigma_2(t) \\ \sigma_3(t) \end{pmatrix} \tag{3.28b}$$

The on-line estimation of $\tilde{T}_d(t)$ in (3.24), from the assumed available on-line measurements of $\{\phi(t), T_c(t)\}$ is accomplished by use of a composite state-observer of the type (3.15). For this purpose the composite plant/disturbance model (3.11) is obtained from (3.24b), (3.28) as follows. The plant state variables can be defined as

$$x_1 = \phi \; ; \; x_2 = \dot{\phi} \tag{3.29}$$

Then, the corresponding plant state-model (3.10) becomes

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \tilde{J}_{cg}^{-1} \end{pmatrix} T_c + \begin{pmatrix} 0 \\ \tilde{J}_{cg}^{-1} \end{pmatrix} \tilde{T}_d \tag{3.30a}$$

$$y = \phi = (1,0)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{3.30b}$$

Combining (3.30) and (3.28) then yields the composite state-model (3.11) as

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \tilde{J}_{cg}^{-1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \tilde{J}_{cg}^{-1} \\ 0 \\ 0 \\ 0 \end{pmatrix} T_c + \begin{pmatrix} 0 \\ 0 \\ \sigma_1(t) \\ \sigma_2(t) \\ \sigma_3(t) \end{pmatrix} \tag{3.31a}$$

$$y = \phi = (1,0,0,0,0) \begin{pmatrix} x_1 \\ x_2 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} \tag{3.31b}$$

The full-order composite-state observer (3.15), corresponding to (3.31), can now be written out in full as

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 - k_{01}(y - \hat{x}_1) \\ \dot{\hat{x}}_2 &= \tilde{J}_{cg}^{-1}\, \hat{z}_1 + \tilde{J}_{cg}^{-1}\, T_c - k_{02}(y - \hat{x}_1) \\ \dot{\hat{z}}_1 &= \hat{z}_2 - k_{03}(y - \hat{x}_1) \\ \dot{\hat{z}}_2 &= \hat{z}_3 - k_{04}(y - \hat{x}_1) \\ \dot{\hat{z}}_3 &= - k_{05}(y - \hat{x}_1) \end{aligned} \tag{3.32}$$

and the associated estimation-error dynamics (3.16) is given by

$$\begin{pmatrix} \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \\ \dot{\epsilon}_4 \\ \dot{\epsilon}_5 \end{pmatrix} = \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \bar{J}_{cg}^{-1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{pmatrix} k_{01} \\ k_{02} \\ k_{03} \\ k_{04} \\ k_{05} \end{pmatrix} (1,0,0,0,0) \right\} \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix} \qquad (3.33)$$

Thus, in accordance with the remarks below (3.16) the designer should choose the observer "gains" $\{k_{01}, k_{02}, k_{03}, k_{04}, k_{05}\}$ in (3.22) to place the five eigenvalues $\{\lambda_{01}, \lambda_{02}, \lambda_{03}, \lambda_{04}, \lambda_{05}\}$ of the composite matrix

$$[\tilde{A} + \tilde{K}_0 \tilde{C}] = \begin{bmatrix} k_{01} & 1 & 0 & 0 & 0 \\ k_{02} & 0 & \bar{J}_{cg}^{-1} & 0 & 0 \\ k_{03} & 0 & 0 & 1 & 0 \\ k_{04} & 0 & 0 & 0 & 1 \\ k_{05} & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.34)$$

sufficiently deep in the left-half of the complex-plane.

The *characteristic polynomial* $\rho(\lambda)$ of (3.34) is easily calculated to be

$$\rho(\lambda) = \lambda^5 - k_{01}\lambda^4 - k_{02}\lambda^3 - k_{03}\bar{J}_{cg}^{-1}\lambda^2 - k_{04}\bar{J}_{cg}^{-1}\lambda - k_{05}\bar{J}_{cg}^{-1} = 0 \qquad (3.35)$$

Now suppose the *desired* values of the five eigenvalues λ_{oi} of (3.35) are denoted by $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$. Then the corresponding *desired* characteristic polynomial $\rho_d(\lambda)$ for (3.34) can be computed by the formula

$$\rho_d(\lambda) = (\lambda-\lambda_1)\cdot(\lambda-\lambda_2)\cdot(\lambda-\lambda_3)\cdot(\lambda-\lambda_4)\cdot(\lambda-\lambda_5) \quad (3.36a)$$

$$= \lambda^5 + \alpha_5\lambda^4 + \alpha_4\lambda^3 + \alpha_3\lambda^2 + \alpha_2\lambda + \alpha_1 \quad (3.36b)$$

where the $\{\alpha_i\}$ will be precisely determined by the desired values of the $\{\lambda_i\}$ It is now a simple matter to equate corresponding coefficients in (3.35) and (3.36b) to obtain the following explicit design formulae for the observer gains $\{k_{oi}\}$.

$$\begin{aligned} k_{01} &= -\alpha_5 \\ k_{02} &= -\alpha_4 \\ k_{03} &= -\bar{J}_{cg}\alpha_3 \\ k_{04} &= -\bar{J}_{cg}\alpha_2 \\ k_{05} &= -\bar{J}_{cg}\alpha_1 \quad ; \ \alpha_i \text{ – given by (3.36b)} \end{aligned} \quad (3.37)$$

The set of equations (3.32) together with the gain formulae (3.37) constitute the complete solution for the disturbance-identifier (estimator) for this generic MPMSP problem, where

$$\hat{\tilde{T}}_d = \hat{z}_1(t) \quad (3.38)$$

3.6 Design of a Disturbance-Adaptive Controller for the MPMSP Generic Model

The rotational motions $\phi(t)$ of the platform (center-link) in the MPMSP generic model shown in Figures 2.1 and 3.4 are governed by the inertia-augmented Newtonian equation of motion (3.24b) where $\tilde{T}_d$ is *defined* as that part of (3.23) that does *not* depend explicitly on $\ddot{\phi}$, and where the $\tilde{T}_d(t)$ waveform is *modeled* by (3.25), (3.28). As stated in Chapter 2, see Equation (2), the task of the platform controller T_c is to achieve and

maintain the "quiet" angular condition $\phi(t) \approx 0$, $\dot{\phi}(t) \approx 0$ in the face of all expected external torques $T_e(t)$ and arbitrary motions $\theta_1(t)$, $\theta_2(t)$ of the experiment equipment (the end links).

Using the standard control engineering symbolism

$$u = \text{control-input} = T_c \tag{3.39}$$

in (3.24), together with the standard DAC control-splitting technique (3.17), one obtains (3.24b) in the form

$$\tilde{J}_{cg}\ddot{\phi} = u_p + u_d + \tilde{T}_d(t) \tag{3.40}$$

the general DAC design ideas summarized in Section 3.4, it is clear from (3.40) that one certainly can design u_d in (3.40) to (ideally) counteract the $\tilde{T}_d(t)$ effect on $\phi(t)$ by simply choosing

$$u_d = -\tilde{T}_d(t) \quad \text{(ideal design)} \tag{3.41}$$

Moreover, with $\tilde{T}_d(t)$ so cancelled, the remaining control term u_p can be chosen (ideally) as the classical "proportional and derivative" feedback law

$$u_p = -k_1 \phi - k_2 \dot{\phi} \qquad (k_1, k_2) > 0 \tag{3.42}$$

to achieve the condition that $\phi(t) \to 0$; $\dot{\phi}(t) \to 0$ promptly, from any initial conditions. In particular, if k_1, k_2 are chosen as

$$k_1 = \tilde{J}_{cg}\,\omega_n^2 \;;\; k_2 = 2\,\tilde{J}_{cg}\,\zeta\,\omega_n \tag{3.42}$$

the corresponding closed-loop equation of motion (3.40) becomes (ideally) the classical damped 2nd-order linear system

$$\ddot{\phi} + (2\,\zeta\,\omega_n)\dot{\phi} + (\omega_n^2)\phi = 0 \quad \text{(ideal)} \tag{3.43}$$

so that in (3.42) the designer can select the "damping factor" $\zeta > 0$ and "undamped natural frequency" $\omega_n > 0$ to achieve the desired qualitative and quantitative behavior of $\phi(t)$ as $\phi(t) \to 0$ in (3.43).

In summary, the (idealized) control $u = T_c$ for the platform is given by

$$\begin{aligned} u &= T_c = u_p + u_d \\ &= -k_1 \phi - k_2 \dot{\phi} - \tilde{T}_d(t) \qquad \text{(ideal)} \end{aligned} \tag{3.44}$$

In a realistic practical application, the term $\tilde{T}_d(t)$ in (3.44) would be replaced by the real-time *estimate* $\hat{\tilde{T}}_d(t)$ as obtained from (3.38) and the observer (3.32). Also, if the platform angular rate $\dot{\phi}(t)$ happened not to be available as a direct (sensor) measurement, one would replace $\dot{\phi}$ in (3.44) by the observer-produced estimate $\hat{\dot{\phi}} = \hat{x}_2$ in (3.32). In this way, the *practical implementation* of the platform controller (3.44) would take the form

$$u = T_c = -k_1 \phi - k_2 \hat{\dot{\phi}} - \hat{\tilde{T}}_d \; ; \; \begin{cases} \hat{\tilde{T}}_d = \ell_1 \text{ in (3.32)} \\ \hat{\dot{\phi}} = \ell_2 \text{ in (3.32)} \end{cases} \tag{3.45}$$

The estimate $\hat{\phi} = \hat{x}_1$ produced by (3.32) is not used in (3.45) since, presumably, $\phi(t)$ is directly measurable.

The control expression (law) (3.45) together with the composite state observer (3.32), (3.37), constitutes the complete disturbance-adaptive controller design for the platform controller T_c as shown in Figure 3.6. This controller will automatically adapt-to and quickly *cancel-out* any platform disturbance actions $\tilde{T}_d(t)$ that can be represented, at least over short intervals of time, by the *quadratic-spline* waveform-model (3.25). This includes disturbances that are: "stepwise-constant," uncertain combinations of "constants + ramps + accelerations," and general, uncertain, meandering-type functions, such as shown in Figure 3.5. The latter

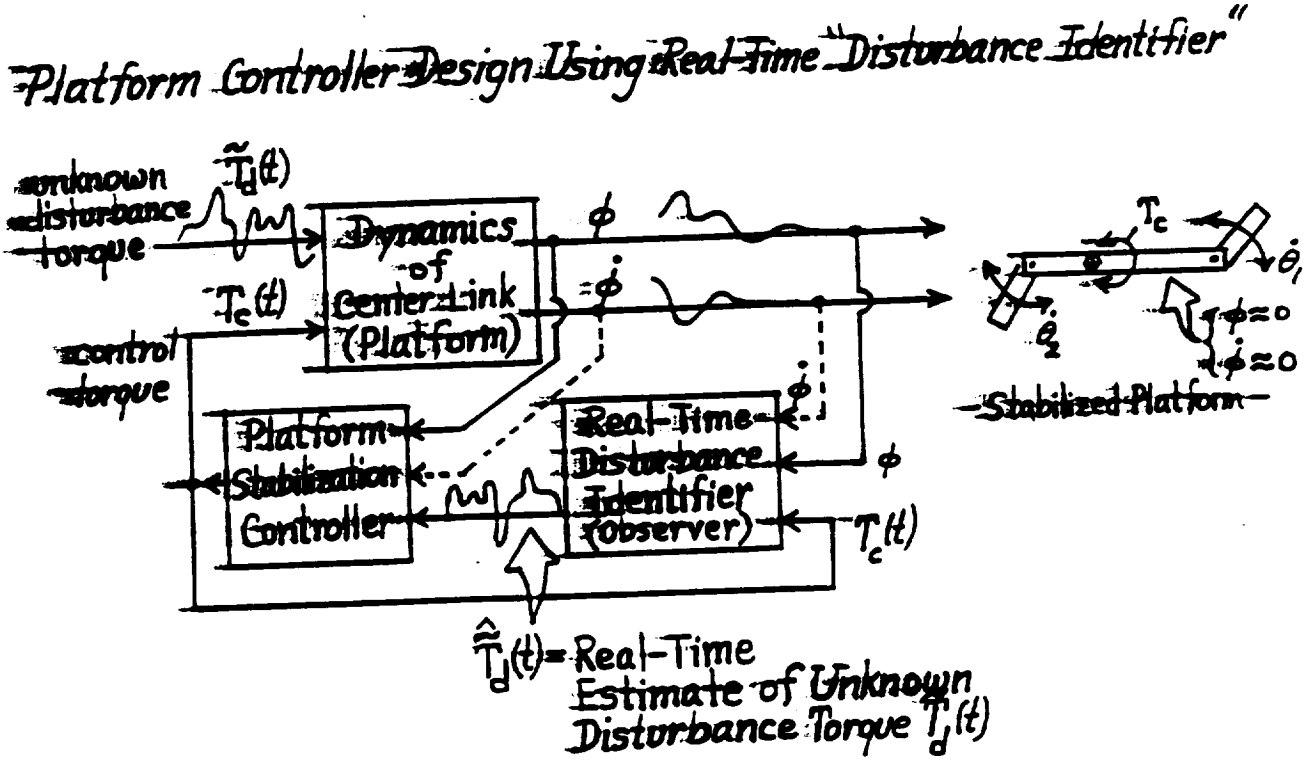


Figure 3.6 Proposed Disturbance-Adaptive Platform Controller Arrangement.

category includes slowly varying periodic and exponential- type disturbances with unknown frequencies, time-constants, etc.

3.7. Performance Testing of the Disturbance-Adaptive Platform Controller Using a Surrogate Disturbance Simulation Model

The ultimate simulation test of the proposed platform controller (3.45) , (3.32) , (3.37) consists of implementing the equations for that controller on a simulation of the complete, exact, equations of motion of the entire 3-link concept model shown in Figure 2.1, and then exercising that closed-loop simulation by controlling the end-links, via $T_1(t)$ and $T_2(t)$, to perform back-and-forth slewing motions to emulate typical "experiment motion" disturbances. Under such a test scenario, the platform angle $\phi(t)$ should promptly approach zero, $\phi(t) \rightarrow 0$, and consistently *remain at or near* zero in the face of all experiment motions, gravity-gradient torques, solar-pressure torques, etc. The results of such a "full-up" test will be described in Chapter 5.

In this section, we will present the results of a somewhat lower-fidelity simulation test of the platform controller, in which the platform is correctly modeled by (3.24b) but where the actual disturbances $\bar{T}_d(t)$ due to end-link motions, etc., are *synthetically generated* on the computer by a "function-generation" sub-routine that produces artificial $\tilde{T}_d(t)$ disturbance functions. This procedure avoids the (difficult) derivation and programming of the *enormously complicated* exact equations of motion of the 3-link assembly shown in Figure 2.1.

As previously stated, the derivation and simulation of those exact equations of motion will be addressed in Chapter 4.

To demonstrate the validity of the preceding real-time disturbance identification/cancellation technique, the mathematical model (3.24b) with the somewhat arbitrary value $\tilde{J}_{cg} = 10$ was simulated on a digital computer, using a digital simulation program called *Dynasim*[•]. The "disturbance torque" $\tilde{T}_d(t)$ in (3.24) was created by combining various standard mathematical time-functions (steps, ramps, sinusoids, delays, etc.) available in the *Dynasim* program. The disturbance-adaptive control torque $T_c(t)$ in (3.24) was generated in the simulation by equation (3.45) where the real-time estimates $\hat{\dot{\phi}}(t)$ and $\hat{\tilde{T}}_d(t)$ were created by the real-time composite state-observer (3.32), (3.37) using the desired observer eigenvalues

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = -3 \tag{3.46}$$

The observer gains $\{k_{oi}\}$ in (3.32) corresponding to (3.46) were computed from (3.37) to be

$$\begin{aligned} k_{01} &= -15 \quad ; \quad k_{02} = -90 \\ k_{03} &= -2700 \quad ; \quad k_{04} = -4050 \\ k_{05} &= -2430 \end{aligned} \tag{3.47}$$

The ideal-model (3.43) of the closed-loop platform dynamics was chosen to have

$$\zeta = 0.7 \; ; \; \omega_n = 1.0 \tag{3.48}$$

Some representative simulation results are shown plotted in Figure 3.7, where it can be seen that, after a short transient, the estimate $\hat{\tilde{T}}_d(t)$ does indeed accurately match the actual real-time behavior of $\tilde{T}_d(t)$. In a realistic application, one would activate the

disturbance observer a short time *before* activating the control u_d to allow time for the estimate $\hat{\tilde{T}}_d(t)$ to accurately match $\tilde{T}_d(t)$, and thereby avoid the undesirable effects of the (sometimes large) transient start-up errors in $\hat{\tilde{T}}_d(t)$. Moreover, the platform controller $T_c(t)$ does indeed regulate $\phi(t)$ to the desired condition $\{\phi(t) \to 0, \ \dot{\phi}(t) \to 0\}$ in the face of initial-conditions $\phi(0)$ and uncertain disturbances $\tilde{T}_d(t)$. These results serve to validate the correct functioning of the algorithms comprising the disturbance-observer (3.32) and the controller (3.45). One shortcoming of using *synthetically generated*, end-link "disturbances" $\tilde{T}_d(t)$, as employed here, is that such disturbances *do not* embody the inevitable *reactions* of, say, the T_1-controller to the control actions of the T_2-controller, and vice versa. Those controller-reaction components of $\tilde{T}_d(t)$ are an inescapable (and potentially de-stabilizing) reality of any real MPMSP dynamic behavior and must be accurately incorporated into any simulation that purports to demonstrate overall MPMSP dynamic behavior, including stability, in a realistic operating scenario. This concern will be addressed in Chapter 5.

