MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Working Paper No. 192 February 1979

Logical Control Theory Applied to Mechanical Arms

Ronald Joseph Pankiewicz

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

Logical Control Theory Applied to Mechanical Arms

by

Ronald Joseph Pankiewicz

Submitted to the Department of Electrical Engineering and Computer Science on January 19, 1979 in partial fulfillment of the requirements for the Degrees of Master of Science and Electrical Engineer.

Abstract

A new control algorithm based upon Logical Control Theory is developed for mechanical manipulators. The controller uses discrete tesselations of state space and a finite set of fixed torques to regulate non-rehearsed movements in real time. Varying effective inertia, coupling between degrees of freedom, and frictional, gravitional and Coriolis forces are readily handled. A logical controller was implemented on a mini-computer for the MIT-Scheinman Vicarm. The controller's performance compares favorably with that of controllers designed according to existing methodologies as used, for example, in the control of present day industrial manipulators.

Thesis supervisor:

Berthold K. P. Horn,
Associate Professor of Electrical Engineering and Computer Science

Acknowledgements

I thank Berthold Horn for his patient and valuable guidance, his welcome insights and his critical readings of this document. Thanks also to Jon Doyle, Jan Galkowski and Marc Raibert for their encouragement, critical readings and contributions to this work, and to Meyer Billmers for his instruction on the use of the Mini-Robot System. I would also like to thank the AI Laboratory for providing me with the resources necessary to undertake this work.

Table of Contents

List of Illustrations

Overview

As highly nonlinear systems, mechanical manipulators are inherently difficult to control. The nonlinear characteristics are due to cross-coupling between degrees of freedom, changes in loading, and the effects of frictional, gravitational and Coriolis forces. Controllers designed with existing methodologies typically utilize complex compensation techniques to handle such systems. These techniques have proven inadequate for the real-time regulation of manipulators executing nonrehearsed movements at a variety of speeds.

This thesis explores the applicability of Logical Control Theory to the task of controlling a manipulator. Our research developed a logical controller that is essentially a finite state machine. It uses discrete tesselations of the system state-space and a set of fixed torques to regulate a mechanical arm. State-space trajectories acquired through modelling and observation are used to develop the tesselations. The number of different control torque levels and their magnitudes are determined on the basis of desired joint dynamics. The controller executes simple calculations to make logical decisions in regulating each joint, making it ideally suited for implementation on a digital computer. Since calculations to service one degree of freedom typically require less than one half of a millisecond on a mini-computer, original movements can be executed in real time.

The controller can regulate a mechanical arm with accuracy that is comparable to that achieved with existing approaches. Positioning accuracy to within ± 0.015 radians, and intermediate velocity regulation to within ± 0.15 radians per second was consistently realized in operating the three major joints of the MIT-Scheinman Vicarm in nonrehearsed

movements. Issues requiring further research were also identified.

I. Introduction

I.A General Background

The modern history of manipulation began in 1947 at Argonne National Laboratory, where the first mechanical arms were developed for handling radioactive materials. These were master-slave devices, in which the remote hand replicated motions of a similar hand controlled by the operator. Force feedback was eventually added to enable the operator to tell what forces the hand was exerting [Goertz, 63].

In 1961 Ernst developed a computer-controlled mechanical hand that had touch feedback [Ernst, 61]. The hand could be used to explore regions, indentifying and moving objects. In 1968 Pieper analyzed the kinematics of manipulators and was able to plan collision-free movements through cluttered spaces [Pieper, 68]. This work was followed by Kahn who analyzed arm dynamics and developed a "bang-bang" sub-optimal controller [Kahn & Roth, 71]. Since then, numerous efforts directed towards computer control have produced a variety of philosophies, of varying sophistication, on the control of mechanical manipulators [Finkel, 76]. Reference [IITRI, 75] provides a sampling of the current state of the art.

Today, applications of mechanical manipulators in industry include their performing boring and dangerous tasks which traditionally have required human participation. The benefits of such applications are increased productivity, production flexibility and improvement of individuals' working conditions. Potential applications includes the performing of routine tasks in hostile environments, such as explosive plants,

radioactive chambers or underground mines, or the performing of research in inaccessible environments, such as deep space or underseas.

The successful utilization of mechanical manipulators for a large portion of these tasks depends on their having the dexterity and reprogrammibility characteristic of their human counterparts. At present, manipulators cannot match the combined speed, strength and agility of the human arm and hand which they are supposed to replace; in fact, the inability to efficiently and effectively control manipulators with such characteristics is the primary limitation in their widespread application [Bejczy, 76]. There exists no currently perfected scheme for the *real-time* control of mechanical manipulators over a wide range of *non-repetitive* motions; for many applications either some of the servoing calculations for original movements must be pre-computed or an inferior controller must be used, thus limiting the flexibility of the manipulator.

I.B Manipulators

The general difficulty in controlling mechanical arms stems from their basic structure: a series of rigid links cascaded into an open-loop kinematic chain, typically with single-degree-of-freedom joints of revolute or prismatic type. One end of the chain is attached to a reference frame; relative motion between the links is effected by forces applied to the links, generally at the joints, chosen to result in particular configurations of the chain and positioning of its tip -the "business" end [Kahn & Roth, 71].

The most striking features of the dynamics of such a device are the interactions between degrees of freedom and the nonlinear characteristics due to the combined effects of:

(1) Effective inertia - The distribution of the arm's mass depends on the relative position of all its links. Changes in the relative position of the links can drastically change the effective inertia of the mass supported by an individual joint and the response of that mass to an actuator force.

(2) Dynamic coupling - A force applied by an actuator to accelerate one link will usually result in the acceleration of other links, depending on their relative orientation. The motion of any individual link can thus be a function of the combined effects of several actuators; regulation of the link must be realized accordingly, by means of all the affecting actuators.

(3) Coriolis forces - A body with a rotational velocity that also moves in a radial direction will experience a force that changes its rotational velocity.

(4) Gravity - In a gravitational field links accelerate differently for different configurations. The net effects of actuator forces on individual links can therefore depend on the current orientation of every link in the arm.

(5) Friction and backlash - Stiction, viscuous friction, coulomb friction, and backlash at each joint combine into a force that is a severely nonlinear and nonrepeatable function of position and velocity.

(6) Actuators - Any device used to generate a force at a joint will, at best, be linear only over a narrow range of operation. Due to phenomena such as saturation and hysteresis, the device is not always capable of producing a force proportional to the control signal.

(7) Sensors - Any instrument used to generate feedback signals from a manipulator will have a linear response only over a limited range of operation, since every transducer has a sensitivity threshold and a saturation point.

These factors are directly or indirectly responsible for the difficulties in properly controlling, or even thoroughly analyzing, most mechanical arms. Such difficulties are further compounded by the fact that a mechanical arm is a time-varying system in that there can be a dramatic change in mass and effective inertia whenever an object is picked up. (A special case of this problem is common to manipulators with hydraulic actuators due to the motion of the actuator pistons and linkages).

I.C Control Techniques

I.C.1 Movement Definition

A mechanical manipulator is operated with a control mechanism that is implemented either with analogue hardware devices or with software on a digital computer. To produce a movement, the controller is repeatedly issued commands (corresponding to desired position and/or velocity) tracing the movement. These commands directly or indirectly control the actuators to produce the desired movement. The task of generating the commands can be viewed as that of interpolating points (commonly called "set points" or "via points") comprising a path between the initial and final positions of each joint. There are essentially three approaches to producing these input commands (position and velocity values) for the controller.

The most common, but least flexible technique is to guide the passive manipulator through the desired movements, while a series of via points for each joint is recorded. If the series of points is sufficiently dense, it can be used as is. Otherwise mathematical interpolation must be used to fill in the series with more intermediate points [Corwin, 75]. This technique is commonly used in applications where the manipulator is to repeat a movement for a prolonged period of time. Guiding the arm through the motions allows a system to memorize (record) the via points comprising the movement, which can then be acted out ad infinitum.

A more flexible approach involves deriving a movement-describing function for each joint. Each function is derived to pass through the endpoints of the state-space trajectory of the movement, and embody certain constraints on the movement, such as

bounds on the acceleration and velocity. These functions are then used to parameterize the movement: via points are computed by tracing along the trajectories defined by the functions. Commonly used functions are trapezoids, polynomials of order sufficient to specify all constraints desired on a movement, or several lower order polynomials splined together [Blanchard, 76; Finkel, 76].