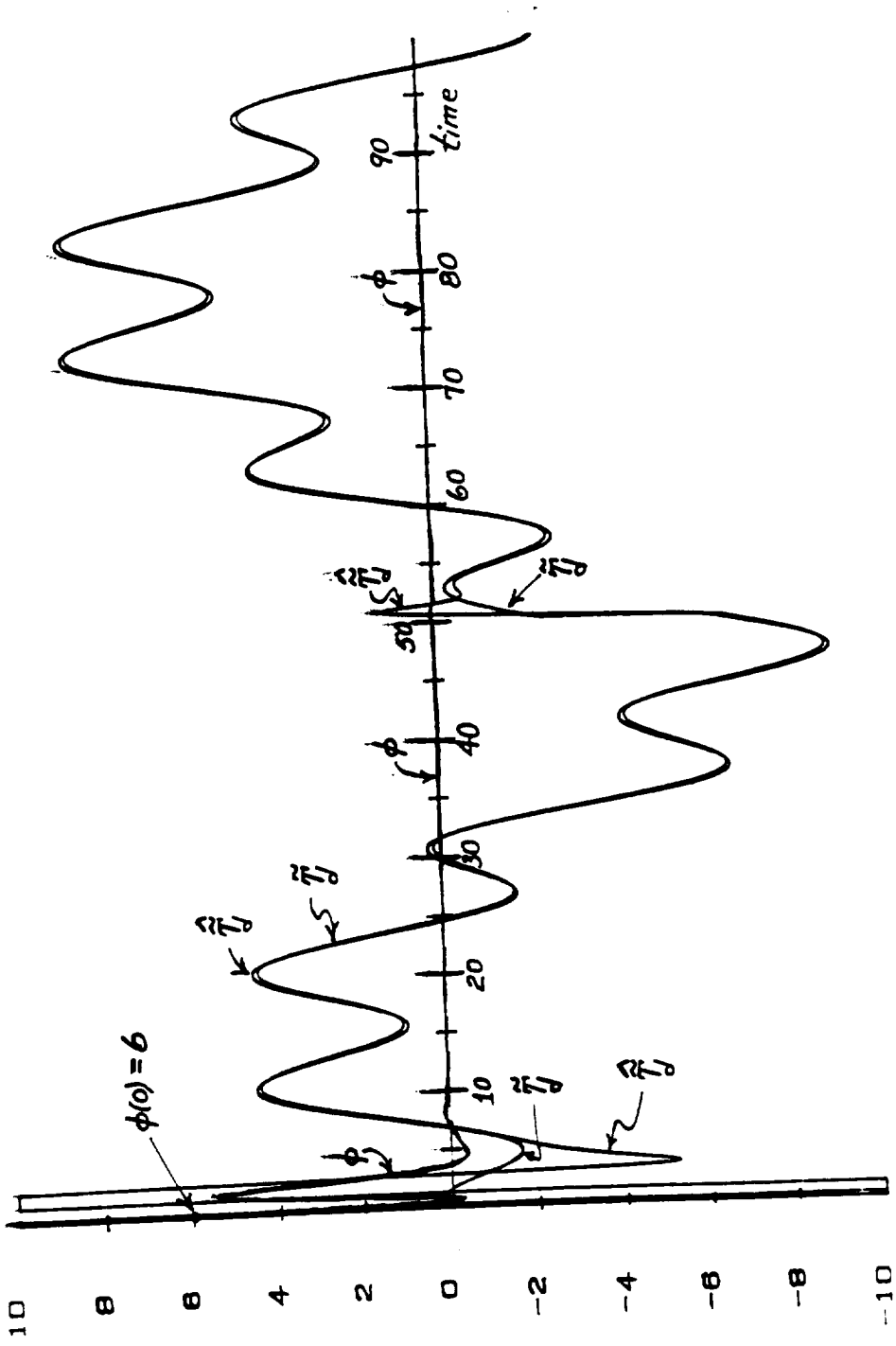


Figure 3.7 Representative Closed-Loop Simulation Results for Plant (3.24b) Using a Platform Controller (3.45) (3.32)-(3.37) and Synthetically Generated Disturbance $T_d(t)$.

Chapter 4

AN EXACT MATHEMATICAL MODEL OF THE GENERIC MPMSP IN PLANAR-MOTION

4.1 The Curse of Dimensionality in Modeling Multi-Body Dynamic Systems

The derivation of the exact equation(s) of motion for an interconnected set of rigid-bodies (hereafter referred to as a *multi-body* dynamic system) would seem, at first glance, to be a straight forward application of the classical Newtonian, Lagrangian, Hamiltonian, etc. principles of dynamics as presenting in literally hundreds of textbooks, and convincingly taught in university classrooms throughout the world over the past century. However, in spite of the intellectual profoundness of those classical principles of dynamics, and an abundance of (deceptively simple) illustrative examples worked out in textbooks and classroom lectures, it turns out that the methods of Newton, Lagrange, Hamilton, etc. are *inherently inadequate* for deriving the equations of motion for all but the simplest cases of multi-body systems.

This "glass ceiling" feature of the classical methods of dynamics has its origin in the "curse of dimensionality" associated with the development of analytical, symbolic solutions to certain sets of simultaneous equations that naturally arise as a necessary intermediate step in applying those classical methods to a high-order multi-body system. For instance, in applying the Newtonian method and "free-body diagrams" to a multi-body system, one must first solve analytically for the equations that define the reaction forces and moments that

occur at the points of interconnection among the set of interconnected bodies---for arbitrary kinematic and kinetic conditions that may occur during general motions of the multi-body system. This step is relatively straightforward for, say, two rigid bodies pin-jointed together (such as the *double-pendulum example* presented in many texts). However, as one considers an increasing number of such rigid bodies linked together by pin-joints, the step of analytically computing the reaction forces and moments at each pin-joint rapidly becomes mathematically intractable. Similar analytical obstacles are encountered in applying the methods of Lagrange, Hamilton, etc. to multi-body systems of higher order.

This inherent limitation of the classical methods of dynamics for deriving equations of motion for multi-body systems has only recently begun to be recognized by industry practitioners, educators, and (a few) textbook writers.

4.2 Kane's Methods for Modeling Multi-Body Systems

Further progress in the analysis and control of multi-body systems is inextricably linked to progress in overcoming the aforementioned fundamental limitation of the classical methods of dynamics. Fortunately, a new method for developing dynamic equations of motion, which effectively overcomes the aforementioned limitations of classical methods, has been discovered by Prof. Thomas Kane of Stanford University. This method, hereafter referred to a Kane's Method, was, in fact, developed and first published in the late 1960's [5], but has become widely recognized as the fundamental contribution that it really is, only in the last decade, [6] - [7].

The basic theory and methodology for Kane's method of dynamic modeling is developed, and illustrated by numerous examples, in a recent text [8]. The reader is referred to that text for further technical details. The essential feature of Kane's method, from the practitioner's point-of-view, is that regardless of the number of bodies in a multi-body system, the *exact* equations of motion can be developed by a well-defined, tractable, systematic procedure involving only the simplest concepts from an introductory course in 3-dimensional dynamics.

4.3 The *Autolev* Computer Program for Automated Modeling of Multi-Body Systems Using Kane's Method

The striking simplicity of Kane's methodology for deriving the exact equations of motion for a complex, multi-body system has inspired David Levinson and his associates to develop a fully automated digital computer program that automatically executes Kane's methodology, and prints out the final equations of motion, for an "arbitrarily" given, complex, multi-body system with complex interconnections . This program, called *Autolev*® is now commercially available[1] and was used in this study to derive, in minutes, the horribly complicated exact equations of (planar) motion for the generic 3-link model of an MPMSP, as shown in Figure 2.1. An attractive feature of the Autolev program is that it will, at the user's request, automatically convert the final equations of motion into Fortran code for

[1] OnLine Dynamics, Inc., 1605 Honfleur Dr., Sunnyvale, CA 94087; (408)-736-9566

subsequent automatic input into the user's existing simulation program, such as *Matrix-X*, etc. In this way, *Autolev* allows the user to go from the initial manually input "description" of the multi-body configuration to be modeled, to a working "simulation" of the multi-body system on a digital computer, in minutes, with no intervening printouts, recopying, paperwork, coding or translating. This complete elimination of virtually all human chores in the modeling and computer simulation of complex multi-body systems represents a major contribution to the industrial community, but clearly places strong requirements on the qualitative correctness and quantitative accuracy of the user's initial input "description" of the multi-body configuration to be studied. Indeed, using *Autolev* one can go from the description of the multi-body system to computer plots of the system dynamic response without ever seeing the system's equations of motion!

4.4 *Autolev* Print-Out of the Exact Equations of Motion for the Generic MPMSP in Planar Motion

The generic MPMSP 3-link configuration shown in Figures 2.1 and 3.4 was manually input into the *Autolev* program by describing, *symbolically*, the geometry $\{\phi, \theta_1, \theta_2, x, y\}$ of the overall link-arrangement, the locations $\{\ell_{oi}, \ell_i\}$ of the centers of gravity of the respective links, the inertia and mass values (J_i, M_i), the experiment control torques (T_i) at each pin-joint, and the platform control and external disturbance torques (T_c, T_e). Using this symbolic description of the multi-body system, the *Autolev* program executed Kane's Method to arrive at the exact equations of motion for the planar-motion case. Those

equations of motion consist of a set of five, 2nd order, inertia-coupled ordinary differential equations which *Autolev* prints-out in the form (all the terms shown on the right side of (4.1) are printed-out on the left side in *Autolev*).

$$z_{27}\ddot{\phi} - z_{28}\ddot{\theta}_1 - z_{29}\ddot{\theta}_2 + z_{30}\ddot{\zeta} + z_{31}\ddot{\eta} = z_{32} - T_c - T_e \quad (4.1a)$$

$$-z_{28}\ddot{\phi} - z_{33}\ddot{\theta}_1 + 0 + z_{34}\ddot{\zeta} - z_{35}\ddot{\eta} = -z_{36} - T_1 \quad (4.1b)$$

$$-z_{29}\ddot{\phi} + 0 - z_{37}\ddot{\theta}_2 - z_{38}\ddot{\zeta} + z_{39}\ddot{\eta} = -z_{40} - T_2 \quad (4,1c)$$

$$z_{30}\ddot{\phi} + z_{34}\ddot{\theta}_1 - z_{38}\ddot{\theta}_2 - z_{41}\ddot{\zeta} + 0 = z_{42} \quad (4.1d)$$

$$z_{31}\ddot{\phi} - z_{35}\ddot{\theta}_1 + z_{39}\ddot{\theta}_2 + 0 - z_{41}\ddot{\eta} = z_{43} \quad (4.1e)$$

where ζ and η denote, respectively, the absolute displacement of the c.g. of the platform in the direction of the platform longitudinal axis L and in the direction N which is normal to L (see Figure 3.4). The z_{ij} symbols in (4.1) represent (typically long, complicated, non-linear) functions of $\{\phi,\theta_1,\theta_2, \dot{\phi},\dot{\theta}_1,\dot{\theta}_2,M_0, M_2, J_0, J_1, J_2, \ell_{01}, \ell_{02}, \ell_1, \ell_2\}$. In particular, the term z_{27} in (4.1a) is the *augmented* inertia function as cited in (3.24b)

$$\tilde{J}_{cg} = z_{27} \quad (4.2)$$

The actual, raw, hard-copy print-outs of (4.1) as generated *Autolev*, for which the definitions of z_{27} in (4.2), and the other z_{ij} in (4.1) can be inferred, are reproduced in Figure 4.1 where $u1=\phi$; $u2 = \dot{\theta}_1$; $u3 = \dot{\theta}_2$; $u4 = \dot{\zeta}$; $u5 = \dot{\eta}$. Additional *Autolev* print-out data relevant to the model (4.1) is reproduced in Appendix A of this report.

$\ddot{\phi}$ -> (145) R = 1: Z27*U1'-Z28*U2'-Z29*U3'+Z *U4'+Z31*U5'-Z32+W = 0

T_c+T_e

$\ddot{\theta}_1$ -> (146) R = 2: -Z28*U1'-Z33*U2'+Z34*U4'-Z35*U5'+Z36+W1 = 0

T_1

$\ddot{\theta}_2$ -> (147) R = 3: -Z29*U1'-Z37*U3'-Z38*U4'+Z39*U5'+Z40+W2 = 0

T_2

$\ddot{\xi}$ -> (148) R = 4: Z30*U1'+Z34*U2'-Z38*U3'-Z41*U4'-Z42 = 0

$\ddot{\eta}$ -> (149) R = 5: Z31*U1'-Z35*U2'+Z39*U3'-Z41*U5'-Z43 = 0

$\ddot{\phi}$ -> (101) R = 1: ((-(-COS(THETA2)*L02-L2)*(-COS(THETA2)*L02-L2)-L
02*L02*SIN(THETA2)*SIN(THETA2))*M2+(-(COS(THETA1)*L01+L1)
*(COS(THETA1)*L01+L1)-L01*L01*SIN(THETA1)*SIN(THETA1))*M1
-J0-J1-J2)*U1'+(-(COS(THETA1)*L01+L1)*L1*M1-J1)*U2'+((-CO
S(THETA2)*L02-L2)*L2*M2-J2)*U3'+(((-COS(THETA2)*L02-L2)*S
IN(THETA2)+COS(THETA2)*L02*SIN(THETA2))*M2+((COS(THETA1)*
L01+L1)*SIN(THETA1)-COS(THETA1)*L01*SIN(THETA1))*M1)*U4'+
((-(-COS(THETA2)*L02-L2)*COS(THETA2)+L02*SIN(THETA2)*SIN(
THETA2))*M2+(-(COS(THETA1)*L01+L1)*COS(THETA1)-L01*SIN(TH
ETA1)*SIN(THETA1))*M1)*U5'-(((-U1+U2)*L01*SIN(THETA1)+COS
(THETA1)*U4-L01*SIN(THETA1)*U2+SIN(THETA1)*U5)*U1+(COS(TH
ETA1)*U4+SIN(THETA1)*U5)*U2-COS(THETA1)*U2*U4-SIN(THETA1)
*U2*U5)*(COS(THETA1)*L01+L1)+((-(COS(THETA1)*L01+L1)*(U1+
U2)+COS(THETA1)*L01*U2-COS(THETA1)*U5-L1*U2+SIN(THETA1)*U
4)*U1-L1*U2*U2)*L01*SIN(THETA1))*M1-(-((-(-COS(THETA2)*L0
2-L2)*(U1+U3)-COS(THETA2)*L02*U3-COS(THETA2)*U5+L2*U3+SIN
(THETA2)*U4)*U1+L2*U3*U3)*L02*SIN(THETA2)+((-(U1+U3)*L02*
SIN(THETA2)+COS(THETA2)*U4+L02*SIN(THETA2)*U3+SIN(THETA2)
*U5)*U1+(COS(THETA2)*U4+SIN(THETA2)*U5)*U3-COS(THETA2)*U3
*U4-SIN(THETA2)*U3*U5)*(-COS(THETA2)*L02-L2))*M2+W = 0

Figure 4.1 Actual Print-Out of Exact Equations of (Planar) Motion for 3-Link Generic MPMSP Model, Fig. 2.1. as Automatically Derived by *Autolev*. Program.

$\ddot{\theta}_1$ -> (102) R = 2: (-(COS(THETA1)*LO1+L1)*L1*M1-J1)*U1'+(-J1-L1*L1*
M1)*U2'+L1*M1*SIN(THETA1)*U4'-COS(THETA1)*L1*M1*U5'-(((U1
+U2)*LO1*SIN(THETA1)+COS(THETA1)*U4-LO1*SIN(THETA1)*U2+SI
N(THETA1)*U5)*U1+(COS(THETA1)*U4+SIN(THETA1)*U5)*U2-COS(T
HETA1)*U2*U4-SIN(THETA1)*U2*U5)*L1*M1+W1 = 0

$\ddot{\theta}_2$ -> (103) R = 3: ((-COS(THETA2)*LO2-L2)*L2*M2-J2)*U1'+(-J2-L2*L2*
M2)*U3'-L2*M2*SIN(THETA2)*U4'+COS(THETA2)*L2*M2*U5'+((-(U
1+U3)*LO2*SIN(THETA2)+COS(THETA2)*U4+LO2*SIN(THETA2)*U3+S
IN(THETA2)*U5)*U1+(COS(THETA2)*U4+SIN(THETA2)*U5)*U3-COS(
THETA2)*U3*U4-SIN(THETA2)*U3*U5)*L2*M2+W2 = 0

Figure 4.1 Actual Print-Out of Exact Equations of (Planar) Motion for 3-Link Generic MPMSP Model, Fig. 2.1, as Automatically Derived by *Autolev*, Program.

```
-> (104) R = 4:    (((-(    (THETA2)*L02-L2)*SIN(THE    )+COS(THETA2)*
L02*SIN(THETA2))*M2+((COS(THETA1)*L01+L1)*SIN(THETA1)-COS
(THETA1)*L01*SIN(THETA1))*M1)*U1'+L1*M1*SIN(THETA1)*U2'-L
2*M2*SIN(THETA2)*U3'+(-M0-M1-M2)*U4'-(((-(-COS(THETA2)*L0
2-L2)*(U1+U3)-COS(THETA2)*L02*U3-COS(THETA2)*U5+L2*U3+SIN
(THETA2)*U4)*U1+L2*U3*U3)*COS(THETA2)-((-(U1+U3)*L02*SIN(
THETA2)+COS(THETA2)*U4+L02*SIN(THETA2)*U3+SIN(THETA2)*U5)
*U1+(COS(THETA2)*U4+SIN(THETA2)*U5)*U3-COS(THETA2)*U3*U4-
SIN(THETA2)*U3*U5)*SIN(THETA2))*M2-(-(((U1+U2)*L01*SIN(TH
ETA1)+COS(THETA1)*U4-L01*SIN(THETA1)*U2+SIN(THETA1)*U5)*U
1+(COS(THETA1)*U4+SIN(THETA1)*U5)*U2-COS(THETA1)*U2*U4-SI
N(THETA1)*U2*U5)*SIN(THETA1)+((-(COS(THETA1)*L01+L1)*(U1+
U2)+COS(THETA1)*L01*U2-COS(THETA1)*U5-L1*U2+SIN(THETA1)*U
4)*U1-L1*U2*U2)*COS(THETA1))*M1+M0*U1*U5 = 0

-> (105) R = 5:    ((-(-COS(THETA2)*L02-L2)*COS(THETA2)+L02*SIN(THE
TA2)*SIN(THETA2))*M2+(-(COS(THETA1)*L01+L1)*COS(THETA1)-L
01*SIN(THETA1)*SIN(THETA1))*M1)*U1'+COS(THETA1)*L1*M1*U2'
+COS(THETA2)*L2*M2*U3'+(-M0-M1-M2)*U5'-(((U1+U2)*L01*SIN
(THETA1)+COS(THETA1)*U4-L01*SIN(THETA1)*U2+SIN(THETA1)*U5
)*U1+(COS(THETA1)*U4+SIN(THETA1)*U5)*U2-COS(THETA1)*U2*U4
-SIN(THETA1)*U2*U5)*COS(THETA1)+((-(COS(THETA1)*L01+L1)*(
U1+U2)+COS(THETA1)*L01*U2-COS(THETA1)*U5-L1*U2+SIN(THETA1
)*U4)*U1-L1*U2*U2)*SIN(THETA1))*M1-(((-(-COS(THETA2)*L02-
L2)*(U1+U3)-COS(THETA2)*L02*U3-COS(THETA2)*U5+L2*U3+SIN(T
HETA2)*U4)*U1+L2*U3*U3)*SIN(THETA2)+((-(U1+U3)*L02*SIN(TH
ETA2)+COS(THETA2)*U4+L02*SIN(THETA2)*U3+SIN(THETA2)*U5)*U
1+(COS(THETA2)*U4+SIN(THETA2)*U5)*U3-COS(THETA2)*U3*U4-SI
N(THETA2)*U3*U5)*COS(THETA2))*M2-M0*U1*U4 = 0
```

Figure 4.1 Actual Print-Out of Exact Equations of (Planar) Motion for 3-Link Generic MPMSP Model. Fig. 2.1. as Automatically Derived by *Autolev*. Program.