A more ambitious approach involves computing via points "on the fly". During a movement of the arm, a matrix relating changes in the real space location of the manipulator's tip to changes in joint-angle space is periodically calculated (this matrix is always a function of the *current* configuration of the arm). Joint via points are then calculated from this matrix and a parameterization of the desired movement of the manipulator tip [Whitney, 72].

Given that there is some high-level planner issuing a series of via points that define some manipulator movement, we now examine the principal types of controllers that are used to execute a movement.

I.C.2 The Classical Approach

The most popular approach involves a classical PID controller. For each degree of freedom, a control signal is computed from a weighted sum of the error (i.e. deviation of the actual position from the desired position), its derivative, and its integral. If the controller has a velocity input, then the velocity-error term is included in the control-signal computation [Ogata, 70]. In essence, the controller is driven by errors. Whether implemented with analog hardware devices or with software, the operational principles are

the same.

The application of the PID control methodology to a mechanical arm presumes that it is a nearly linear, time-invariant system with independent degrees of freedom. But as explained above, a manipulator is a profoundly nonlinear, varying and coupled system. The presumptions that vindicate the use of a PID controller are thus allowable only so long as the arm is operated at low velocities. Under such operating conditions, the unwieldy dynamics of the arm are dominated by frictional forces and can be ignored. However, at moderate velocities, factors such as coupling and nonlinearity become significant in the system's dynamics.

There are a number of cookbook methods for designing the controller, most of which necessarily assume that the to-be-controlled system is reasonably linear and time-invariant. Application of these methods to the designing of PID controllers for such well-behaved systems is a straightforward process that yields acceptable results. However, these techniques are of limited utility with non-linear systems [Tou, 59]. Of course, there exists a variety of techniques to facilitate the analysis and control of nonlinear systems. Among these are:

> (1) The "linearization" of the system's behavior by means of linear approximations for the system. Analysis is then done with techniques developed for linear systems. For slightly nonlinear systems, piecewise linear approximations of the system are used, and nonanalytic nonlinearities such as backlash and coulomb friction are ignored. For severely nonlinear systems, "describing functions" which approximate the response of the system by its fundamental harmonics are used; higher harmonics are assumed to be filtered out by system components with large inertia and long time constants [Thaler & Brown, 60].
>
> (2) The examination of system trajectories in the phase plane. For systems whose

equations can be solved in parametric form, explicit phase-trajectory equations can be determined and phase trajectories plotted directly. For systems with equations that can't be solved, phase trajectories can be approximated with graphical techniques involving isoclines, direction fields, or computer-generated parameterizations of the trajectories. Limit cycles in the system trajectories can indicate the existence and type of oscillatory states in the system. The types and locations of singular points in the trajectories can indicate the nature of equilibrium states and overshoots, as well as the speed of response and stability characteristics of the system [Tomovic, 66].

(3) The numerical evaluation of the system equations. The response characteristics of the system can be assessed from solutions to the system equations obtained by means of numerical integration. The process of numerical integration can be carried out with quadrature formulas based on various interpolative approximations for the system equations [Tou, 59].

(4) The derivation of a Liapunov function for the system. Such a function defines a hyperregion in state space where the system's total energy is continually decreasing. This function verifies the stability of the system within the region, and can also be used to analyze the rapidity of the system's response [Ogata, 70].

Variations on these and other methods provide an extensive repertoire of techniques for managing nonlinear systems, but there is no method generally suitable to such systems. Furthermore, all of these methods apparently depend upon having equations (of various degrees of accuracy and completeness) for the system; due to the interactive nature of the system components, the corresponding equations for most mechanical arms are normally quite difficult to derive. According to R. E. Kalman, "Classical control theory could never really cope with large-scale systems (n large, more than 1 control variable, etc.) because the formulas...are much too complex." [Kalman, Falb & Arbib, 69 (p. 65)]

A more important point is that the classical control strategy proves inadequate for highly nonlinear systems such as a manipulator. As K. Ogata explains, "The main disadvantage of conventional control theory, generally speaking, is that it is applicable only

to linear time-invariant systems having a single input and a single output. It is powerless for time-varying, nonlinear systems (except simple ones), and multiple-input, multiple-output systems." [Ogata, 70 (p. 663)] Also, the controller is driven by errors. Rapid system response thus requires large errors or great sensitivity to the error term. But great sensitivity to the error term can produce dynamic instability, and oscillations around the destination. The very nature of this control strategy thus forces a compromise between responsiveness and controllability plus accuracy: eventual implementation of the controller becomes a task of finding a delicate balance between acceptable regulation and devastating instability [Deutsch, 69]. In effect, responsiveness must be sacrificed to ensure that the system will be "well-behaved", in some real-world sense [Blanchard, 76].

Increasing the order of the servo loop with a double-integration feedback term or incorporating non-linear compensation into the servo loop are techniques used with some success to overcome these problems. In general these techniques are also of limited utility since they, too, can be difficult to analyze and impractical to implement [Blanchard, 76; Cosgriff, 58].

I.C.3 The Modern Approach

What is needed is to take a more realistic and intelligent look at the system to be controlled. Evidently, a control strategy must consider the true characteristics of the system, instead of trying to coerce the system into matching some desirable, but artificial constraints.

The need to control more complicated systems with greater accuracy motivated the development of modern control theory. In modern control theory, a system of equations,

typically several first-order differential equations, are derived to accurately describe the true nature of the system to be controlled. These equations are assembled into a vector-matrix equation to model the system's behavior. Intrinsic to this model is the notion of state, where the state of a system summarizes the effects of all past inputs to the system, and completely determines the future behavior of the system for any future inputs. In contrast to classical control theory, modern control theory naturally addresses multiple-input, multiple-output nonlinear and time-variant systems. Also, modern control theory is a time domain approach. The analysis and design of complex systems can accordingly be a comparatively straightforward, but flexible procedure [Ogata, 70]. Modern control theory thus provides a more versatile structure within which to describe and analyze a system.

All these qualities suggest the application of modern control theory as a reasonable alternative for the controlling of mechanical arms. A noteworthy application of this general approach involves a model that is derived from the formulation of the dynamics of an arm using Lagrange's equation [Paul, 72]. In their complete form, these model equations (one equation for the motion of each degree of freedom in its own generalized coordinate system) account for the simple dynamics of each link in the arm, as well as inertial changes, and forces due to frictions, gravity, and interaction between the links. The model equations are manipulated so that they represent the "inverse" of the arm dynamics. Actuator forces necessary to produce a given desired acceleration can then be computed using the inverse model equations. The actuator forces include compensation for the effects, as predicted by the equations, of "external" influences such as gravitational forces and changes in inertia. Within the accuracy of these equations, it is possible to calculate all

actuator forces necessary to accomplish a complete movement. In practice, feedback terms are incorporated into the calculations to compensate for simplifications in the system model and unmeasureable disturbances that limit the accuracy of the equations.

The shortcoming of such completely detailed equations is their computational density: they are not applicable for real-time calculations on a practical-sized computer. This problem is overcome by discarding terms representing forces that are comparatively small (e.g. velocity-product terms representing Coriolis forces). Again, feedback terms are relied upon to correct for the (increased) inaccuracy of these truncated equations. Still, performance of the system must be compromised to insure that the discarded terms remain "insignificant".

The basic problem with this strategy is that it must work with too much detail, because it is too general. In essence, the controller is designed to deal with an infinite range of inputs and an infinite range of possible system states. But in reality input forces and accelerations must be finite. And within the finite range of possible system inputs, there is, in all practicality, a finite and moderate number of inputs that have a significantly different effect on the system. Though a controller may, in theory, provide any one of infinitely many inputs possible over a given range, a limited number of the possible inputs are detectably different, and of these inputs, even fewer have a discernably different effect on the system. Thus a quantized approximation of the control inputs might suffice to control the system.

In addition, the actual state of the system is resolveable only with limited accuracy. This also indicates that a quantized approximation of the system state could be sufficient. That is, it might not be necessary to perform a completely detailed computation

each time to select a control input and determine the near-future system state. Instead, a controller should be designed to exploit the discrete nature of the control task, along with the predictability of some factors (such as the force of gravity) and thereby simplify the task of regulating the system.

I.C.4 The State-Space approach

One way of reducing the computational density of the Langrangian-equation approach is to simplify expressions for frictional, gravitational, and Coriolis forces. The effects of these forces are quantized or approximated and expressed as vastly simpler terms in the system equations, thus expediting control-force calculations. The abbreviated equations can then be used, in real time, to compute actuator forces with only small reliance on additional corrective feedback [Waters, 73].