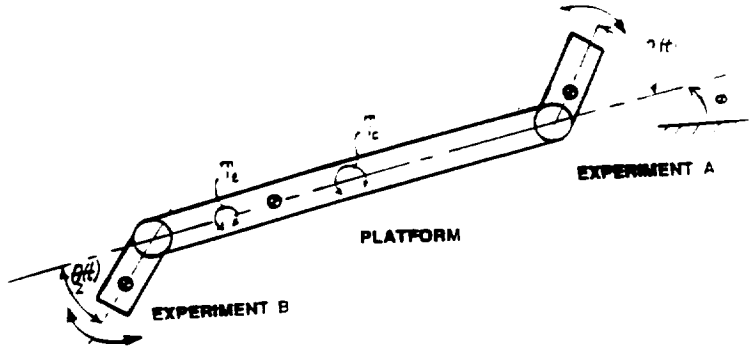
4.5 Validation of the *Autolev*-Generated Equations of Motion for the 3-Link MPMSP Generic Model

As indicated in Section 4.3, the *Autolev* program will, at the user's request, automatically convert the model (4.1) into Fortran code that can then be immediately input into an existing "dynamic-system simulation program," such as *Matrix-X*, etc. This was done for the present study (using *Matrix-X* installed on a UAH 486/33 PC). A listing of that *Autolev*-generated Fortran code is shown in Appendix A. The resulting *Matrix-X* simulation of the 3-link MPMSP model was "exercised" for some representative end-link, back-and-forth slewing motions (controlled via T_1, T_2) with the platform controller T_c and external torques T_e set to zero. The results obtained, using the parameter-values listed in Table 5.1, are shown in Figures 4.2, 4.3, and their seeming agreement with what one would intuitively expect served to establish our confidence in the validity of both the mathematical model (4.1) and the *Matrix-X* simulation implementation, via the Fortran code generated by *Autolev*.

An innovative and useful feature of the *Autolev*-generated Fortran simulation code is that it automatically computes and prints out the time-variation of the total linear momentum and total angular momentum of the multi-body system being simulated. This feature allows the user to easily verify that the simulated multi-body system does, indeed, obey known *momentum-conservation conditions*, when they apply. In our validation runs described above, where $T_c(t) \equiv T_e(t) \equiv 0$, the torques $T_1(t)$, $T_2(t)$ controlling the end-link motions were "internal torques" with respect to the system and, therefore, should not have

altered the system's total angular momentum. That conservation of total angular momentum of the overall system was indeed evidenced in the simulation print-out of instantaneous total angular momentum values vs. time t as shown in the typical result in Table 4.1 and Figure 4.4.

In the next chapter, the "exact" *Autolev*-based computer simulation of the 3-link generic MPMSP model described in this section, is used as a test-bed to verify the effectiveness of the previously designed DAC platform controller (3.45), (3.32), (3.37) in keeping the platform quiet in the face of a general class of controlled end-link motions $\theta_1(t)$, $\theta_2(t)$ and some representative external disturbance torques $T_e(t)$.



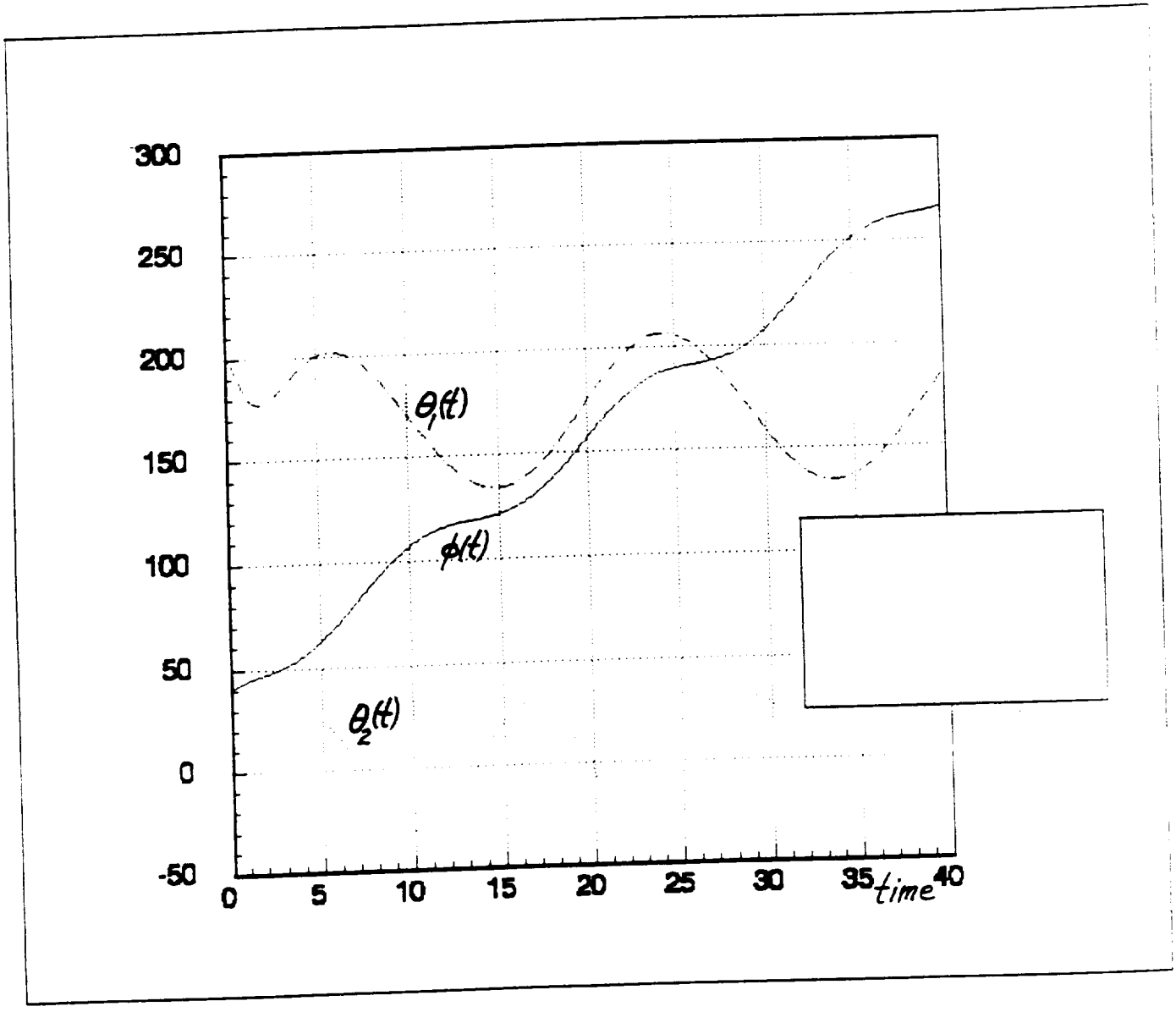
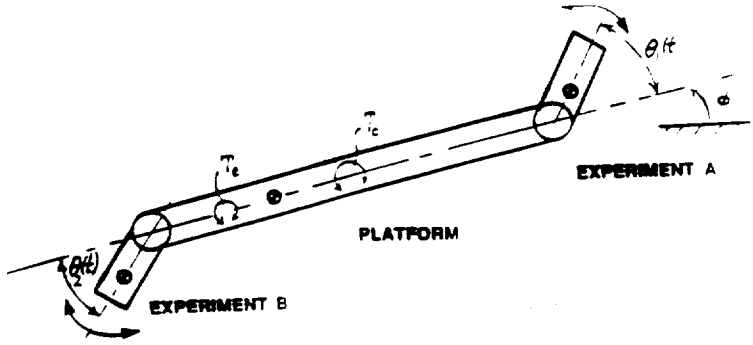


Figure 4.2 Representative Plots from Simulation Validation Tests of the Model (4.1) with $T_c(t) \equiv T_e(t) \equiv 0$; $T_i(t)$ = sinusoidal at freq. ω_i, i = 1,2 (Case: ω_1 = 0.33; ω_2 = 0.5.)



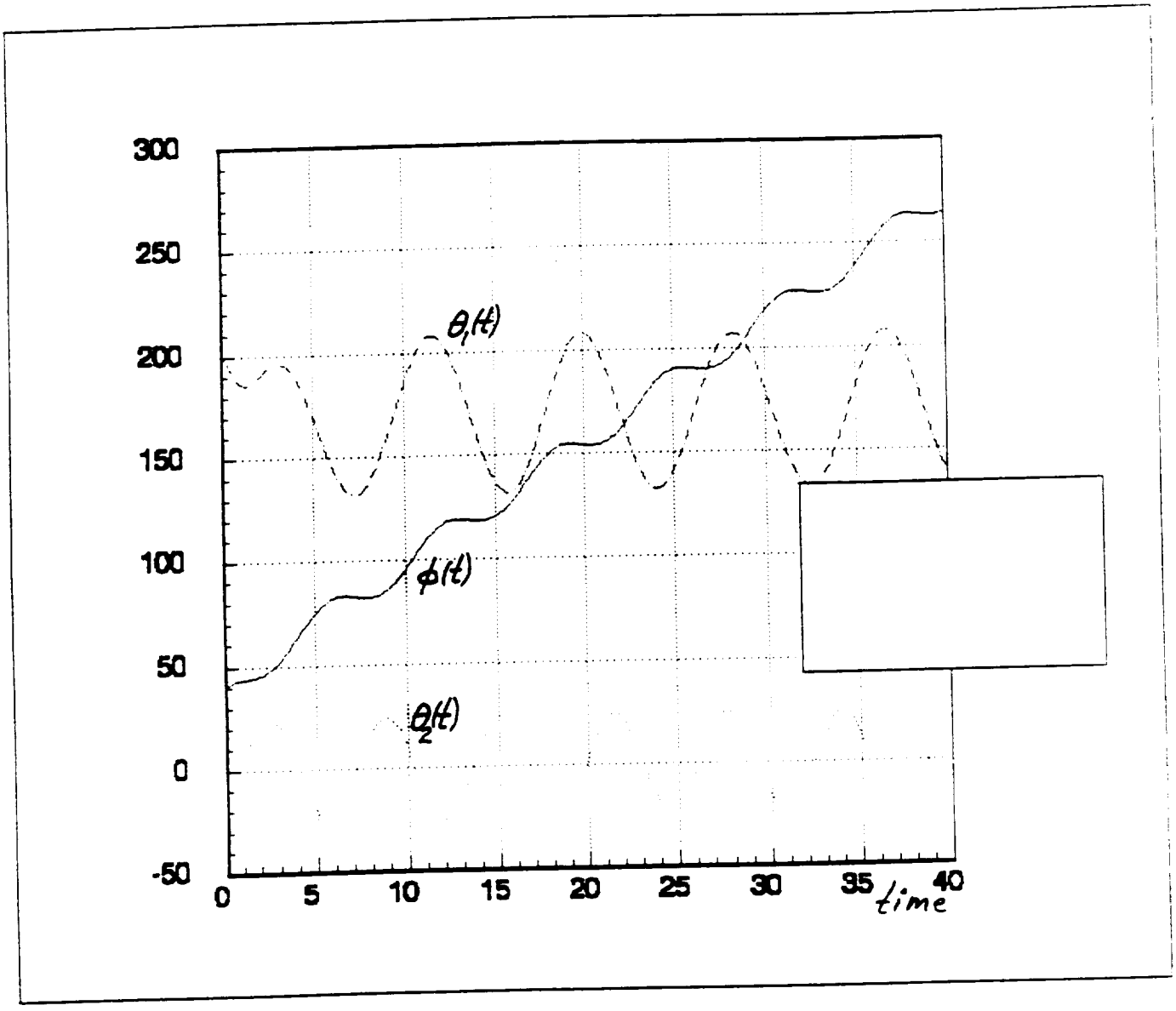


Figure 4.3 Representative Plots from Simulation Validation Tests of the Model (4.1) with $T_c(t) \equiv T_e(t) \equiv 0$; $T_i(t)$ = sinusoidal at freq. ω_i, i = 1,2 (Case: ω_1 = 0.75; ω_2 = 1.0.)

```
      SUBROUTINE ANGMOM(T,U   ,1,HN2,HN3,HN
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIMENSION U(17)
      COMMON/CZEES/Z(45)
      COMMON/CPAR/J0,J1,J2,M0,M1,M2,PI,DEGTORAD,RADTODEG,L01,L02,L1,L2,
     &  L2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,OMEGA1,CENTER2,AMP2,OMEGA2

      PHI = U(6)
      THETA1 = U(7)
      THETA2 = U(8)
      PHIABSINT = U(9)
      PHIINT1 = U(10)
      THE1INT = U(11)
      THE2INT = U(12)
      X01 = U(13)
      X02 = U(14)
      Z01 = U(15)
      Z02 = U(16)
      Z03 = U(17)


      S1 = DSIN(THETA1)
      C1 = DCOS(THETA1)
      S2 = DSIN(THETA2)
      C2 = DCOS(THETA2)

      ZH1 = U(4)*Z(45)-U(5)*Z(44)
      ZH2 = C1*L1+L01-Z(44)
      ZH3 = L1*S1-Z(45)
      ZH4 = C1*Z(11)-S1*Z(12)
      ZH5 = C1*Z(12)+S1*Z(11)
      ZH6 = ZH2*ZH5-ZH3*ZH4
      ZH7 = -C2*L2-L02-Z(44)
      ZH8 = -L2*S2-Z(45)
      ZH9 = C2*Z(20)-S2*Z(21)
      ZH10 = C2*Z(21)+S2*Z(20)
      ZH11 = ZH10*ZH7-ZH8*ZH9
      HN1 = 0.0
      HN2 = 0.0
      HN3 = J0*U(1)+J1*Z(13)+J2*Z(22)+M0*ZH1+M1*ZH6+M2*ZH11
      HN  = DSQRT(HN1*HN1 + HN2*HN2 + HN3*HN3)

      RETURN
      END
```

Table 4.1

Autolev Set-Up for Plotting Instantaneous Total Angular Momentum Values vs. Time for a Typical Simulation Validation Run: $T_c \equiv T_e \equiv 0$; $T_1, T_2 \neq 0$

[Table Continued Next 3 Pages]

*** SIMULATION RUN WITH Tc=0. END LINK COMMAND FREQ.=0.33, 0.5 rad/sec
ALL INITIAL ANGULAR VELOCITIES SET TO ZERO

SIMULATION RESULTS *(Validation Run #1)*

T (S)	HN1 (N*M*S)	HN2 (N*M*S)	HN3 (N*M*S)	*Total Angular Momentum* HN (N*M*S)
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5.00000E-01	0.00000E+00	0.00000E+00	-5.65480E-11	5.65480E-11
1.00000E+00	0.00000E+00	0.00000E+00	-5.13900E-11	5.13900E-11
1.50000E+00	0.00000E+00	0.00000E+00	-5.08386E-11	5.08386E-11
2.00000E+00	0.00000E+00	0.00000E+00	-5.16865E-11	5.16865E-11
2.50000E+00	0.00000E+00	0.00000E+00	-5.15269E-11	5.15269E-11
3.00000E+00	0.00000E+00	0.00000E+00	-5.13147E-11	5.13147E-11
3.50000E+00	0.00000E+00	0.00000E+00	-5.13134E-11	5.13134E-11
4.00000E+00	0.00000E+00	0.00000E+00	-5.14228E-11	5.14228E-11
4.50000E+00	0.00000E+00	0.00000E+00	-5.15336E-11	5.15336E-11
5.00000E+00	0.00000E+00	0.00000E+00	-5.15838E-11	5.15838E-11
5.50000E+00	0.00000E+00	0.00000E+00	-5.15573E-11	5.15573E-11
6.00000E+00	0.00000E+00	0.00000E+00	-5.14896E-11	5.14896E-11
6.50000E+00	0.00000E+00	0.00000E+00	-5.14017E-11	5.14017E-11
7.00000E+00	0.00000E+00	0.00000E+00	-5.13689E-11	5.13689E-11
7.50000E+00	0.00000E+00	0.00000E+00	-5.13518E-11	5.13518E-11
8.00000E+00	0.00000E+00	0.00000E+00	-5.14125E-11	5.14125E-11
8.50000E+00	0.00000E+00	0.00000E+00	-5.14827E-11	5.14827E-11
9.00000E+00	0.00000E+00	0.00000E+00	-5.15638E-11	5.15638E-11
9.50000E+00	0.00000E+00	0.00000E+00	-5.15906E-11	5.15906E-11
1.00000E+01	0.00000E+00	0.00000E+00	-5.15725E-11	5.15725E-11
1.05000E+01	0.00000E+00	0.00000E+00	-5.15130E-11	5.15130E-11
1.10000E+01	0.00000E+00	0.00000E+00	-5.14497E-11	5.14497E-11
1.15000E+01	0.00000E+00	0.00000E+00	-5.14316E-11	5.14316E-11
1.20000E+01	0.00000E+00	0.00000E+00	-5.14674E-11	5.14674E-11
1.25000E+01	0.00000E+00	0.00000E+00	-5.15362E-11	5.15362E-11
1.30000E+01	0.00000E+00	0.00000E+00	-5.15778E-11	5.15778E-11
1.35000E+01	0.00000E+00	0.00000E+00	-5.16166E-11	5.16166E-11
1.40000E+01	0.00000E+00	0.00000E+00	-5.15843E-11	5.15843E-11
1.45000E+01	0.00000E+00	0.00000E+00	-5.14958E-11	5.14958E-11
1.50000E+01	0.00000E+00	0.00000E+00	-5.14400E-11	5.14400E-11
1.55000E+01	0.00000E+00	0.00000E+00	-5.14303E-11	5.14303E-11
1.60000E+01	0.00000E+00	0.00000E+00	-5.14924E-11	5.14924E-11
1.65000E+01	0.00000E+00	0.00000E+00	-5.15889E-11	5.15889E-11
1.70000E+01	0.00000E+00	0.00000E+00	-5.16742E-11	5.16742E-11
1.75000E+01	0.00000E+00	0.00000E+00	-5.16924E-11	5.16924E-11
1.80000E+01	0.00000E+00	0.00000E+00	-5.16360E-11	5.16360E-11
1.85000E+01	0.00000E+00	0.00000E+00	-5.15061E-11	5.15061E-11
1.90000E+01	0.00000E+00	0.00000E+00	-5.14074E-11	5.14074E-11
1.95000E+01	0.00000E+00	0.00000E+00	-5.13447E-11	5.13447E-11
2.00000E+01	0.00000E+00	0.00000E+00	-5.13661E-11	5.13661E-11
2.05000E+01	0.00000E+00	0.00000E+00	-5.14390E-11	5.14390E-11
2.10000E+01	0.00000E+00	0.00000E+00	-5.15552E-11	5.15552E-11
2.15000E+01	0.00000E+00	0.00000E+00	-5.16568E-11	5.16568E-11
2.20000E+01	0.00000E+00	0.00000E+00	-5.16923E-11	5.16923E-11
2.25000E+01				5.16607E-11
2.30000E+01				5.15791E-11
2.35000E+01				5.14899E-11
2.40000E+01				5.14568E-11
2.45000E+01	0.00000E+00	0.00000E+00	-5.14904E-11	5.14904E-11
2.50000E+01	0.00000E+00	0.00000E+00	-5.15407E-11	5.15407E-11

Table 4.1

Instantaneous Total Angular Momentum Values vs. Time for a Typical Simulation Validation Run: $T_c \equiv T_e \equiv 0$; $T_1, T_2 \neq 0$

*** SIMULATION RUN WITH Tc=0. END LINK COMMAND FREQ.=0.75, 1.0 rad/sec
*** INITIAL ANGULAR VELOCITY OF CENTER LINK=0.1 rad/sec
*** ALL OTHER INITIAL VELOCITIES EQUAL 0.0

SIMULATION RESULTS (Validation Run #2)

Total Angular Momentum

T (S)	HN1 (N*M*S)	HN2 (N*M*S)	HN3 (N*M*S)	HN (N*M*S)
0.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
5.00000E-01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
3.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
3.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
4.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
4.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
5.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
5.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
6.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
6.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
7.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
7.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
8.00000E+00	0.00000E+00	0.00000E-00	2.28998E+01	2.28998E+01
8.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
9.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
9.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.00000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.05000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.10000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.15000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.20000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.25000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.30000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.35000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.40000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.45000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.50000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.55000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.60000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.65000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.70000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.75000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.80000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.85000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.90000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.95000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.00000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.05000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.10000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.15000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.20000E+01				2.28998E+01
2.25				2.28998E+01
2.30				2.28998E+01
2.35				2.28998E+01
2.40				2.28998E+01

ORIGINAL PAGE IS OF POOR QUALITY

Table 4.1

Instantaneous Total Angular Momentum Values vs. Time for a Typical Simulation Validation Run: $T_c \equiv T_e \equiv 0$: $T_1 . T_2 \neq 0$

FILE: DAC.F (OUTPUT F... PROGRAM DAC.FOR

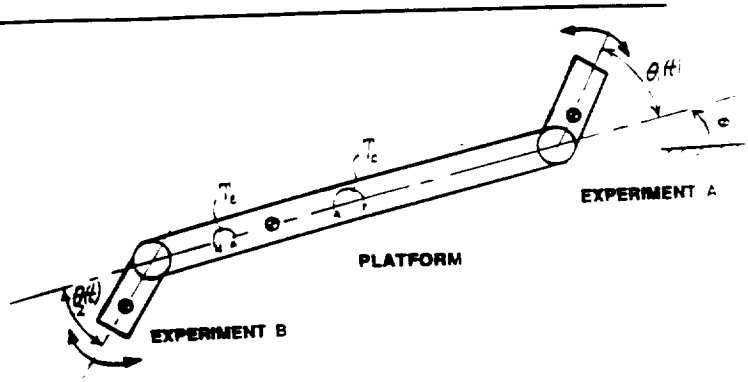
*** SIMULATION RUN WITH Tc=0. END LINK COMMAND FREQ.=0.00. 0.5 rad/sec
*** INITIAL ANGULAR VELOCITY OF CENTER LINK=0.1 rad/sec
*** ALL OTHER INITIAL VELOCITIES EQUAL 0.0

SIMULATION RESULTS (Validation Run #3)

T (S)	HN1 (N*M*S)	HN2 (N*M*S)	HN3 (N*M*S)	Total Angular Momentum HN (N*M*S)
0.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
5.00000E-01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
3.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
3.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
4.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
4.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
5.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
5.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
6.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
6.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
7.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
7.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
8.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
8.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
9.00000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
9.50000E+00	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.00000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.05000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.10000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.15000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.20000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.25000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.30000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.35000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.40000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.45000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.50000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.55000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.60000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.65000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.70000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.75000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.80000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.85000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.90000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
1.95000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.00000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.05000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.10000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.15000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.20000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01
2.25...	...	...	...	...
2.30...	...	...	...	...
2.35...	...	...	...	...
2.40000E+01	0.00000E+00	0.00000E+00	2.28998E+01	2.28998E+01

Table 4.1

Instantaneous Total Angular Momentum Values vs. Time for a Typical Simulation Validation Run: $T_c \equiv T_e \equiv 0$: $T_1, T_2 \neq 0$



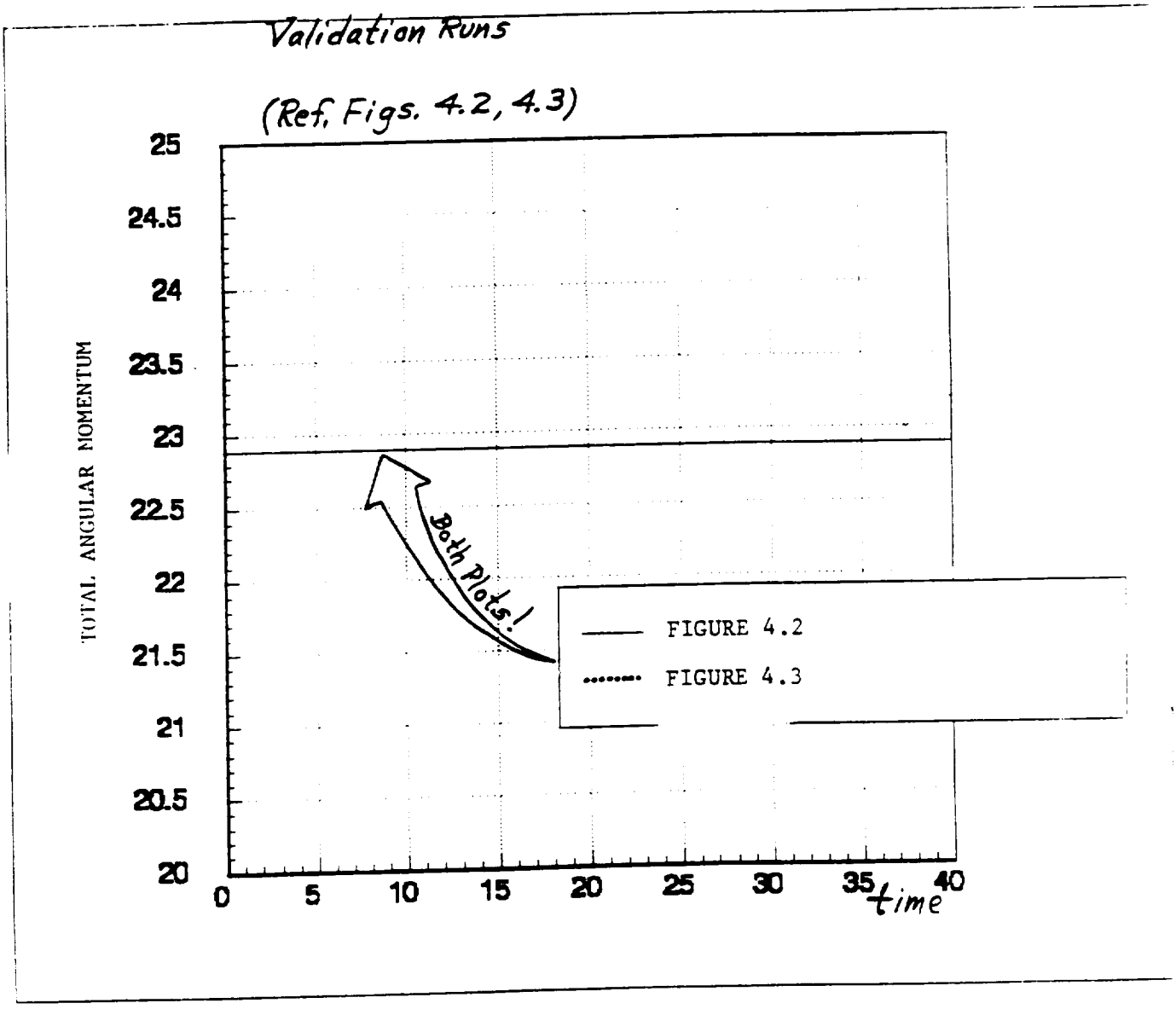


Figure 4.4 Time-Plots of Variation in Total Angular Momentum of 3-Link Generic MPMSP Model During Validation Tests in Figures 4.2 and 4.3.

Chapter 5

PERFORMANCE EVALUATION OF THE DISTURBANCE-ADAPTIVE PLATFORM CONTROLLER USING AN "EXACT" SIMULATION OF THE 3-LINK GENERIC MPMSP MODEL IN PLANAR MOTION

In this chapter we present the closed-loop simulation results obtained by exercising the "exact" dynamic model (4.1) of the 3-link generic MPMSP system using a simulation of the proposed disturbance-adaptive platform controller designed in Section 3.6 of Chapter 3. The results clearly show the effectiveness of the platform controller in identifying and adapting to the kind of complex, uncertain, time-varying disturbances that will be common on any realistic MPMSP. In addition, the simulation results in this chapter provide vivid evidence of the "controller-induced destabilization" phenomena described in Chapter 1; namely, that the individually stable, pointing/tracking controllers associated with each on-board experiment can, under certain conditions, begin "fighting" with each other, with the result that the entire MPMSP then undergoes a chaotic-like motion that can lead to instability of the entire MPMSP system.

5.1 Overview of the Performance Evaluation Test Procedure

The purpose of the simulation tests described herein is to demonstrate that the disturbance-adaptive platform controller (3.32), (3.37), (3.45) controlling the "exact" dynamic model (4.1) of the 3-link system of Figure 2.1 does indeed regulate the platform angular motion $\phi(t)$ to essentially zero, in the face of a general class of experiment pointing/slewing motions (end-link motions) $\theta_1(t)$, $\theta_2(t)$. For this purpose, the two experiment (torque)

controllers $T_1(\cdot)$, $T_2(\cdot)$ that control $\theta_1(t)$, $\theta_2(t)$ respectively, were designed as conventional Proportional, Integral, Derivative (PID) controllers, having the form

$$T_i = k_{i1}\left(\theta_{ic} - \theta_i\right) + k_{i2}\,\dot{\theta}_i + k_{i3}\int_{t_o}^{t}\left\{\theta_{ic}(\tau) - \theta_i(\tau)\right\} d\tau \qquad (5.1)$$

$$i = 1, 2$$

where the k_{ij} , j = 1, 2, 3 are constant controller gains chosen to make each $\theta_i(t)$ accurately track the given time-varying "command" motion $\theta_{ic}(t)$ - - assuming a quiet platform $\phi(t) \equiv 0$. The commanded scanning motions $\theta_{ic}(t)$ for the two experiments were assumed to have the biased-sinusoid form

$$\theta_{ic}(t) = \theta_{io} + \theta_{ia}\,\sin\left(\omega_i t\right) \;\; ; \; i = 1, 2 \qquad (5.2)$$

where the values of the parameters $\{\theta_{io}, \theta_{ia}, \omega_i\}$ were chosen to simulate generic experiment slewing commands.

The two experiments' controlled, but un-coordinated, slewing motions $\theta_1(t)$, $\theta_2(t)$ impart an uncertain, random-like "reaction torque" on the center-link (platform) in Figure 2.1. We will assume those reaction torques cannot be directly measured in real-time. In addition to the reaction-type sources of platform disturbances, a separate and more general type of uncertain, unmeasurable external disturbance torque $T_e(t)$ was considered to act on the platform, simulating the combined effects of uncertain gravity gradient torques, solar pressure torques from the solar panels, and any other external sources of torques that might act on an actual MPMSP. Accordingly, the generic waveform structure of $T_e(t)$ was assumed to be represented by

$$T_e(t) = C_1 + C_2 t + C_3 t^2 + C_4 \sin \omega_a t + C_5 \cos \omega_b t \qquad (5.3)$$

where ω_a ω_b are presumed known characteristic frequencies of $T_e(t)$ and the C_i, i = 1, 2, ... 5, are weighting "constants" that may occasionally jump in value in a once-in-a-while fashion as in Figure 3.1. Thus, (5.3) is a spline-function model of $T_e(t)$ that can emulate a rich variety of uncertain environmental torques that an actual MPMSP might experience on-station.

The effectiveness of the platform controller in regulating $\phi(t) \rightarrow 0$, and maintaining $\phi(t) \approx 0$ for an extended period of time, was tested by starting with $\phi(t_o) \neq 0$ and commanding $\theta_1(t)$, $\theta_2(t)$ to track the continually varying $\theta_{ic}(t)$ in (5.2) while the platform was subjected to a given persistent disturbance $T_e(t)$ in (5.3). In some runs, $T_e(t)$ was set to zero for comparison purposes.

The resulting time-variations of $\{\theta_1(t), \theta_2(t), \phi(t), T_c(t), T_e(t)\}$, obtained from the simulation, were then plotted.

5.2 Parameter Values Used in the Simulation Tests

As explained in Chapter 1, Section 1.4, the main focus of this research effort was to develop and demonstrate a new control concept that could form the technology basis for designing high-performance platform controllers for MPMSP-type projects. Thus, the numerical parameter-values selected for the simulation exercises were not chosen to represent any specific MPMSP, or specific experiments, that may be currently under consideration, but rather to represent a size/scale range that seemed reasonable for a generic MPMSP system configured in the form of the simplified 3-link model in Figure 2.1. Based

on these considerations the parameter-values for the 3-link model in Figures 2.1, 3.4 were chosen as shown in Table 5.1 below: (values shown are in non-dimensional units)

M_0	=	mass of platform (center link)	=	100.0
M_1	=	mass of right-link (Experiment #A)	=	5.0
M_2	=	mass of left-link (Experiment #B)	=	7.5
J_0	=	rotational moment of inertia of platform w.r.t.cg.	=	100.0
J_1	=	" " " " right-link w.r.t.cg.	=	5.0
J_2	=	" " " " left-link w.r.t.cg.	=	7.5
ℓ_{01}	=	(See Figure 3.4)	=	2.0
ℓ_{02}	=	(")	=	2.0
ℓ_1	=	(")	=	1.5
ℓ_2	=	(")	=	2.0

Table 5.1 Simulation Parameter Values for 3-Link MPMSP Model in Figures 2.1, 3.4 (Values are in non-dimensional units)

The gains k_{ij} for the PID experiment controllers (5.1) at each end link were chosen to yield closed-loop $\theta_i(t)$-dynamics having their 3 closed-loop poles (under the idealized assumption that $\phi(t) \equiv 0$) as follows:

Case 1: Both experiments have their 3 closed-loop poles set at

$$\lambda_1 = \lambda_2 = \lambda_3 = -1 \tag{5.4}$$

Case 2: Both experiments have their 3 closed-loop poles set at

$$\lambda_1 = \lambda_2 = \lambda_3 = -2.5 \tag{5.5}$$

The 2nd order ideal-model (3.43) for the closed-loop platform dynamics $\phi(t)$ was

chosen to have the parameter values

$$\xi = 0.89 \quad ; \quad \omega_n = 4.472 \tag{5.6}$$

which correspond to the two closed-loop poles

$$\lambda_{1,2} = -4.0 \pm j\, 2.0 \tag{5.7}$$

The gains k_{oi} for the composite plant/disturbance observer (3.32) were designed to place the five poles λ_{oi} of the associated estimation error dynamics (3.33) at the locations:

$$\lambda_{01} = \lambda_{02} = \ldots = \lambda_{05} = -5.0 \tag{5.8}$$

The specific sinusoidal scanning commands (5.2) for the end-links were chosen as follows:

For the Right-Link

$$\theta_{ic}(t) = 170 + 30.0 \sin (1.45\, t) \tag{5.9}$$

For the Left-Link

$$\theta_{2c}(t) = 0 + 25 \sin (1.60\, t) \tag{5.10}$$

Finally, the initial-conditions for the angles ϕ, θ_1, θ_2 in Figure 2.1 were chosen as

$$\begin{aligned} \phi\,(0) &= 30 \\ \theta_1(0) &= 200 \\ \theta_2(0) &= 0.0 \end{aligned} \tag{5.11}$$

Using the numerical parameter values in Table 5.1 and in (5.4) - (5.11), the *Autolev/Matrix-X* simulation was exercised for a run-time of 40 - 50 units. Some representative results obtained for Case 1 of (5.4) are shown in Figures 5.1 - 5.4, where it can be seen that $\phi(t)$ is gracefully regulated to the desired value $\phi(t) = 0$, and is closely maintained there, while the two end-links undergo their continuous back-and-forth scanning