Such a strategy is employed in "state-space" control [Raibert, 77]. Actuator forces to counteract changes in effective inertia, coupling torques, and gravitational and Coriolis forces affecting the arm at different points in position-velocity state space are precomputed and stored for regions of the state space. During a movement, actuator forces necessary to produce a desired acceleration are calculated by simply using Newton's Law and combined with appurtenant precomputed forces to drive the manipulator (open loop). The major limitation of this approach is that the accuracy of the precomputed terms is critically dependent on how finely the position-velocity space is partitioned. Implementation becomes a trade-off between precision and the space required for storing a geometric explosion of terms. This then is a different type of tradeoff to be considered: space required to store

quantizations vs accuracy of the quantizations.

I.C.5 The "Logical" Approach

All of the above approaches have shortcomings. Computer control of nonrehearsed movements at a variety of speeds seems to depend on having a mathematical model of the manipulator [Kalman, et al, 69]. It is possible to identify all factors affecting each link of the arm (i.e the principles of mechanics can be exhaustively applied to resolve the effect of every force acting on each link) to derive a complete set of equations which accurately model the dynamics of the arm. But once these equations are obtained, it can be an awesome task to comprehensively analyze them and so design an attractive controller. And once the controller has been designed, it is likely to be impossible to implement it for real-time application [Roderick, 76].

This research has been aimed at analyzing and implementing a computationally economical scheme for the effective control of a mechanical manipulator. In particular, we have investigated the applicability of logical control theory to this task. Logical control theory involves the following general structure including a logical controller and a relation enforcer regulating the system's controlled variables, and a set of condition detectors monitoring the system's observed variables.

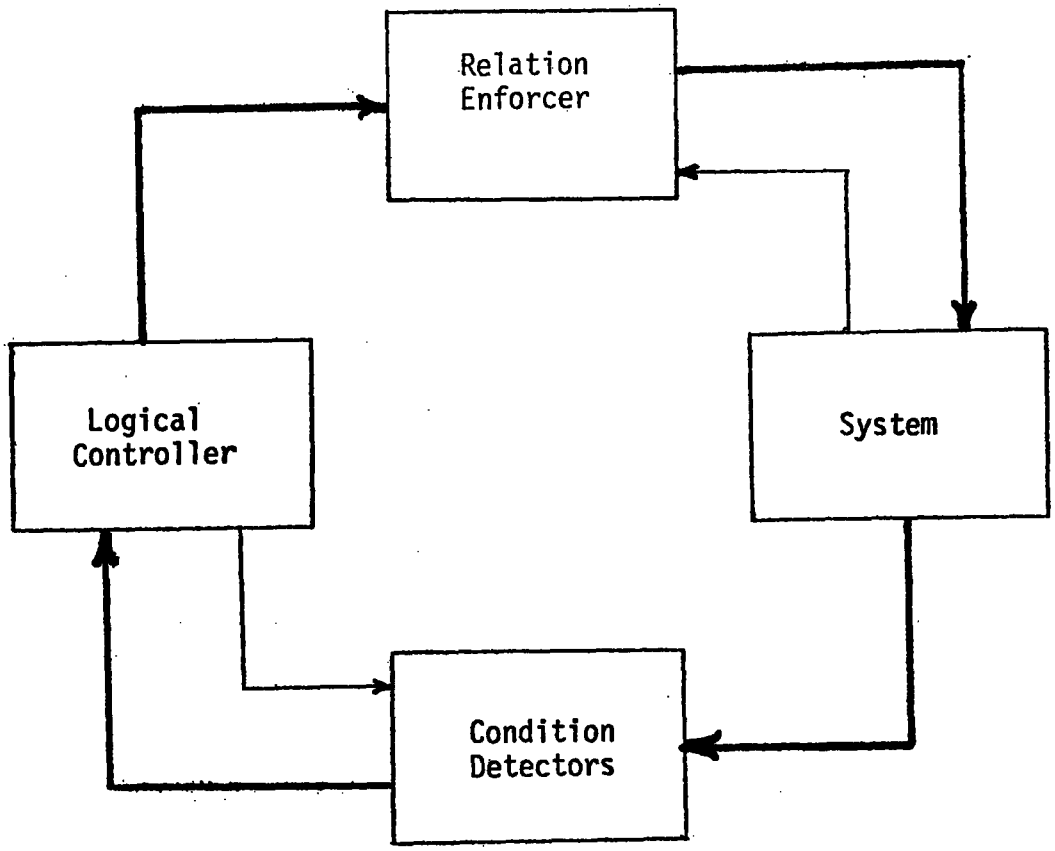


Fig. I.1 Block diagram of a logical controller.

The system is the manipulator being controlled. The observed variables are the position and velocity of each of its joints. The controlled variables are actuator-torques applied by the arm's actuators.

The condition detectors ("demons" in more popular terminology) are logical units that each have one or more (possibly continuous) input signals and each produce one "on" or "off" output signal. The status of the system is revealed by the condition detectors. In practice, the condition detectors sample the observed parameters at finite intervals, and issue prompts (interrupts) to the controller, which is thus a discrete controller.

The logical controller is essentially a finite state machine, or sequential controller. The status of the system, as represented by the condition detectors, affects the future state of the controller. The state of the controller determines its (necessarily discrete) output which directs the relation enforcer. The state of the controller can also affect the condition detectors by modifying the nature of the conditions to be detected.

The relation enforcer serves to establish and maintain the status of each controlled variable, as specified by the logical controller. This device forces the motor-currents -which are the control inputs to the system- to take on the commanded values.

The thermostat is an elementary example of a logical controller. The versatility of control systems employing this general structure on a grander scale is indicated by the properties of telephone systems and of digital computers. Previous efforts to control mechanical limbs with finite state machines are represented primarily by the work of Tomovic and McGhee, who conducted partially successful experiments in pedipulation with

a mechanical quadruped controlled by a logical controller [McGhee, 67].

Sequential controllers, or simple logical controllers have traditionally utilized relays. Recently computer-like programmable controllers have become a popular replacement for relays since they replace hard wiring with software at a reasonable cost. Software problems that characterize other control applications of digital computers are alleviated by omitting many features that make digital computers versatile. However, these controllers retain negative features typical of relay systems: the controller cost depends heavily on the number of variables that can be stored -whether or not these variables are used for the excitation of actuators.

In contrast, digital computers are ideally suited for storing a large number of variables, at little cost. They are also well suited to the quasi-parallel (i.e. reservation time-sliced [Dertouzos, 73]) running of a generous collection of condition detectors, each of which typically involves little computation. Finally, computers provide tremendous flexibility in structuring the finite state machine. Digital computers are thus well suited for the implementation of a logical controller [Hopkins & Quagliata].

This research includes an implementation of a logical controller that regulates a Scheinman Vicarm. The manipulator is scaled to two-thirds the proportions of a human arm, and has six revolute joints. Each joint is powered by a DC torque motor and has a clutch-type brake used to hold the joint stationary when no movement is in progress. A potentiometer and tachometer on each joint provide feedback signals that are proportional to angular postion and velocity. The arm is controlled by means of a PDP 11/45 computer with a twelve bit analog-to-digital and digital-to-analog interface.

II. A Logical Controller

II.A Introduction

In this section we develop a logical controller for a system with a single degree of freedom. For the sake of completeness, the ensuing discussion will begin in the context of an ideal mechanical "arm" with one degree of freedom: a frictionless rotary joint with a vertical axis. We begin with this simple system and then refine the controller several times to take into account friction and loading, oscillations, intermediate-velocity regulation, and gravity.

II.B Initial Approach

The very nature of a logical controller determines the issues that must be considered in designing the controller: we must understand what is the possible behavior of the system, and what conditions must be detected to make control decisions for the system. Also, we must select an effective, but manageable set of (discrete) control inputs for the system. The bang-bang control methodology serves as a reasonable foundation on which to build.

A bang-bang controller is typically designed around a state-space (here, phase plane) trajectory for the system. (Henceforth, "trajectory" refers to the curve that is a plot of the system state in a state-space plane.) The trajectory to the origin defines a curve that bisects the system's state space into two zones. In these, the controller applies one of two opposite input forces. Thus, the trajectory can be used to define a decision boundary for the controller. One of the two equal but opposite control inputs is applied according to which of

the two zones the system is currently in. The switching boundary is chosen to coincide with the state-space trajectory of the system operating under the input that drives its state to the origin. This insures that the system state converges on the origin, i.e. the goal state.

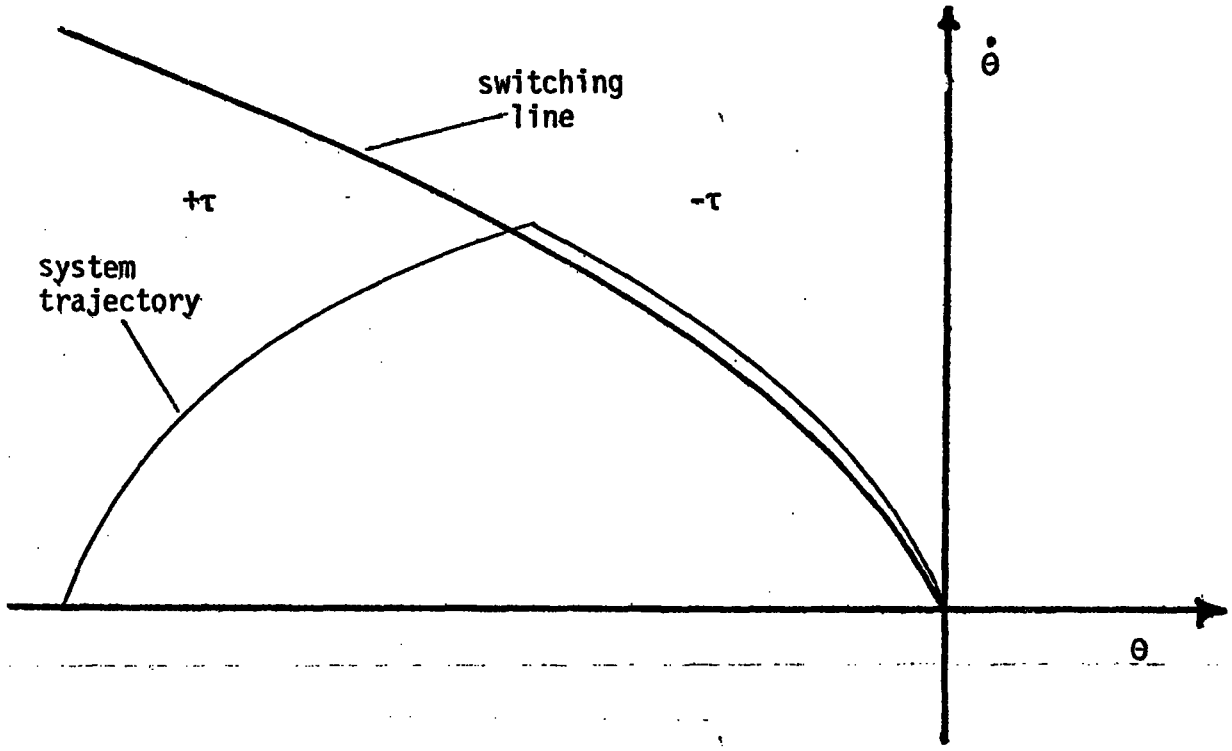


Fig. II.1 System state-space trajectory under a bang-bang controller.

A moment's reflection will reveal that the system trajectories are parabolic. If we were to work in state space with the velocity terms squared, the trajectories would be (approximately) linear. Linear decision boundaries are easiest to use in control-decision calculations, but the squared velocity term decreases the controller accuracy near the origin. So we retain a linearly scaled state space and use piecewise linear approximations of the parabolic trajectories. Additionally, we shall often refer to these parabolic trajectories and their piecewise linear approximations as "lines" even though, strictly speaking, they do not conform to the definition of a line in this Euclidean space.

Because they capture important information about the current and future state of the system, state-space trajectories are used to develop decision criterea for the condition detectors and the control function for the logical controller. State-space trajectories can be acquired from a mathematical model of the system, or from direct observation of the system operating under known inputs. We shall henceforth discuss condition detecting by the logical controller's detectors in terms of crossing state-space decision boundaries. The state-space trajectories for our work were acquired from both direct observation of the system and from a mathematical model of the system. Our model was developed around the Lagrangian derived for the Vicarm. This approach is similar to that taken in [Horn, 75] and, to a lesser extent, in [Horn, 77] and [Paul, 72]. Our model appears in the appendix.

In regulating the single-joint arm, the bang-bang controller applies an input which causes the system state to converge on the switching line. When the state crosses the switching line, the control force is reversed, causing the system state to track the switching line and converge on the origin.

In practice, effects such as friction must be taken into account by modifying the position of the decision line such that the controller switches back and forth between the two control inputs. This causes the system state to oscillate across the switching line as it converges on the origin. Such oscillation is known as "chattering along the switching line," and is generally undesireable because it aggrevates problems with backlash, noise and resonance, and also wastes energy and causes mechanical fatigue.

One way of eliminating the problem of chatter is to introduce hysteresis, or simply a delay in the controller's switching. While hysteresis eliminates the chattering, the delay also introduces a certain amount of error in the switching, and thus in the controller accuracy.

II.C Friction and Loading

Unfortunately, the bang-bang strategy, with or without the switching delay, is not very appropriate for controlling a joint of a mechanical arm. The major problem is the transient positioning of the switching boundary. If determined as discussed in the previous section, the switching line will be shifting and distorting even during a single movement of the arm, because of the changes in inertia and the effects of gravity and friction. (Admittedly, it makes little sense to talk about effects of gravity or changes of inertia for our one-joint arm; this statement is made in anticipation of advancing to more interesting arms.) Of course, it may be possible to compute (before or during a movement) a series of switching lines for each joint to be used during a movement, but such a task would involve computational difficulty similar to that of the modern control theory system model

calculations.

A much simpler approach to coping with the problem of moving decision boundaries is to change the nature of the switching decisions by adding an intermediate state-space zone wherein the control input is zero (i.e. zero torque). The zero-torque zone is bound by positive and negative torque switching lines. The negative torque switching line is positioned to approximate the arm's state-space trajectory when it exhibits minimal responsiveness (e.g. maximal inertia) and is under the controller's negative input. Since this switching line represents a sort of "last chance to decelerate before overshooting" boundary, we must position the line to insure that all external "disturbances" can also be overcome. The positive torque switching line is positioned to define a lower bound on joint velocity and the smoothness of the approach to the origin. The zero torque zone should thus contain the arm's trajectories as it coasts to the origin under zero controller input.

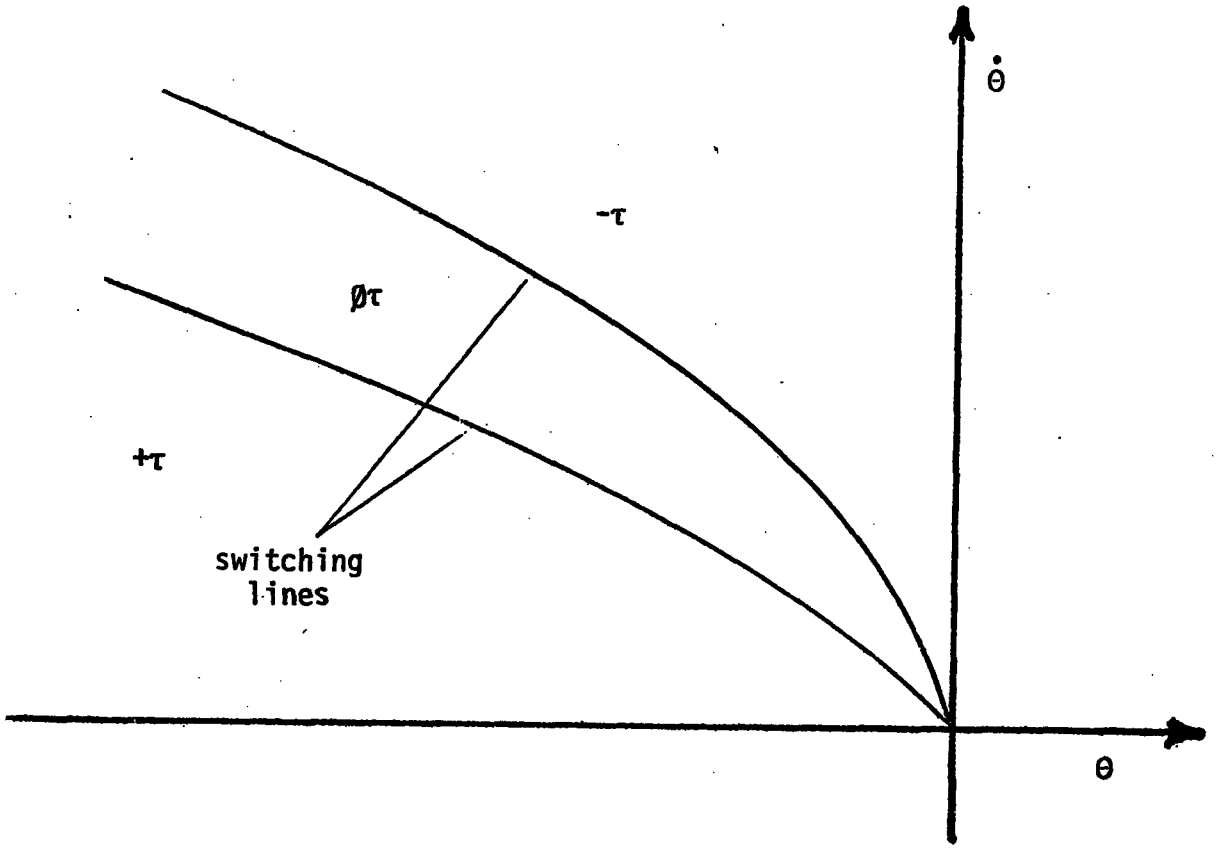


Fig. II.2 System state-space with three control zones.