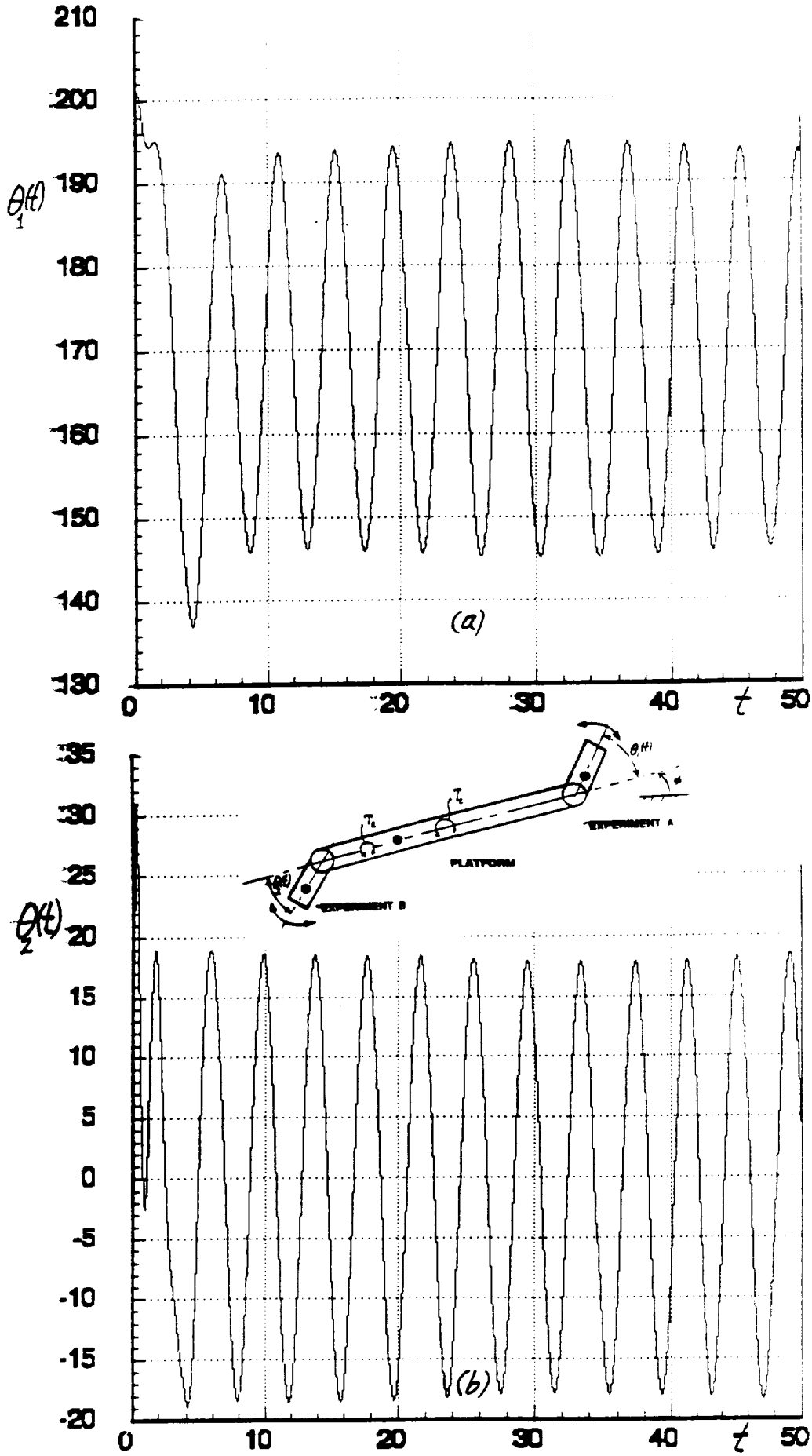


Figure 5.1 Closed-Loop Simulation Results for the Case 1 Configuration (5.4) with $T_e(t) \equiv 0$:, $\omega_1 = 1.45$, $\omega_2 = 1.60$; (a) θ_1 , (b) θ_2 [results cont'd. in Fig. 5.2].

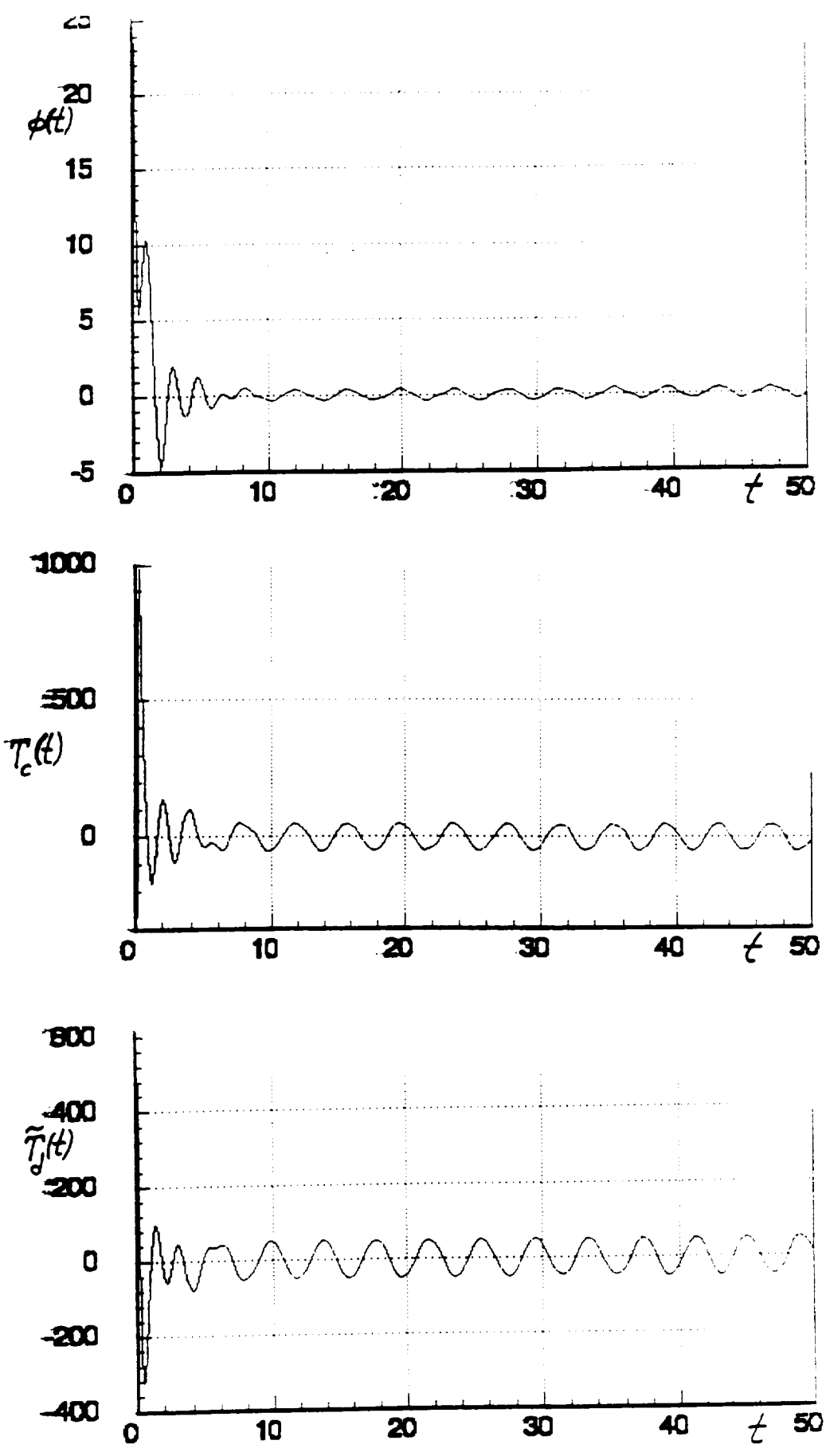


Figure 5.2 Closed-Loop Simulation Results for the Case 1 Configuration (5.4) with $T_e(t) \equiv 0$:, $\omega_1 = 1.45$, $\omega_2 = 1.60$

References Cited

1. Johnson, C.D., "Accommodation of Disturbances in Linear Regulator and Servomechanism Problems," *IEEE Trans. Auto. Cont., AC-16 No. 6*, p. 635, 1971. (Special Issue on the Linear-Quadratic-Gaussian Problem).

2. Johnson, C. D., "Theory of Disturbance-Accommodating Controllers," Chapt. in *Control and Dynamic Systems: Advances in Theory and Applications, Vol. 12*, ed. C. T. Leondes, Academic Press, Inc., New York, 1976.

3. Johnson, C. D., "A Discrete-Time, Disturbance-Accommodating Control Theory for Digital Control of Dynamical Systems,: Chapt in *Control and Dynamic Systems: Advances in Control and Dynamic Systems, Vol. 27*, ed. C. T. Leondes, Academic Press, 1988.

4. Johnson, C. D., "Disturbance-Accommodating Control; An Overview," *Proc. 1986 Amer. Control Conf.*, Seattle, Wash., Vol. 1, pp. 526-536, June 1986.

5. Kane, Thomas R., *Dynamics*, Holt, Rinehart and Winston, Inc., N.Y., 1968.

6. Townsend, Miles A., "Kane's Equations, Lagrange's Equations, and Virtual Work," *AIAA J. of Guid. and Control, Vol. 15, No. 1* , p. 277, Jan-Feb. 1992.

7. Storch, J. and S. Gates, "Motivating Kane's Method for Obtaining Equations of Motion for Dynamical Systems," *AIAA J. of Guid. and Control, Vol. 14, No. 4*, p. 593. 1989.

8. Kane, Thomas R., and David Levinson, *Dynamics; Theory and Applications*, McGraw-Hill Publishing Co., 1985.

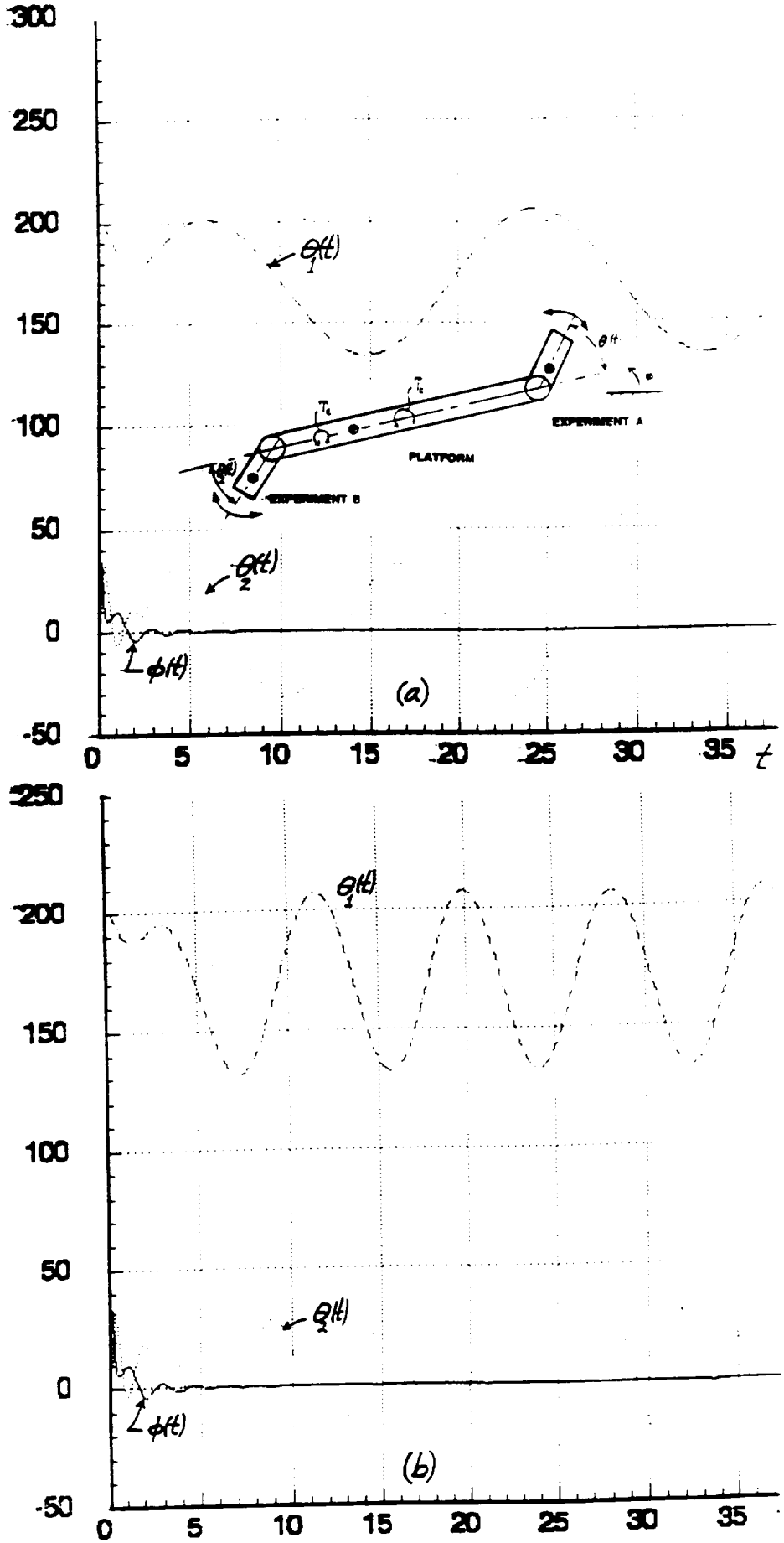
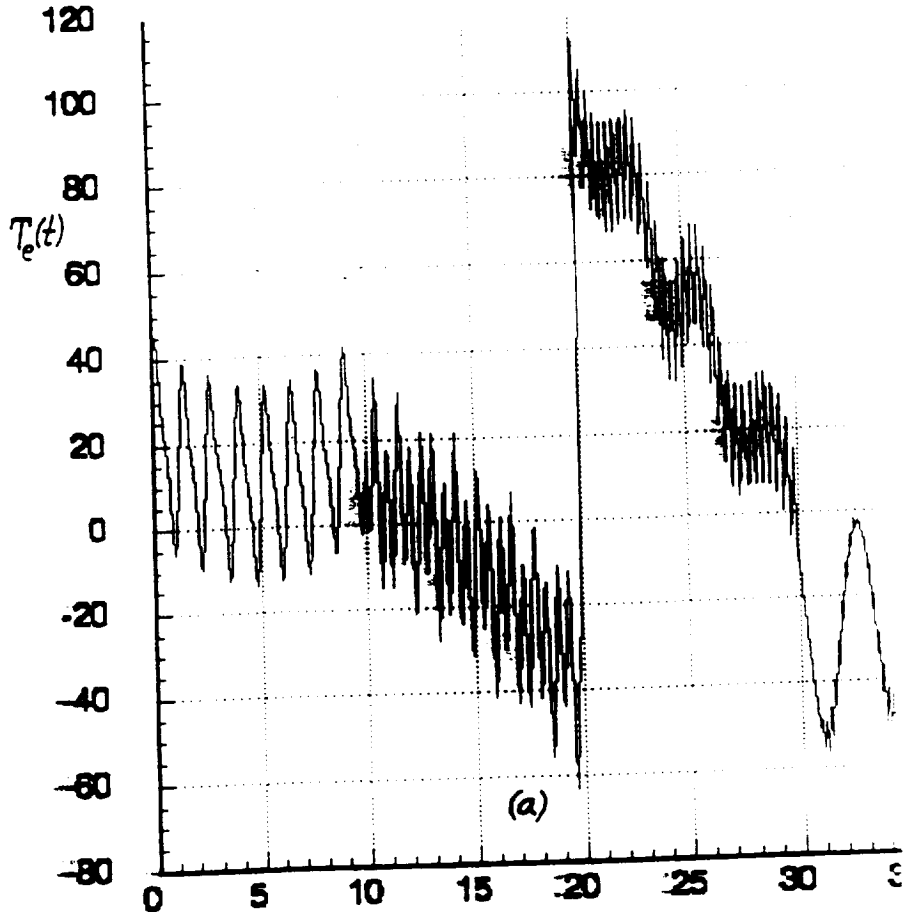


Figure 5.3 Closed-Loop Simulation Results for the Case 1 Configuration (5.4) with $T_e(t) \equiv 0$: (a) with $\omega_1 = 0.33$, $\omega_2 = 0.5$; (b) with $\omega_1 = 0.75$ $\omega_2 = 1.0$.



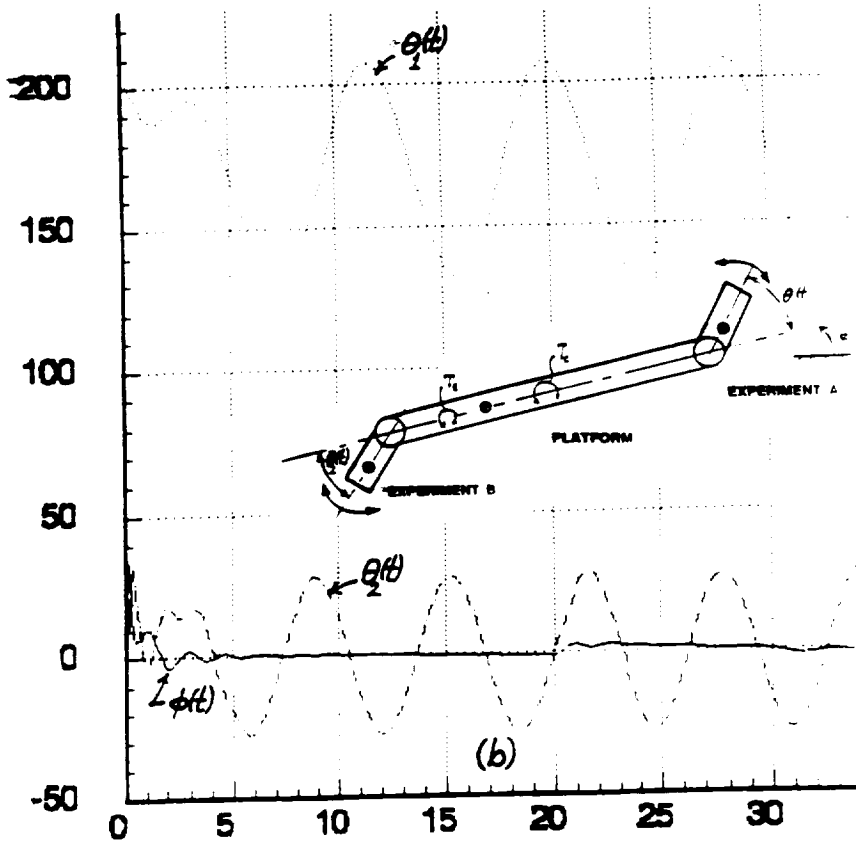


Figure 5.4 Closed-Loop Simulation Results for the C
Configuration (5.4) with $T_e(t) \neq 0$: ω_1 = (
ω_2 = 1.0; (a) $T_e(t)$; (b) $\theta_1(t)$, θ_2 (t), $\phi(t)$.

motions (5.9), (5.10). In other simulation runs for Case 1 of (5.4), the initial-conditions (5.11) were varied over a wide range with results essentially the same as shown in Figures 5.1 - 5.4.