With three zones, the controller can readily cope with a system that exhibits changing dynamics. But once again there is a problem of chattering along either of the switching lines corresponding to the arm maintaining an extreme configuration (i.e. of maximal or minimal inertia). As in the earlier two-zone controller, the chatter is eliminated by introducing hysteresis; in this case an asymmetric switching delay. Upon crossing into one of the "on" torque zones from the zero torque zone, the switching is immediate, but upon

crossing into the zero torque zone from either of the "on" torque zones the switching is delayed. The duration of this delay must be a function of the joint velocity, set to insure that the system state will move well into, but not across the zero-torque zone before the switching occurs. To enforce this variable delay, the controller now needs a memory. In theory, the memory requires an infinite number of computer bits for real delays. We shall find that this requirement can be reduced considerably.

The asymmetric delay eliminates the chatter and does not introduce any control inaccuracy so long as the system state remains contained within the zero-torque zone during the switching delay. To insure such containment, the correct timing of the switching delay might alternately be realized by introducing a second pair of decision lines inside the existing two. These are used as zero torque switching lines, and the original two lines remain positive and negative torque switching lines. The controller no longer requires a memory to enforce switching delays. However, we would prefer that the system state move well into the zero-torque zone before any switching to a zero torque occurs.

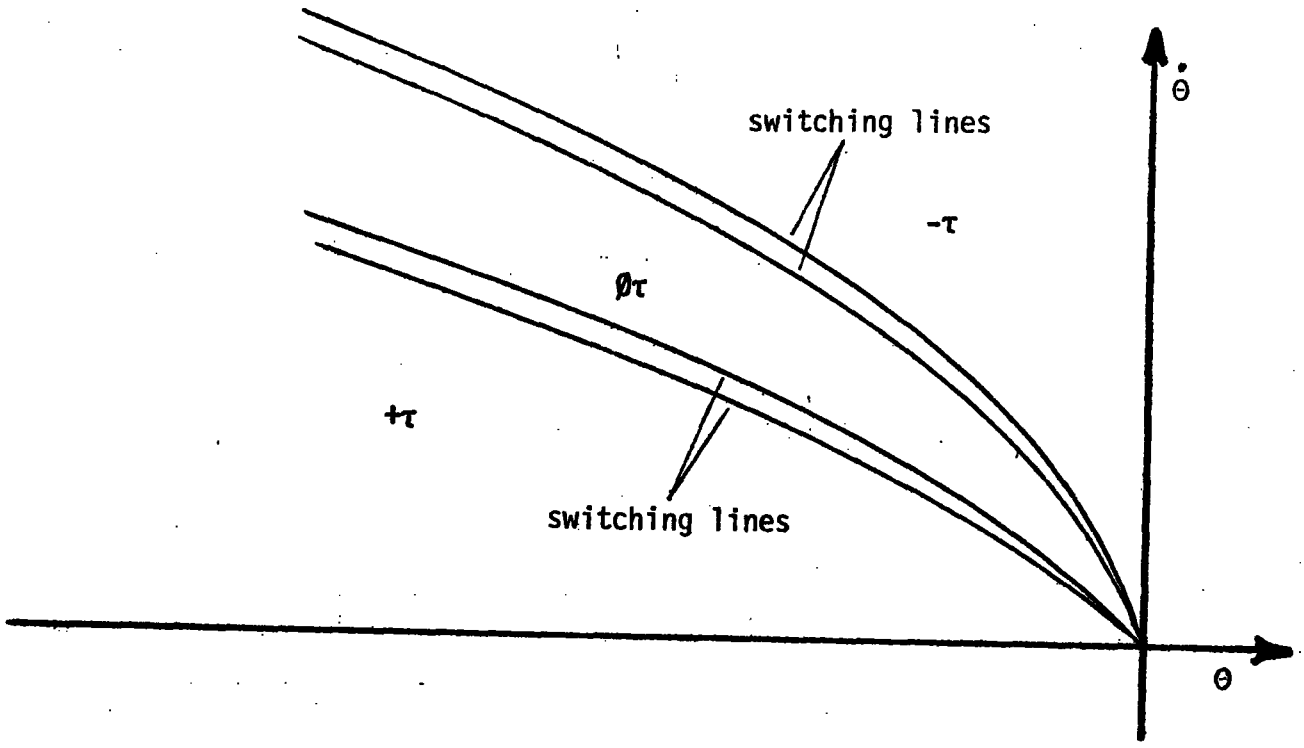


Fig. II.3 System state-space divided into three control zones with delayed switching to eliminate chatter.

This four-line controller is thus improved and simplified by merging the two zero-torque switching lines into one decison line and positioning it in the center of the zero-force zone. The switching criteria for the three-line controller parallels that of the above four-line controller. When the system state crosses into either of the "on" torque zones for the zero-torque zone, the switching is immediate. But upon crossing into the zero-torque zone from either of the "on" torque zones, the switching occurs only when the system state

crosses the center switching line. This insures that the state will always move well into the center of the zero-torque zone.

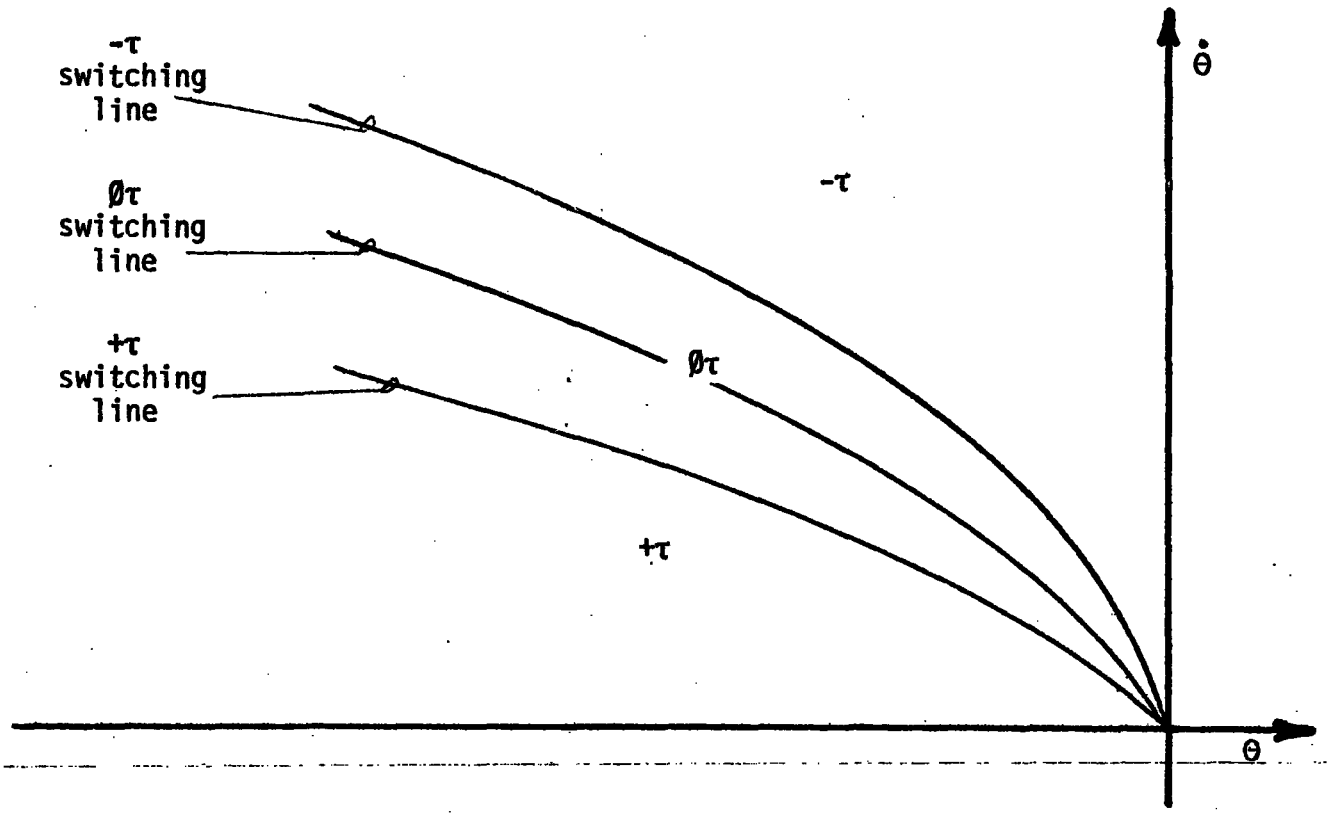


Fig. II.4 System state-space with simplified division into three control zones that takes advantage of system dynamics.

One could interpret the four-decision-line controller as having bisymmetric delays, but the notion of simple switching delays is now inaccurate. Rather, the controller has variable hysteresis that is automatically determined in performing switching decisions with the three decision lines. And in the simplest implementation, this controller still requires no memory; torque switching occurs only when the system state crosses a decision line. (Note that if the system state follows a trajectory that oscillates back and forth across any one decision line as it converges on the origin, the controller will be repeating the decision to apply the same torque. While this is harmless, we may prefer to apply control decisions only when they result in a change in the torque being applied to the system. Operationally, this requires two bits of memory to keep tract of when the system state crosses any decision line for the first time in a row.)

The virtue of employing the three switching boundaries with hysteresis to make control decisions is that it reliably directs the system state toward the center zone and insures that it converges smoothly on the origin without introducing any controller inaccuracies. A sample controlled trajectory is shown below.

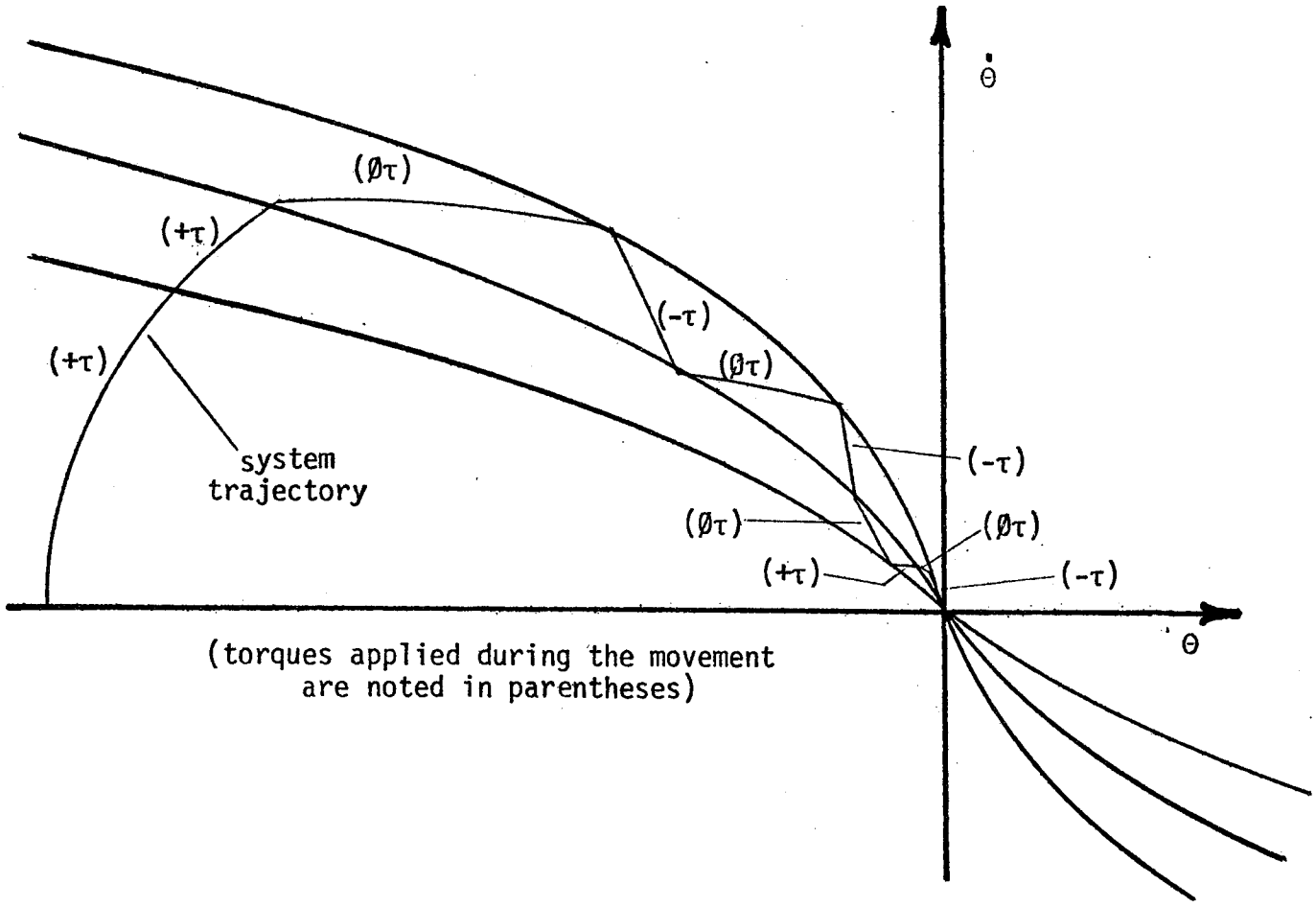


Fig. II.5 Sample system trajectory under logical controller using the three-zone scheme (the same decision boundaries are projected onto the other half of the phase plane). Note that the objective of the controller is to move the system state into the "envelope" defined by the outer two switching lines and then to contain the state within that envelope to insure convergence on the origin.

Let us look in detail at how the position of each switching line is determined. We must first select the magnitude of the torques that the controller will use. This magnitude is a function of the maximum joint velocity at which we intend to operate the arm, or the maximum rate at which we intend to accelerate and decelerate the arm. The chosen torques will be used in obtaining state-space trajectories for the arm.

For the negative-torque switching line, we need the state-space trajectory of the system that converges on the origin when it is operating {1} under the negative torque, {2} in a configuration corresponding to maximal inertia or minimal responsiveness, and {3} under the maximal influence of any external forces that counteract deceleration by the negative torque. This trajectory represents a worst-case response of the system to the deceleration torque, and as such, can be used to insure that we'll always be able to reach the origin without overshooting. Let us call it simply the benchmark trajectory.

The negative-torque switching line is positioned before the benchmark trajectory. "Before" is defined relative to the movement of the system state through the state space in that a system trajectory intercepts the switching line before the benchmark trajectory. (much like a stop sign is before an intersection, and a normal driver arrives at the stop sign before the intersection).

Exactly how far the switching line is before the benchmark trajectory is determined as follows: we must be sure that the controller can always affect a control decision before the system state moves beyond the benchmark trajectory (otherwise we'll overshoot the origin). Note that to affect a control decison, the controller must determine the new control torque and apply that torque to the system. Thus the negative-torque switching

line must be positioned such that the controller can respond to the system state crossing that line before it crosses the benchmark trajectory. Since we know the velocity for any point in the state space, and we select the joint-service interval (i.e. how often the routine that computes the control decision is executed), it is straightforward to compute the minimum distance, calibrated along the state-space position axis, by which the switching line must come before the benchmark trajectory. This interspace is simply the maximum distance that the system state can move, at any given velocity, during the total predetermined and fixed amount of time that we must wait for the controller (i.e. computer) to get around to servicing this joint. Servicing a joint includes making a control decision and then possibly affecting that control decision (i.e. applying a new torque). As an example, if the controller requires a maximum of 0.35 milliseconds to service one joint and there are four joints in the system, then we determine the distance in state space that the system state can move, at any given velocity, in 1.4 milliseconds. The principle is illustrated below.