Thus it can be concluded that for the Case 1 configuration defined in (5.4) the proposed platform controller does indeed achieve and maintain a "quiet" platform $\phi(t) \approx 0$, in the face of the typical equipment scanning motions $\theta_1(t)$, $\theta_2(t)$ associated with the scanning commands (5.9) , (5.10) *—provided the two experiment controllers have their individual closed-loop poles set at* $\lambda_1 = \lambda_2 = \lambda_3 = -1$. The reader is cautioned that this conclusion is highly dependent on the assumption of a Case 1 configuration (5.4) for the experiment controller closed-loop poles. In the next section, it will be demonstrated that a seemingly innocuous tightening of the experiments' individual closed-loop responses (i.e. placing the experiment closed-loop poles deeper into the left-half plane as in Case 2, Equation (5.5) will result in the onset of *instability* for the whole MPMSP system!

5.3 Simulation Results for the Case 2 Configuration (5.5)

The simulation runs associated with Figures 5.2, 5.2 were repeated with exactly the same parameter values, with the exception that the individual experiment's closed-loop poles were moved deeper into the left-half plane in accordance with Case 2 defined in (5.5). The corresponding plots of $\phi(t)$, $\theta_1(t)$, $\theta_2(t)$, etc. are shown in Figures 5.5 and 5.6, where it can be seen that the platform tilt $\phi(t)$ in this case *does not* approach zero, but rather oscillates with increasing amplitude. This overall system instability is physically due to the individual experiment's controllers being *too reactive* to the platform base-motions induced by each

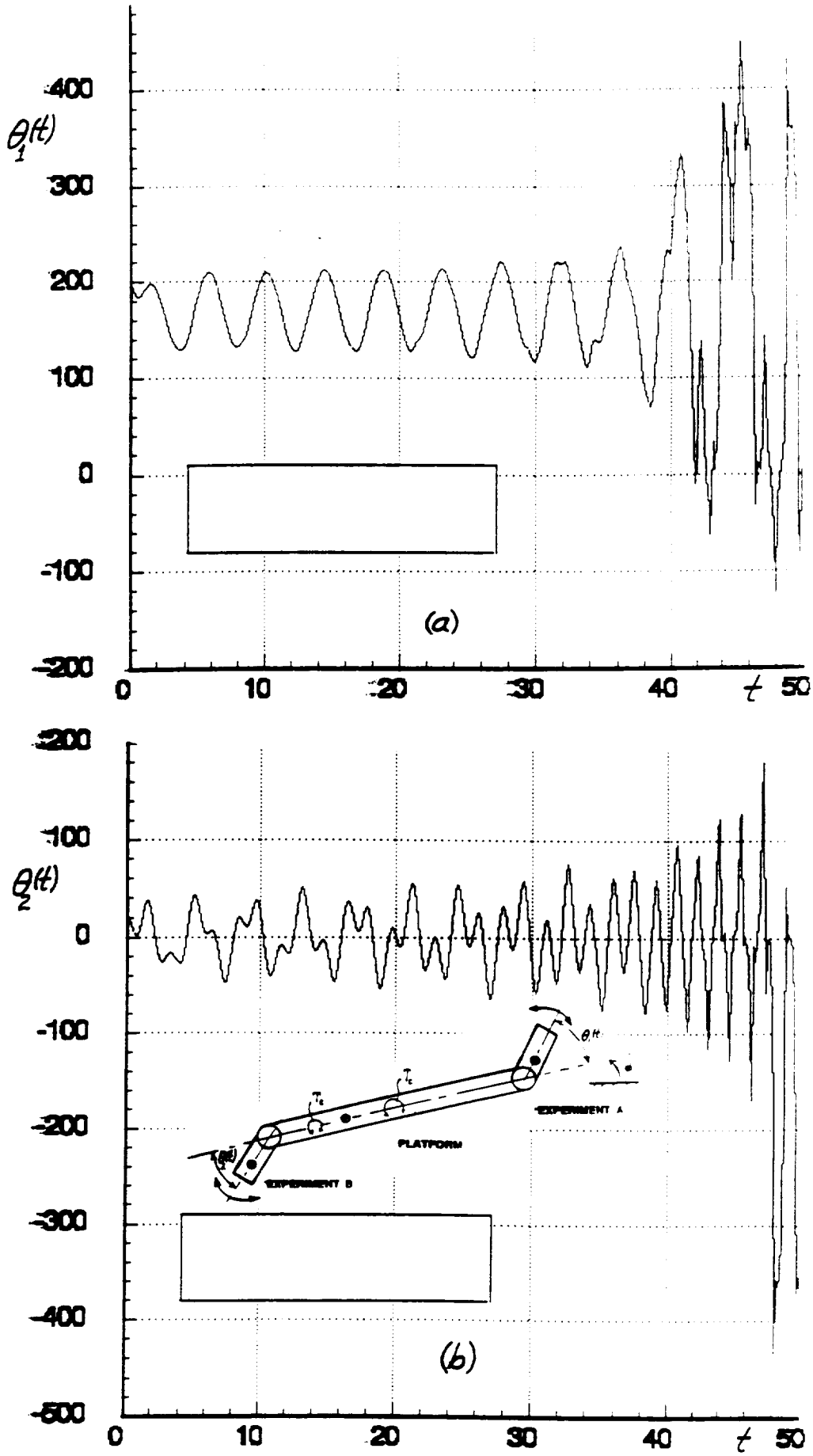
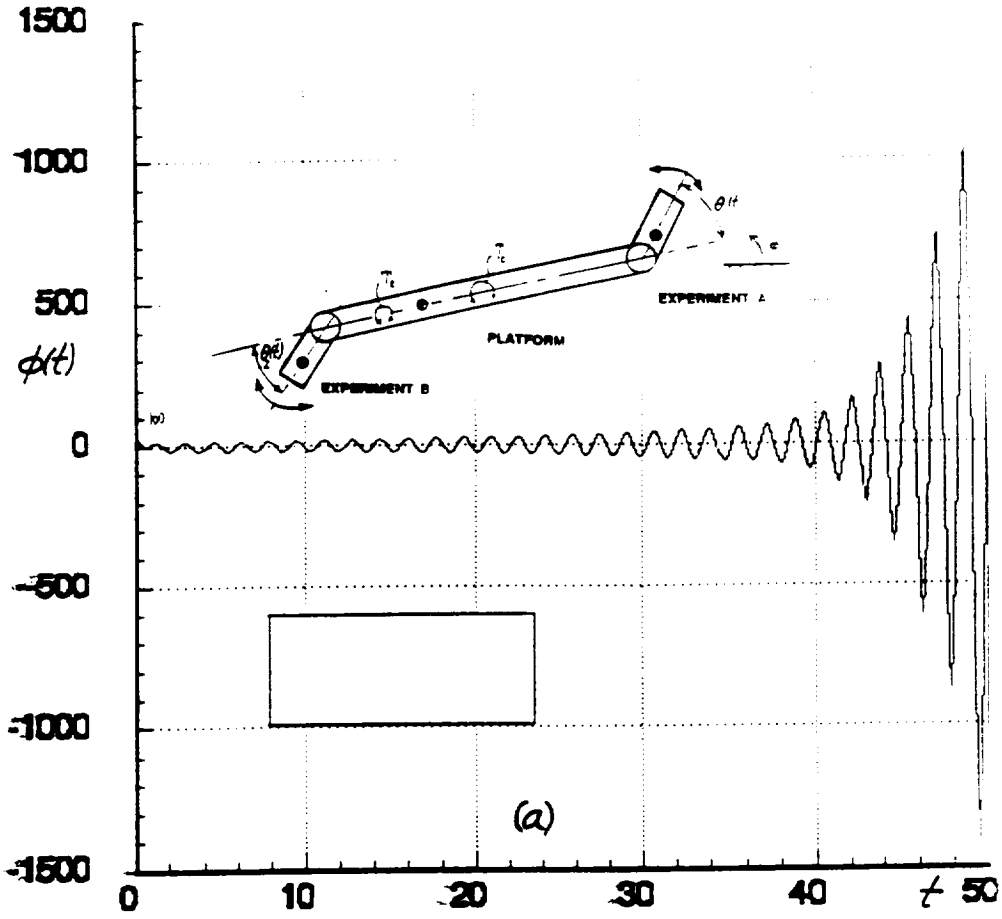


Figure 5.5 Closed-Loop Simulation Results for the Case 2 Configuration (5.5) Showing De-stabilization Phenomena Caused by Experiment Controllers Being Too Stable:(a) $\theta_1(t)$, (b) $\theta_2(t)$ [Results cont'd in Fig. 5.6].



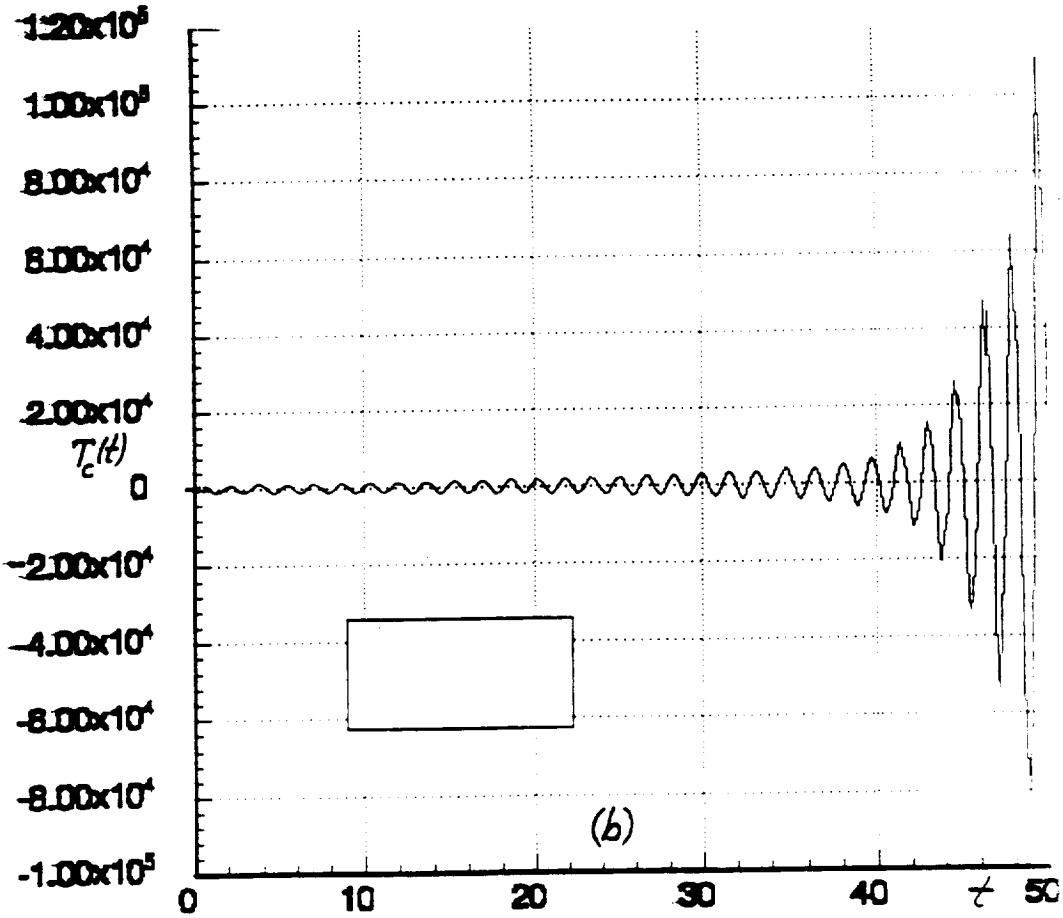


Figure 5.6 Closed-Loop Simulation Results for the Case 2 Configuration (5.5) Showing De-stabilization Phenomena Caused by Experiment Controllers Being Too Stable:(a) ϕ (t), (b) $T_c(t)$.

experiments' motions $\theta_i(t)$. In particular, it appears that the *integral-terms* in the experiments' PID controllers (5.1) are too reactive (i.e. respond too quickly) to the platform base-motion disturbances $\phi(t)$ in the Case 2 configuration.

Of course, technically all the system parameters play some role in creating a condition of dynamic instability, but the experience gained in exercising the Case 1 and Case 2 configurations seems to suggest that the *controller gains* k_{i3} associated with the *integral-terms* of the PID controllers (5.1) are the primary determinants of the observed instability. For example, if the integral-term gains k_{i3} in (5.1) for each experiment controller are set to zero, while in the Case 2 configuration (5.5), the system once again becomes stable. In that case, the local experiment controllers have only two (2) closed-loop poles each. However, if the remaining (P.D.) gains (k_{i1}, k_{i2}) in (5.1) are then re-adjusted to place those 2 experiment controller closed-loop poles at $\lambda_1 = \lambda_2 = -4.0$ (for each experiment), it was observed that *the system instability reappeared.*

5.4 An Analytical Approach to the Analysis and Prediction of Conditions that Cause MPMSP System Instability

The stable and unstable performances experimentally observed in the closed-loop simulation studies of Cases 1 and 2 in (5.4), (5.5) as described in the preceding two sections can, in principle, be studied analytically by *linearizing* the "exact" nonlinear system model (4.1) and employing the classical Routh/Hurwitz stability conditions on the resulting linear, constant coefficient model [with the linear platform controller (3.32), (3.37), (3.45) and linear experiment controllers (5.1) installed]. In this way one can develop a set of simultaneous,

algebraic inequalities (involving the various controller gains and system parameters) that define the necessary and sufficient conditions for overall stability of the MPMSP system. This would permit a more definitive assessment of the precise conditions, among the system parameters, that trigger the on-set of controller-induced destabilization in this class of MPMSP problems. However, the set of algebraic inequalities so obtained will undoubtedly be rather large in number (>7) and will most surely be very complicated and nonlinear in structure. Consequently, the user "visualization" of the exact parametric stability conditions corresponding to those Routh/Hurwitz inequalities may be somewhat elusive.

An attempt to develop the Routh/Hurwitz stability inequalities, using the aforementioned procedure, was initiated near the end of the period of performance for the present contract. However, the time available did not permit completion of that attempt. This is an important area for further study in any follow-on effort.

5.5 Summary of Chapter 5

The simulation studies described in this Chapter have verified that the proposed MPMSP disturbance-adaptive platform controller (3.32), (3.37), (3.45) can, in principle, achieve $\phi(t) \approx 0$ while identifying and adapting to the kind of persistent, complex disturbances induced by the motions $\theta_1(t)$, $\theta_2(t)$ of experiment equipment mounted on the platform. The simulations have also revealed that even with the platform controller installed, the potential for the local experiment controllers in an MPMSP to begin "fighting" each other, and thereby triggering instability of the entire MPMSP system, as described in Chapter 1 of this report, is very real. In fact, it appears that *this instability liability will exist*

for any realistic form of platform controller. The challenge, therefore, is to understand the exact mechanism of this instability and to design the platform and experiment controllers (and other features of the MPMSP) to yield a comfortable margin of "overall system stability" under *all* operating conditions.

Chapter 6

SUMMARY AND RECOMMENDATIONS FOR FURTHER WORK

6.1 Summary of Findings and Lessons Learned

This study has revealed that the analytical modeling of the dynamics of a Multip Pointing-Mount Space Platform (MPMSP) is a surprisingly complicated task, even structural flexibilities are neglected and even if only planar rotational motions a considered. It appears that all of the traditional principles of classical dynamics are woefu inadequate for deriving the "exact" equations of motion for MPMSP-type systems. On t other hand, the relatively new (*circa* 1968) method of dynamic modeling developed by Ka [8] appears to be ideally suited for rapidly deriving the exact equations of motion f MPMSP-type systems. Kanes' method, as embodied in Levinson's computer-aided modeli program, *Autolev*@, was used in this study to automatically derive the exact equations planar motion (4.1) for the generic MPMSP as shown in Figure 2.1.

The principles of Disturbance-Accommodating Control (DAC), with so modifications, were used here to derive and demonstrate a new control concept f stabilizing the platform motions $\phi(t)$ in the generic MPMSP model. This new platfor controller concept does not rely on the direct, real-time measurement of the "disturbanc torques" induced on the platform by the respective experiment motions, but rather uses "disturbance-observer" to identify the resultant of those disturbance torques, in real-tim from measurements of only the platform motions $\phi(t)$. The resulting controller is able

quickly adapt to a rather wide variety of environment, equipment, and experiment-relat uncertain, time-varying disturbances of the type that are expected in a realistic MPM project.

In a typical MPMSP, the maintenance of a "quiet" platform (i.e. $\phi(t) \approx 0$) is obviou essential for preventing one experiment's motions from interfering with the precis pointing requirements of other (simultaneously acting) experiments. A not so obvious f uncovered in this study, is that a sufficiently quiet platform is also essential for prevent the independent experiment controllers from "fighting" each other (overly reacting to disturbances that each experiment's motions induce on the other experiments, via platform "base-motions") and thereby triggering an unexpected violent instability of entire MPMSP system. This overall instability tends to be counter-intuitive because it occur even though the individual experiment controllers, and the platform controller, are very stable. In fact, experimental evidence developed in this study suggests that the ove system instability tends to occur because the individual experiment controllers are too sta (i.e.. the experiment controllers have their individual closed-loop poles too deep in the l half plane) and are thus too responsive!