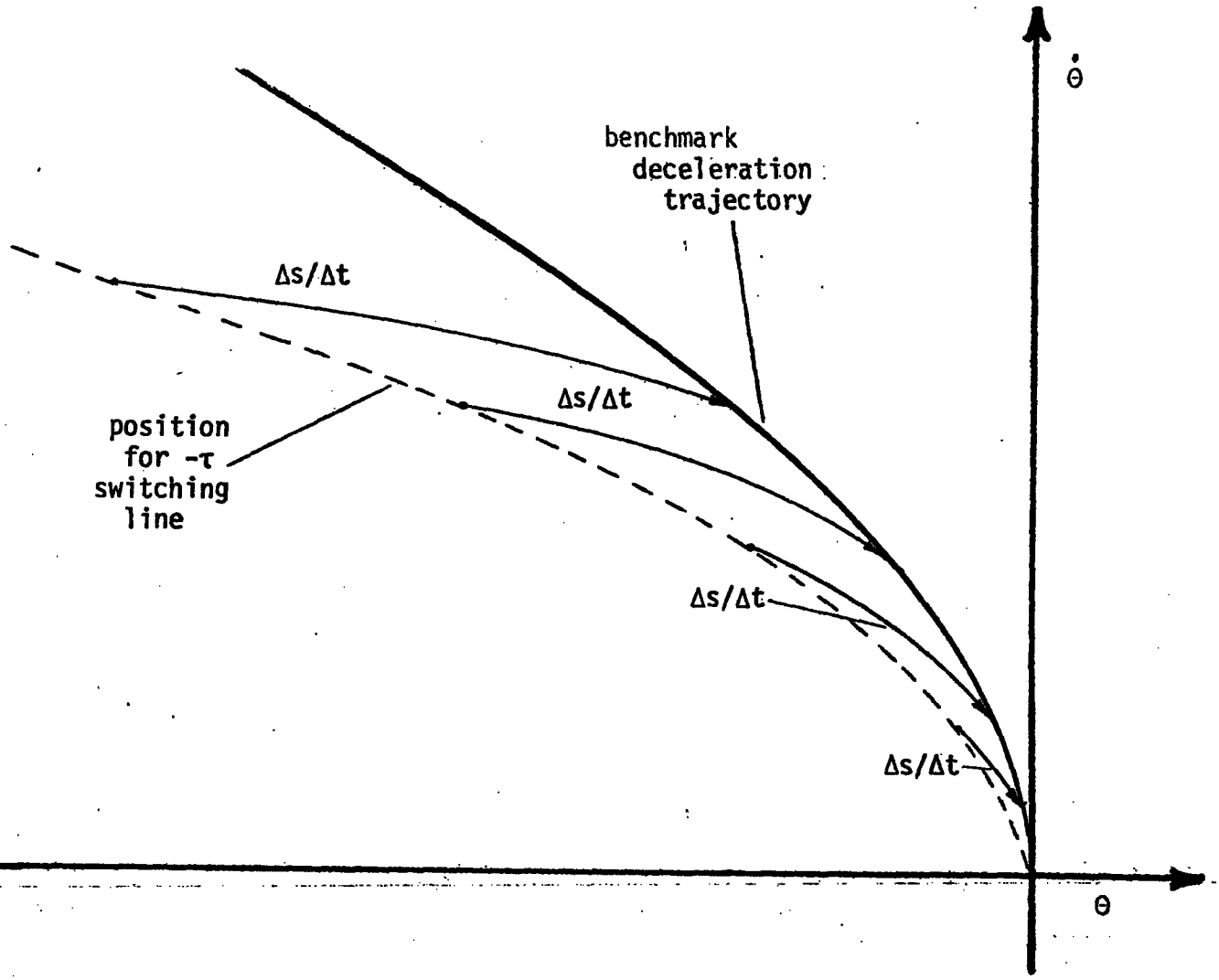


Fig. II.6 Determining the position of the negative-torque switching line.

Δs = change in state (position and velocity) that will occur during service interval plus time taken for controller to affect a control decision, at indicated velocity. Before crossing the negative torque line, the controller is applying a zero torque, so the system trajectories are as shown. Note vertical axis represents joint velocity and horizontal axis represents joint position.

The above diagram shows the minimum distance by which the two lines must be seperated. The important point is that positioning the negative-torque switching line in this manner results in the system state always being driven back towards the center of the envelope defined by the set of switching lines. And our original intent was to contain the system state with this envelope to insure convergence on the origin. Of course the negative-torque switching line can always come even further before the benchmark trajectory.

The positive-torque switching line is configured and positioned to define a lower bound on joint velocity, and the nature of the approach to the origin (i.e. a lower bound on deceleration). Observe in the two diagrams below two different joint-velocity minimums have been defined by changing the position of the positive-torque switching line.

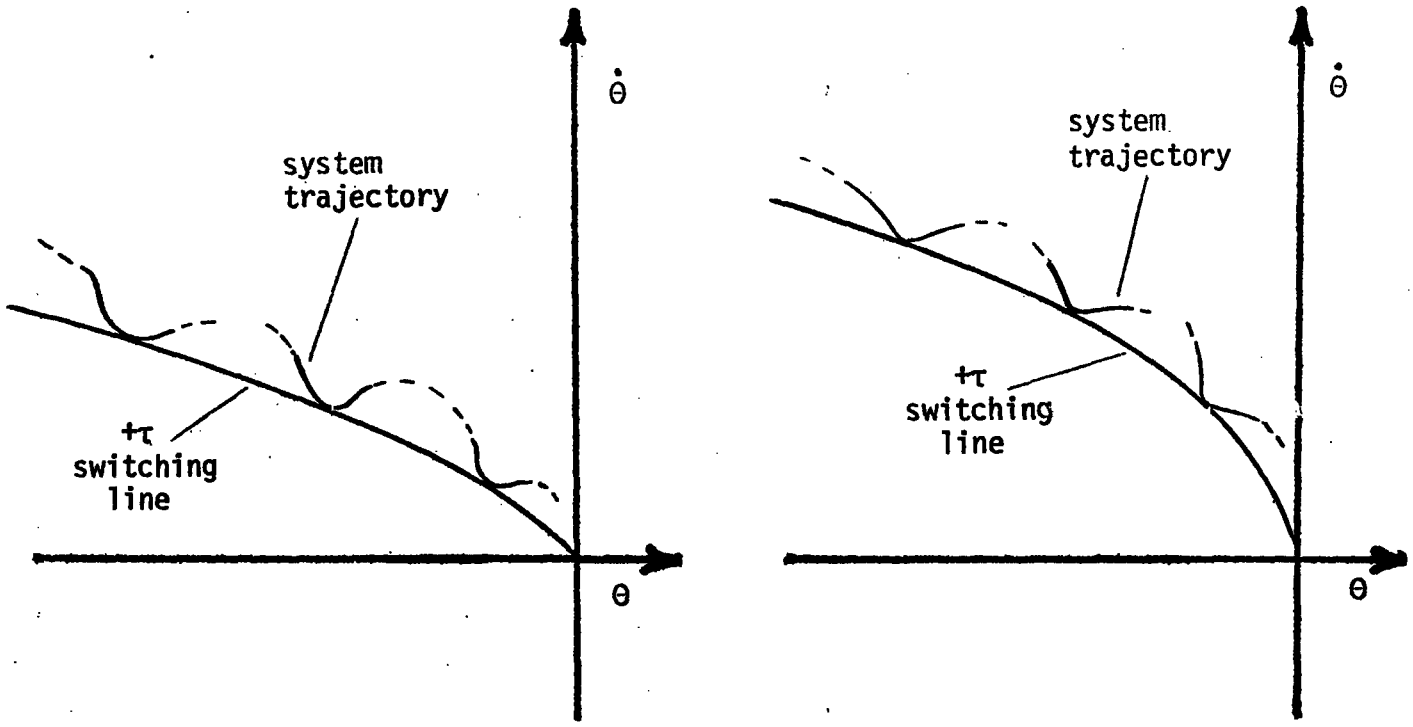


Fig. II.7 Positive-torque switching line defines a lower bound on the joint velocity.

Likewise, in the two diagrams below two different approaches to the origin (i.e. lower bounds on deceleration) have been defined.