It appears that for any realistic-type platform controller (including the one develo here) there will always exist a set of (seemingly stable) experiment controller parameters t will, in fact, cause *instability* of the overall MPMSP system. The through understandin this phenomena, and steps one can take to assure it does not occur, should be a major fa of concern in any planned MPMSP design.

6.1.1 A Video Animation of the MPMSP Destabilization Phenomena

A video tape recording of a computer-generated animation of the 3-link model, Figure 2.1, in closed-loop stable and unstable motion was prepared and provided to MSFC's Pointing Control Systems Branch, to illustrate this important feature. A description of that computer-animation program is presented in Appendix B of this report.

6.2 Recommendations for Further Work

The analytical analysis and visualization of the parametric "mechanism of instability" in an MPMSP system is considered highly important for the safe, effective design and gain-sizing of both experiment controllers and the platform controller. In any follow-on effort this topic should be addressed via the linearization/Routh-Hurwitz methodology outlined in Section 5.5 of this report.

The platform controller designed herein was based on the worst-case assumption of having access to only the *one* measurement $\phi(t)$. Further studies should consider cases in which one has access to , say, platform rate and acceleration measurements $\phi(t)$, $\dot{\phi}(t)$ and to direct measurement of, say, the controlled torques $T_1(t)$, $T_2(t)$ exerted by the individual experiment controllers. Such considerations may result in reduced complexity and enhanced performance of the new platform controller proposed here.

Acknowledgments

The author would like to acknowledge the many helpful suggestions and dynamically illuminating insights provided by Dr. Henry Waites, Pointing Control Systems Branch, NASA Marshall Space Flight Center, during the course of this investigation.

The exercising of the *Autolev*@ simulation model and the development of the animated-motion, computer monitor presentation of the dynamics of the 3-link model were provided by Mr. Mitchell Hunt, UAH/ECE Ph.D. student, who also made important contributions to the DAC-theory modifications needed to accomplish real-time identification of disturbances $\tilde{T}_d$ (t) in this particular application.

Finally, the author would like to express his sincere appreciation to David Levinson, Lockhheed Missiles and Space Co. Inc., who introduced this author to Kane's method and to the *Autolev* modeling program at a time when it appeared (to this author) that the intrinsic difficulty of the 3-link modeling task posed an insurmountable roadblock to the completion of this contract.

APPENDIX A

Autolev-Generated Fortran Code for Simulation of the 3-Link "Exact" Model

```
C      THE NAME OF THIS PROGRAM IS DAC3L.FOR
C      CREATED BY AUTOLEV ON 06-03-1992 AT 07:12:58
C
       IMPLICIT DOUBLE PRECISION (A-Z)
       INTEGER JLOOP,NSTEPS,NCUTS,NEQS,ILOOP,COUNTER,NPSTEP
       LOGICAL STPSZ
       EXTERNAL EQNS
       CHARACTER MSG(75)
       DIMENSION U(17)
       COMMON/CZEES/Z(45)
       COMMON/CPAR/J0,J1,J2,M0,M1,M2,PI,DEGTORAD,RADTODEG,L01,L02,L1,L
      &  L2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,OMEGA1,CENTER2,AMP2,OMEGA
       COMMON/CONT/W,W1,W2,THE1SP,THE2SP
       COMMON/DFQLST/T,STEP,RELERR,ABSERR,NCUTS,NEQS,STPSZ
C
       OPEN(UNIT=11,FILE='DAC3L.IN ',STATUS='UNKNOWN')
       OPEN(UNIT=12,FILE='DAC3L.OU1',STATUS='UNKNOWN')
       OPEN(UNIT=13,FILE='DAC3L.OU2',STATUS='UNKNOWN')
       OPEN(UNIT=14,FILE='DAC3L.OU3',STATUS='UNKNOWN')
       OPEN(UNIT=15,FILE='DAC3L.OU4',STATUS='UNKNOWN')
       OPEN(UNIT=21,FILE='DAC3L.H  ',STATUS='UNKNOWN')
       OPEN(UNIT=31,FILE='DAC3L.CO1',STATUS='UNKNOWN')
       PI = 4.0D0*DATAN(1.0D0)
       DEGTORAD = PI/180.0D0
       RADTODEG = 1.0D0/DEGTORAD
       WRITE(*,6001)
C
*******************************************************************
*
*              NOTE REGARDING INPUT AND OUTPUT DATA FILES
*
*  The user must supply an input data file to this program.  The
*  file must be named FILENAME.IN , where FILENAME is obtained from
*  the first line of this program.  The data must be arranged in
*  accordance with the READ statements that immediately follow this
*  NOTE.
*
*  The output from the program is sent to data files whose names
*  appear on the screen at the completion of each run.  The first
*  column in each output data file contains the time T,
*  running from zero to TMAX in increments of PSTEP.  TMAX, PSTEP,
*  and STEP are input from the terminal by the user at runtime.  STEP
*  being the initial integration stepsize, a number that should be
*  chosen to be less than or equal to PSTEP.  The terminal also
*  prompts the user for a message identifying the run.  This message
*  is printed on each of the output files.  Output files ending
*  in .OUn contain time-histories of generalized speeds and
*  generalized coordinates; files ending in .NRG contain kinetic
*  energy, potential energy, and total energy time-histories; files
*  ending in .H contain angular momentum time-histories; files ending
*  in .COn contain time-histories of quantities appearing as
*  arguments in CONTROLS commands; files ending in .SPn contain
*  time-histories of SPECIFIED variables; and files ending in .AUn
*  contain time-histories of force and/or torque measure numbers
*  corresponding to AUXILIARY generalized speeds.
```

```
*
*****************************************************************
C
      READ(11,*) L01,L02,L1,L2,KL2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,
     &  A1,CENTER2,AMP2,OMEGA2
      READ(11,*) M0,M1,M2
      READ(11,*) J0
      READ(11,*) J1
      READ(11,*) J2
      READ(11,*) U(1),U(2),U(3),U(4),U(5)
      READ(11,*) PHI,THETA1,THETA2,PHIABSINT,PHIINT1,THE1INT,THE2INT
     &  ,X02,Z01,Z02,Z03
C
C
      WRITE(* ,6002)
      READ( * ,6003) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(* ,6009)
      READ(*,*) TMAX,PSTEP,STEP0
      NPSTEP = IDINT((PSTEP-1.D-8)/STEP0 + 1)
      STEP   = PSTEP/NPSTEP
C
      WRITE(* ,6012)
      WRITE(* ,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(12,6101)
      WRITE(12,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(13,6102)
      WRITE(13,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(14,6103)
      WRITE(14,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(15,6104)
      WRITE(15,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(21,6151)
      WRITE(21,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(31,6201)
      WRITE(31,6010) (MSG(ILOOP),ILOOP = 1,75)
      WRITE(* ,6011)
      WRITE(12,6011)
C
      WRITE(* ,6500) L01,L02,L1,L2,KL2,KL1,KL0,KR2,KR1,KR0,CENTER1,A
     &  OMEGA1,CENTER2,AMP2,OMEGA2
      WRITE(12,6500) L01,L02,L1,L2,KL2,KL1,KL0,KR2,KR1,KR0,CENTER1,A
     &  OMEGA1,CENTER2,AMP2,OMEGA2
      WRITE(* ,6512) J0
      WRITE(12,6512) J0
      WRITE(* ,6513) J1
      WRITE(12,6513) J1
      WRITE(* ,6514) J2
      WRITE(12,6514) J2
      WRITE(* ,6600) M0,M1,M2
      WRITE(12,6600) M0,M1,M2
      WRITE(* ,6601) U(1),U(2),U(3),U(4),U(5)
      WRITE(12,6601) U(1),U(2),U(3),U(4),U(5)
      WRITE(* ,6602) PHI,THETA1,THETA2,PHIABSINT,PHIINT1,THE1INT,THE1
     &  ,X01,X02,Z01,Z02,Z03
      WRITE(12,6602) PHI,THETA1,THETA2,PHIABSINT,PHIINT1,THE1INT,THE1
```

```
     &  ,X01,X02,Z01,Z02,Z03
      WRITE(* ,6006) TMAX,PSTEP,STEP,STEP0
      WRITE(12,6006) TMAX,PSTEP,STEP,STEP0
C
      U(6) = PHI
      U(7) = THETA1
      U(8) = THETA2
      U(9) = PHIABSINT
      U(10) = PHIINT1
      U(11) = THE1INT
      U(12) = THE2INT
      U(13) = X01
      U(14) = X02
      U(15) = Z01
      U(16) = Z02
      U(17) = Z03
C
      WRITE(* ,6007)
      WRITE(21,6007)
      WRITE(12,6701)
      WRITE(13,6702)
      WRITE(14,6703)
      WRITE(15,6704)
      WRITE(31,6751)
C
      NEQS   = 17
      NCUTS  = 20
      T      = 0.0
      RELERR = 1.0D-8
      ABSERR = 1.0D-8
      STPSZ  = .FALSE.
      NSTEPS = IDINT(TMAX/STEP+0.1)+1
C
      COUNTER = 0
C
C
      DO 1000 JLOOP = 1 , NSTEPS
      CALL ZEES(T,U)
C
      IF (COUNTER.EQ.NPSTEP.OR.COUNTER.EQ.0) THEN
      CALL ANGMOM(T,U,HN1,HN2,HN3,HN)
      WRITE(21,6005) T,HN1,HN2,HN3,HN
      WRITE(* ,6005) T,HN1,HN2,HN3,HN
      WRITE(12,6005) T,U(1),U(2),U(3),U(4),U(5)
      WRITE(13,6005) T,U(6),U(7),U(8),U(9),U(10)
      WRITE(14,6005) T,U(11),U(12),U(13),U(14),U(15)
      WRITE(15,6005) T,U(16),U(17)
      CALL CNTRL(T,U)
      WRITE(31,6005) T,W,W1,W2,THE1SP,THE2SP
      COUNTER = 0
      ENDIF
C
      COUNTER = COUNTER + 1
      IF (JLOOP.EQ.NSTEPS) GO TO 1000
      CALL DEQS(EQNS,U,*99)
```

```
 1000 CONTINUE
C
      WRITE(*,6999)
C
      STOP
   99 WRITE(*,6004)
C
 6001 FORMAT(/1X,'SYSTEM PARAMETERS AND INITIAL CONDITIONS
     &       2X,'ARE NOW BEING READ FROM THE INPUT FILE'/)
 6002 FORMAT(1X,'INPUT A DESCRIPTION OF THIS RUN'/)
 6003 FORMAT(75A1)
 6004 FORMAT(1X,'STEPSIZE HALVED TOO MANY TIMES'/)
 6005 FORMAT(6(1X,1PE12.5))
 6006 FORMAT(11X,'TMAX = ',1PE12.5,' S'/10X,'PSTEP = ',1PE12.5,' S
     &'STEP = ',1PE12.5,' S    (USER INPUT VALUE = ',1PE12.5,' S)
 6007 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',11X,'HN1',10X,'HN2
     &HN3',10X,'HN',/6X,'(S)',8X,'(N*M*S)',6X,'(N*M*S)',6X,'(N*M*S
     &'(N*M*S)',/)
 6008 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',11X,'KE',11X,'PE',9
     &+ PE',/6X,'(S)',8X,'(UNITS)',6X,'(UNITS)',6X,'(UNITS)',/)
 6009 FORMAT(/1X,'INPUT TMAX, PSTEP, STEP  '/
     &1X,'=============================================='/
     &1X,'|    TMAX :  FINAL TIME                       |'/
     &1X,'|    PSTEP:  TIME INTERVAL FOR PRINTING        |'/
     &1X,'|    STEP :  MAXIMUM INTEGRATION TIME STEP     |'/
     &1X,'=============================================='/)
 6010 FORMAT(1X,'*** ',75A1)
 6011 FORMAT(//1X,'SYSTEM PARAMETERS'/)
 6012 FORMAT(1X,'OUTPUT FROM PROGRAM DAC3L.FOR'//)
 6101 FORMAT(1X,'FILE: DAC3L.OU1    (OUTPUT FROM PROGRAM DAC3L.FOR)
 6102 FORMAT(1X,'FILE: DAC3L.OU2    (OUTPUT FROM PROGRAM DAC3L.FOR'
 6103 FORMAT(1X,'FILE: DAC3L.OU3    (OUTPUT FROM PROGRAM DAC3L.FOR'
 6104 FORMAT(1X,'FILE: DAC3L.OU4    (OUTPUT FROM PROGRAM DAC3L.FOR
 6151 FORMAT(1X,'FILE: DAC3L.H    (OUTPUT FROM PROGRAM DAC3L.FOR)'
 6201 FORMAT(1X,'FILE: DAC3L.CO1    (OUTPUT FROM PROGRAM DAC3L.FOR
 6500 FORMAT(12X,'L01 = ',1PE12.5,' M'/12X,'L02 =  ,1PE12.5,' M'/1
     & = ',1PE12.5,' M'/13X,'L2 = ',1PE12.5,' M'/12X,'KL2 =  ,1PE1
     &UNITS'/12X,'KL1 = ',1PE12.5,' UNITS'/12X,'KL0 = ',1PE12.5,'
     &/12X,'KR2 = ',1PE12.5,' UNITS'/12X,'KR1 = ',1PE12.5,' UNITS
     &KR0 = ',1PE12.5,' UNITS'/8X,'CENTER1 = ',1PE12.5,' UNITS'/1
     &1 = ',1PE12.5,' UNITS'/9X,'OMEGA1 = ',1PE12.5,' UNITS'/8X,'O
     & = ',1PE12.5,' UNITS'/11X,'AMP2 = ',1PE12.5,' UNITS'/9X,'OME
     &',1PE12.5,' UNITS'/)
 6512 FORMAT(13X,'J0 = ',1PE12.5,' KG*M^2'/)
 6513 FORMAT(13X,'J1 = ',1PE12.5,' KG*M^2'/)
 6514 FORMAT(13X,'J2 = ',1PE12.5,' KG*M^2'/)
 6600 FORMAT(13X,'M0 = ',1PE12.5,' KG'/13X,'M1 =  ,1PE12.5,  KG'/1
     & = ',1PE12.5,' KG'//)
 6601 FORMAT(/1X,'INITIAL CONDITIONS'//10X,'U1(0) =  ,1PE12.5,  RA
     &0X,'U2(0) =  .1PE12.5,' RAD/S'/10X,'U3(0) =  .1PE12.5,  RAD
     &,'U4(0) = ',1PE12.5,' M/S'/10X,'U5(0) =  ,1PE12.5,  M/S'/)
 6602 FORMAT(9X,'PHI(0) = ',1PE12.5,' RAD'/6X,'THETA1(0) =  .1PE12
     &AD'/6X,'THETA2(0) =  .1PE12.5,' RAD'/3X,'PHIABSINT(0) =  .1P
     &' UNITS'/5X,'PHIINT1(0) = ',1PE12.5,' UNITS'/5X,'THE1INT(0)
     &E12.5,' UNITS'/5X,'THE2INT(0) =  .1PE12.5,' UNITS'/9X,'X01(
```

```
     &1PE12.5,' RAD'/9X,'XO2(O) =  .1PE12.5,' RAD'/9X,'ZO1(O) =  .
     &5,' RAD'/9X,'ZO2(O) = ',1PE12.5,' RAD'/9X,'ZO3(O) =  .1PE12.
     &D'//)
6701 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',11X,'U1',11X,'U2',1
     &,11X,'U4',11X,'U5',/6X,'(S)',8X,'(RAD/S)',6X,'(RAD/S)',6X
     &,7X,'(M/S)',8X,'(M/S)',/)
6702 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',11X,'PHI',9X,'THET
     &'THETA2',6X,'PHIABSINT',5X,'PHIINT1',/6X,'(S)',9X,'(RAD)',8
     &),8X,'(RAD)',7X,'(UNITS)',6X,'(UNITS)',/)
6703 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',9X,'THE1INT',6X,'TH
     &,8X,'XO1',10X,'XO2',10X,'ZO1',/6X,'(S)',8X,'(UNITS)',6X,'(UN
     &7X,'(RAD)',8X,'(RAD)',8X,'(RAD)',/)
6704 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',11X,'ZO2',10X,'ZO3
     &(S)',9X,'(RAD)',8X,'(RAD)',/)
6751 FORMAT(//1X,'SIMULATION RESULTS'//7X,'T',12X,'W',11X,'W1',11
     &,9X,'THE1SP',7X,'THE2SP',/6X,'(S)',9X,'(N*M)',7X,'(UNITS)',6
     &ITS)',6X,'(UNITS)',6X,'(UNITS)',/)
6999 FORMAT(//1X,'OUTPUT IS ON FILES:  ','DAC3L.OU1'/22X,'DAC3L.O
     &X,'DAC3L.OU3'/22X,'DAC3L.OU4'/22X,'DAC3L.H'/22X,'DAC3L.CO1'
      END
```

```
      SUBROUTINE EQNS(T,U,UDOT)
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIMENSION U(17),UDOT(17),COEF(5,5),RHS(5
      COMMON/CZEES/Z(45)
      COMMON/CPAR/J0,J1,J2,M0,M1,M2,PI,DEGTORAD,RADTODEG,L01,L02,L1,L
     &  L2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,OMEGA1,CENTER2,AMP2,OMEGA
      COMMON/CONT/W,W1,W2,THE1SP,THE2SP
C
C
      CALL ZEES(T,U)
C
      PHI = U(6)
      THETA1 = U(7)
      THETA2 = U(8)
      PHIABSINT = U(9)
      PHIINT1 = U(10)
      THE1INT = U(11)
      THE2INT = U(12)
      XO1 = U(13)
      XO2 = U(14)
      ZO1 = U(15)
      ZO2 = U(16)
      ZO3 = U(17)
C
C
      CALL CNTRL(T,U)
C
      COEF(1,1) = Z(27)
      COEF(1,2) = -Z(28)
      COEF(1,3) = -Z(29)
      COEF(1,4) = Z(30)
      COEF(1,5) = Z(31)
      COEF(2,1) = -Z(28)
      COEF(2,2) = -Z(32)
      COEF(2,3) = 0.0
      COEF(2,4) = Z(34)
      COEF(2,5) = -Z(35)
      COEF(3,1) = -Z(29)
      COEF(3,2) = 0.0
      COEF(3,3) = -Z(37)
      COEF(3,4) = -Z(38)
      COEF(3,5) = Z(39)
      COEF(4,1) = Z(30)
      COEF(4,2) = Z(34)
      COEF(4,3) = -Z(38)
      COEF(4,4) = -Z(41)
      COEF(4,5) = 0.0
      COEF(5,1) = Z(31)
      COEF(5,2) = -Z(35)
      COEF(5,3) = Z(39)
      COEF(5,4) = 0.0
      COEF(5,5) = -Z(41
```

```
      RHS(1) = -W+Z(32)
      RHS(2) = -W1-Z(36)
      RHS(3) = -W2-Z(40)
      RHS(4) = Z(42)
      RHS(5) = Z(43)

      CALL UNCOPL(5,COEF,RHS,UDOT)

C
C
C     U6 IS DEFINED TO BE PHI
      UDOT(6) = U(1)

C
C     U7 IS DEFINED TO BE THETA1
      UDOT(7) = U(2)

C
C     U8 IS DEFINED TO BE THETA2
      UDOT(8) = U(3)

C
C     U9 IS DEFINED TO BE PHIABSINT
      UDOT(9) = ABS(PHI)

C
C     U10 IS DEFINED TO BE PHIINT1
      UDOT(10) = PHI

C
C     U11 IS DEFINED TO BE THE1INT
      UDOT(11) = THE1SP-THETA1

C
C     U12 IS DEFINED TO BE THE2INT
      UDOT(12) = THE2SP-THETA2

C
C     U13 IS DEFINED TO BE X01
      UDOT(13) = X02+25.*(PHI-X01)

C
C     U14 IS DEFINED TO BE X02
      UDOT(14) = Z01/J0+W/J0+250.*(PHI-X01)

C
C     U15 IS DEFINED TO BE Z01
      UDOT(15) = Z02+1250.*J0*(PHI-X01)

C
C     U16 IS DEFINED TO BE Z02
      UDOT(16) = Z03+3125.*J0*(PHI-X01)

C
C     U17 IS DEFINED TO BE Z03
      UDOT(17) = 3125.*J0*(PHI-X01)

C
C

      RETURN
      END
```

```
C
C
C
      SUBROUTINE UNCUPL(NDIM,COEF,RHS,UDOT)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER NDIM,IPS(50)
      DIMENSION COEF(NDIM,NDIM),RHS(NDIM),UDOT(NDIM)
C
      CALL DECMP2(NDIM,COEF,NDIM,COEF,IPS,*901,*902)
      CALL SOLVE2(NDIM,COEF,NDIM,RHS,UDOT,IPS)
C
      RETURN
C
  901 WRITE(*,601)
      STOP
  902 WRITE(*,602)
      STOP
  601 FORMAT(/1X,'ALL ELEMENTS IN A ROW OF COEF ARE ZEROS'/)
  602 FORMAT(/1X,'A PIVOT ELEMENT ENCOUNTERED IN THE DECOMPOSITION
     C          ' OF COEF IS ZERO')
      END
```

```
C
C
C
      SUBROUTINE ZEES(T,U)
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIMENSION U(17)
      COMMON/CZEES/Z(45)
      COMMON/CPAR/J0,J1,J2,M0,M1,M2,PI,DEGTORAD,RADTODEG,L01,L02,L1,
     &  L2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,OMEGA1,CENTER2,AMP2,OMEGA
C
      PHI = U(6)
      THETA1 = U(7)
      THETA2 = U(8)
      PHIABSINT = U(9)
      PHIINT1 = U(10)
      THE1INT = U(11)
      THE2INT = U(12)
      XO1 = U(13)
      XO2 = U(14)
      ZO1 = U(15)
      ZO2 = U(16)
      ZO3 = U(17)
C
C
      S1 = DSIN(THETA1)
      C1 = DCOS(THETA1)
      S2 = DSIN(THETA2)
      C2 = DCOS(THETA2)
C
      Z(1) = L01*S1
      Z(2) = C1*L01
      Z(3) = L1+Z(2)
      Z(4) = L02*S2
      Z(5) = C2*L02
      Z(6) = L2+Z(5)
      Z(7) = U(1)*U(5)
      Z(8) = U(1)*U(4)
      Z(9) = C1*L01*U(2)
      Z(10) = L01*S1*U(2)
      Z(11) = C1*U(4)+S1*U(5)+U(1)*Z(1)
      Z(12) = C1*U(5)+L1*U(2)-S1*U(4)+U(1)*Z(3)
      Z(13) = U(1)+U(2)
      Z(14) = C1*U(5)-S1*U(4)
      Z(15) = U(1)*Z(9)+U(2)*Z(14)-Z(12)*Z(13)
      Z(16) = C1*U(4)+S1*U(5)
      Z(17) = U(1)*Z(10)+U(2)*Z(16)-Z(11)*Z(13)
      Z(18) = C2*L02*U(3)
      Z(19) = L02*S2*U(3)
      Z(20) = C2*U(4)+S2*U(5)-U(1)*Z(4)
      Z(21) = C2*U(5)-L2*U(3)-S2*U(4)-U(1)*Z(6)
      Z(22) = U(1)+U(3)
      Z(23) = C2*U(5)-S2*U(4)
      Z(24) = U(1)*Z(18)-U(3)*Z(23)+Z(21)*Z(22)
      Z(25) = C2*U(4)+S2*U(5)
      Z(26) = U(1)*Z(19)-U(3)*Z(25)+Z(20)*Z(22)
```

```
      Z(27) = (-Z(1)*Z(1)-Z(3)*Z(3))*M1+(-Z(4)*Z(4)-Z(6)*Z(6))*M2-J0-J
     &  -J2
      Z(28) = J1+L1*M1*Z(3)
      Z(29) = J2+L2*M2*Z(6)
      Z(30) = (-C1*Z(1)+S1*Z(3))*M1+(C2*Z(4)-S2*Z(6))*M2
      Z(31) = (-C1*Z(3)-S1*Z(1))*M1+(C2*Z(6)+S2*Z(4))*M2
      Z(32) = (Z(1)*Z(15)-Z(17)*Z(3))*M1+(Z(24)*Z(4)-Z(26)*Z(6))*M2
      Z(33) = J1+L1*L1*M1
      Z(34) = L1*M1*S1
      Z(35) = C1*L1*M1
      Z(36) = L1*M1*Z(17)
      Z(37) = J2+L2*L2*M2
      Z(38) = L2*M2*S2
      Z(39) = C2*L2*M2
      Z(40) = L2*M2*Z(26)
      Z(41) = M0+M1+M2
      Z(42) = (-C2*Z(24)-S2*Z(26))*M2+(C1*Z(15)+S1*Z(17))*M1-M0*Z(7)
      Z(43) = (-C1*Z(17)+S1*Z(15))*M1+(C2*Z(26)-S2*Z(24))*M2+M0*Z(8)
      Z(44) = ((-C2*L2-L02)*M2+(C1*L1+L01)*M1)/Z(41)
      Z(45) = (L1*M1*S1-L2*M2*S2)/Z(41)
C
      RETURN
      END
```

```
C
C
C
      SUBROUTINE CNTRL(T,U)
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIMENSION U(17)
      COMMON/CZEES/Z(45)
      COMMON/CPAR/J0,J1,J2,M0,M1,M2,PI,DEGTORAD,RADTODEG,L01,L02,L
     &  L2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,OMEGA1,CENTER2,AMP2,OME
      COMMON/CONT/W,W1,W2,THE1SP,THE2SP
C
      PHI = U(6)
      THETA1 = U(7)
      THETA2 = U(8)
      PHIABSINT = U(9)
      PHIINT1 = U(10)
      THE1INT = U(11)
      THE2INT = U(12)
      XO1 = U(13)
      XO2 = U(14)
      ZO1 = U(15)
      ZO2 = U(16)
      ZO3 = U(17)
C
C
      S1 = DSIN(OMEGA1*T)
      C1 = DCOS(OMEGA1*T)
      S2 = DSIN(OMEGA2*T)
      C2 = DCOS(OMEGA2*T)
C
      W = -J0*(7.0*U(1)+20.0*PHI+24.0*PHIINT1)-ZO1
      W1 = J1*(KL2*(-U(2))+KL1*(THE1SP-THETA1)+KL0*THE1INT)
      W2 = J2*(KR2*(-U(3))+KR1*(THE2SP-THETA2)+KR0*THE2INT)
      THE1SP = CENTER1+AMP1*S1
      THE2SP = CENTER2+AMP2*S2
C
      RETURN
      END
```

```
C
C
C
      SUBROUTINE ANGMOM(T,U,HN1,HN2,HN3,HN)
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIMENSION U(17)
      COMMON/CZEES/Z(45)
      COMMON/CPAR/J0,J1,J2,M0,M1,M2,PI,DEGTORAD,RADTODEG,L01,L02,L1
     &  L2,KL1,KL0,KR2,KR1,KR0,CENTER1,AMP1,OMEGA1,CENTER2,AMP2,OME
C
      PHI = U(6)
      THETA1 = U(7)
      THETA2 = U(8)
      PHIABSINT = U(9)
      PHIINT1 = U(10)
      THE1INT = U(11)
      THE2INT = U(12)
      X01 = U(13)
      X02 = U(14)
      Z01 = U(15)
      Z02 = U(16)
      Z03 = U(17)
C
C
      S1 = DSIN(THETA1)
      C1 = DCOS(THETA1)
      S2 = DSIN(THETA2)
      C2 = DCOS(THETA2)
C
      ZH1 = U(4)*Z(45)-U(5)*Z(44)
      ZH2 = C1*L1+L01-Z(44)
      ZH3 = L1*S1-Z(45)
      ZH4 = C1*Z(11)-S1*Z(12)
      ZH5 = C1*Z(12)+S1*Z(11)
      ZH6 = ZH2*ZH5-ZH3*ZH4
      ZH7 = -C2*L2-L02-Z(44)
      ZH8 = -L2*S2-Z(45)
      ZH9 = C2*Z(20)-S2*Z(21)
      ZH10 = C2*Z(21)+S2*Z(20)
      ZH11 = ZH10*ZH7-ZH8*ZH9
      HN1 = 0.0
      HN2 = 0.0
      HN3 = J0*U(1)+J1*Z(13)+J2*Z(22)+M0*ZH1+M1*ZH6+M2*ZH11
      HN  = DSQRT(HN1*HN1 + HN2*HN2 + HN3*HN3)
C
      RETURN
      END
```

```
      SUBROUTINE DECMP2(N,A,IDIM,LU,IPS,*,*)
      IMPLICIT DOUBLE PRECISION (A-Z)
      INTEGER N,IDIM,IPS(N),I,J,K,IP,KP,KP1,NM1,IDXPI
      DIMENSION A(IDIM,N),LU(IDIM,N),SCALES(100)
      ZERO=0.0D0
      DO 5 I=1,N
      IPS(I)=I
      ROWNRM=0.0D0
      DO 2 J=1,N
      LU(I,J)=A(I,J)
      ROWNRM=DMAX1(ROWNRM,DABS(LU(I,J)))
    2 CONTINUE
      IF(ROWNRM.EQ.ZERO) RETURN 1
      SCALES(I)=1.0/ROWNRM
    5 CONTINUE
      NM1=N-1
      DO 17 K=1,NM1
      BIG=0.0D0
      DO 11 I=K,N
      IP=IPS(I)
      SIZE=DABS(LU(IP,K))*SCALES(IP)
      IF(SIZE.LE.BIG) GO TO 11
      BIG=SIZE
      IDXPIV=I
   11 CONTINUE
      IF(BIG.EQ.ZERO) RETURN 2
      IF(IDXPIV.EQ.K) GO TO 15
      J=IPS(K)
      IPS(K)=IPS(IDXPIV)
      IPS(IDXPIV)=J
   15 KP=IPS(K)
      PIVOT=LU(KP,K)
      KP1=K+1
      DO 16 I=KP1,N
      IP=IPS(I)
      EM=LU(IP,K)/PIVOT
      LU(IP,K)=EM
      DO 16 J=KP1,N
      LU(IP,J)=LU(IP,J)-EM*LU(KP,J)
   16 CONTINUE
   17 CONTINUE
      IF(LU(IPS(N),N).EQ.ZERO) RETURN 2
      RETURN
      END
```

```
      SUBROUTINE SOLVE2(N,LU,IDIM,B,X,IPS)
      IMPLICIT DOUBLE PRECISION (A-H
      INTEGER I,J,IP,IP1,IM1,NP1,IBACK,N,IDIM,IPS(N
      DIMENSION LU(IDIM,N),B(N),X(N)
      NP1=N+1
      X(1)=B(IPS(1))
      DO 2 I=2,N
      IP=IPS(I)
      IM1=I-1
      SUM=0.0D0
      DO 1 J=1,IM1
      SUM=SUM+LU(IP,J)*X(J)
    1 CONTINUE
      X(I)=B(IP)-SUM
    2 CONTINUE
      X(N)=X(N)/LU(IPS(N),N)
      DO 4 IBACK=2,N
      I=NP1-IBACK
      IP=IPS(I)
      IP1=I+1
      SUM=0.0D0
      DO 3 J=IP1,N
      SUM=SUM+LU(IP,J)*X(J)
    3 CONTINUE
    4 X(I)=(X(I)-SUM)/LU(IP,I)
      RETURN
      END
```

```
      SUBROUTINE DEOS(F,*)
      IMPLICIT DOUBLE PRECISION (A-H
      INTEGER I,NCUTS,NEQ
      LOGICAL DBL,STPSZ
      EXTERNAL F
      COMMON/DFOLST/T,STEP,REL,ABS,NCUTS,NEQ,STPSZ
      DIMENSION F0(200),F1(200),F2(200),Y1(200),Y2(200),*(NEQ
      DATA HC/0.0D0/
C *** CHECK FOR INITIAL ENTRY AND ADJUST HC, IF NECESSARY.
      IF(NEQ.NE.0) GO TO 10
      HC=STEP
      RETURN
   10   IF(STEP.EQ.0.0D0) RETURN 1
C *** CHANGE DIRECTION, IF REQUIRED.
      IF(HC*STEP) 20,30,40
   20      HC=-HC
      GO TO 40
   30   HC=STEP
C *** SET LOCAL VARIABLES
   40   EPSL=REL
      FINAL=T+STEP
      H=HC
      TT=T+H
      T=FINAL
      H2=H/2.0D0
      H3=H/3.0D0
      H6=H/6.0D0
      H8=H/8.0D0
C *** MAIN KUTTA-MERSON STEP
   50   IF((H.GT.0.0D0.AND.TT.GT.FINAL).OR.
     C      (H.LT.0.0D0.AND.TT.LT.FINAL)) GO TO 190
   60   CALL F(TT-H,Y,F0)
      DO 70 I=1,NEQ
   70   Y1(I)=F0(I)*H3+Y(I)
      CALL F(TT-2.0*H3,Y1,F1)
      DO 80 I=1,NEQ
   80   Y1(I)=(F0(I)+F1(I))*H6+Y(I)
      CALL F(TT-2.0*H3,Y1,F1)
      DO 90 I=1,NEQ
   90   Y1(I)=(F1(I)*3.0+F0(I))*H8+Y(I)
      CALL F(TT-H2,Y1,F2)
      DO 100 I=1,NEQ
  100   Y1(I)=(F2(I)*4.0-F1(I)*3.0+F0(I))*H2+Y(I)
      CALL F(TT,Y1,F1)
      DO 110 I=1,NEQ
  110   Y2(I)=(F2(I)*4.0+F1(I)+F0(I))*H6+Y(I)
C *** DOES THE STEPSIZE H NEED TO BE CHANGED?
      IF(EPSL.LE.0.0D0) GO TO 170
      DBL=.TRUE.
      DO 160 I=1,NEQ
      ERR=DABS(Y1(I)-Y2(I))*0.2
      TEST=DABS(Y1(I))*EPSL
      IF(ERR.LT.TEST.OR.ERR.LT.ABS) GO TO 150
```

```
C *** HALVE THE STEPSIZE
      H=H2
      TT=TT-H2
      IF(.NOT.STPSZ) GO TO 120
      TEMP=TT-H2
      WRITE(*,200) H,TEMP
C *** HAS THE STEPSIZE BEEN HALVED TOO MANY TIMES?
  120   NCUTS=NCUTS-1
      IF(NCUTS.GE.0) GO TO 130
      T=TT-H2
      WRITE(*,210) T
      RETURN 1
C *** IF STEPSIZE IS TOO SMALL RELATIVE TO TT TAKE RETURN 1
  130   IF(TT+H.NE.TT) GO TO 140
      T=TT
      RETURN 1
  140   H2=H/2.0D0
      H3=H/3.0D0
      H6=H/6.0D0
      H8=H/8.0D0
      GO TO 60
  150   IF(DBL.AND.64.0D0*ERR.GT.TEST
     C        .AND.64.0D0*ERR.GT.ABS) DBL=.FALSE.
  160   CONTINUE
C *** DOUBLE THE STEPSIZE, MAYBE.
      IF(.NOT.DBL.OR.DABS(2.0D0*H).GT.DABS(STEP).OR.
     C     DABS(TT+2.0D0*H).GT.DABS(FINAL).AND.
     C     DABS(TT-FINAL).GT.DABS(FINAL)*1.0D-7) GO TO 170
      H2=H
      H=H+H
      IF(STPSZ) WRITE(*,200) H,TT
      H3=H/3.0D0
      H6=H/6.0D0
      H8=H/8.0D0
      NCUTS=NCUTS+1
  170   DO 180 I=1,NEQ
  180   Y(I)=Y2(I)
      TT=TT+H
      GO TO 50
  190   IF(EPSL.LT.0.0D0) RETURN
C *** NOW BE SURE TO HAVE T=FINAL.
      HC=H
      H=FINAL-(TT-H)
      IF(DABS(H).LE.DABS(FINAL)*1.0D-7) RETURN
      TT=FINAL
      EPSL=-1.0D0
      H2=H/2.0D0
      H3=H/3.0D0
      H6=H/6.0D0
      H8=H/8.0D0
      GO TO 60
  200   FORMAT(1X,'THE STEPSIZE IS NOW  ',1PD12.4,'  AT T =  ',1PD12
  210   FORMAT(1X,'THE STEPSIZE HAS BEEN HALVED TOO MANY TIMES;
     C          'T =  ',1PD12.4)
      END
```

APPENDIX B

Description of the Computer Animation Program for the "Exact" 3-Link MPMSP Model