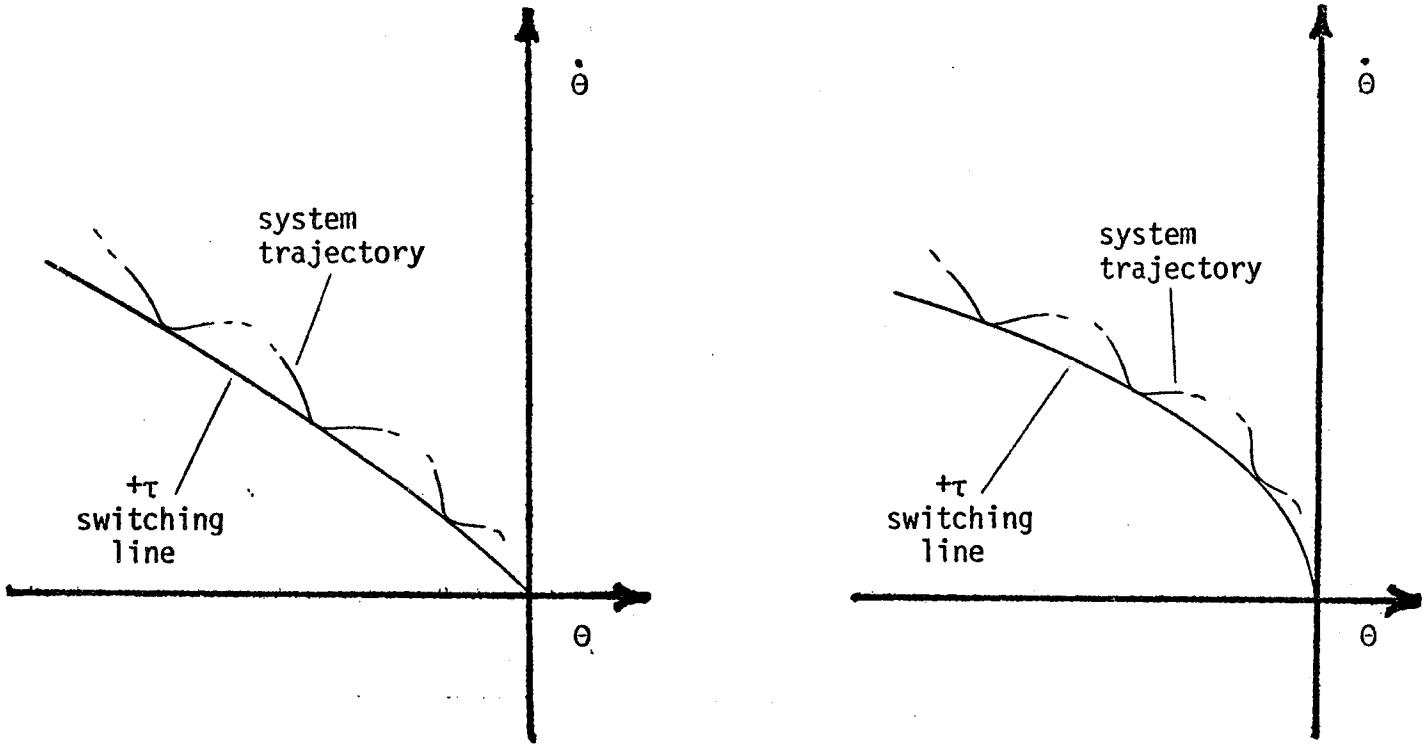


Fig. II.8 Positive-torque switching line defines a lower bound on joint deceleration into the origin.

As with the negative-torque switching line, we must insure that when the system state crosses the positive-torque decision line the controller can affect a control decision before the system exceeds some boundary condition. For the positive-torque switching line, this means positioning it a certain distance above our lower bound on joint velocity, as illustrated below.

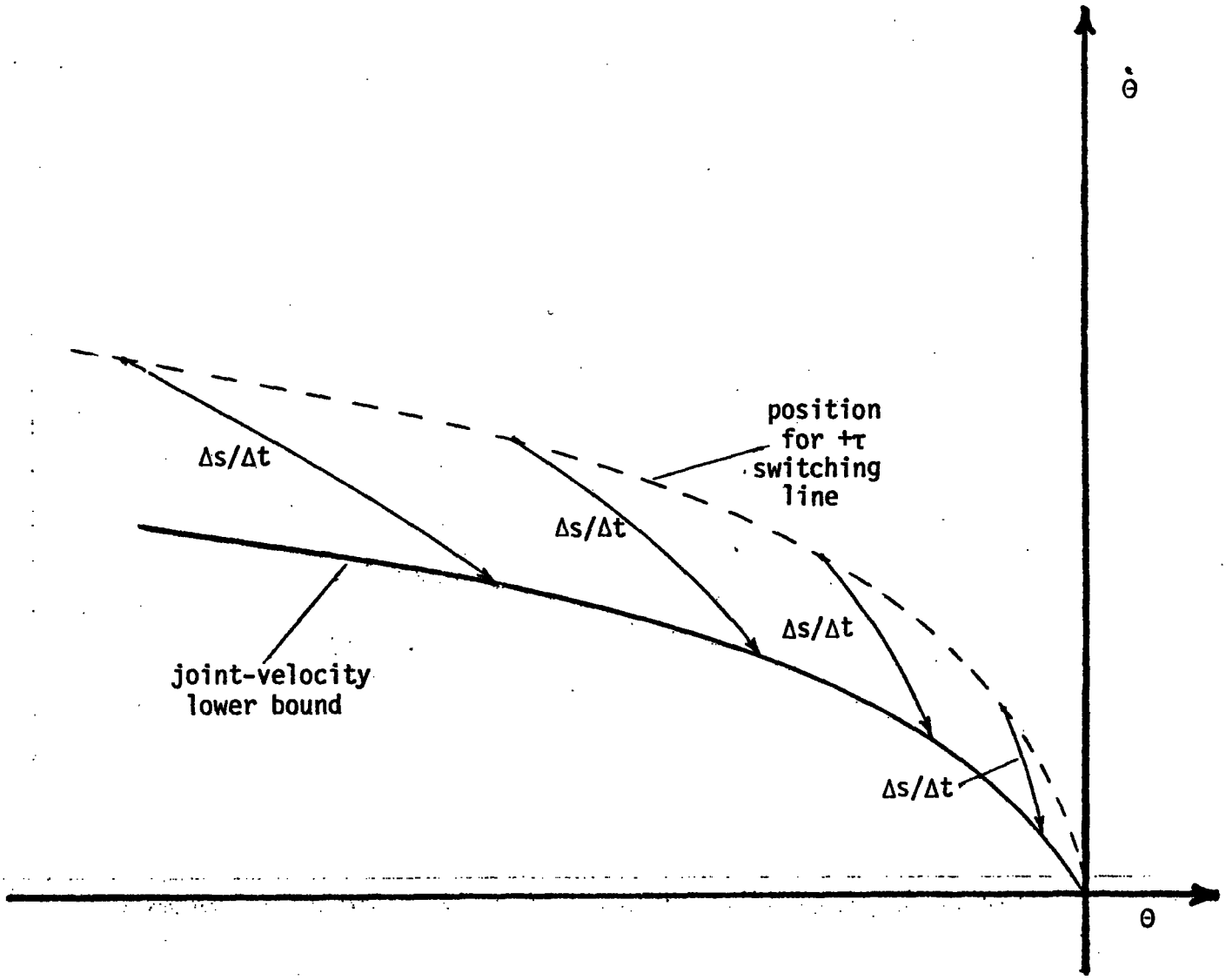


Fig. II.9 Determining the position of the positive-torque switching line.

Δs = change in state that will occur during service interval plus time taken for controller to affect a control decision, at indicated velocity.

In practice, the positioning of the positive-torque switching line must be coordinated with the positioning of the other two switching lines when constraining joint velocities. This will be discussed in section II.5. Also, the approach to the origin must be such that the decceleration torque is capable of containing the system trajectory within the envelope (i.e. the positive-torque switching line must come before the negative-torque switching line and thus its derivative must be everywhere less than that of the negative-torque line).

The zero-torque switching line can simply be positioned halfway between the other two lines. This is acceptable if the system state has an equal tendency to gravitate from the zero torque line towards either of the two outer switching lines and move outside the envelope. Alternatively, we can obtain a state-space trajectory of the system when it is exhibiting some average intertia and coasting to the origin under zero controller input. The zero-torque switching line is then positioned to coincide with this system trajectory. This contributes directly to the controller's objective of causing the system state to converge on the origin since once the zero torque switching line is intercepted, the system state will move roughly along that line, coasting toward the origin.

Condition-detecting amounts to performing some trivial and thus fast calculations to determine on which side of a switching boundary the current system state occurs.w And the finite state machine that makes up the logical controller is basically a three state automaton:

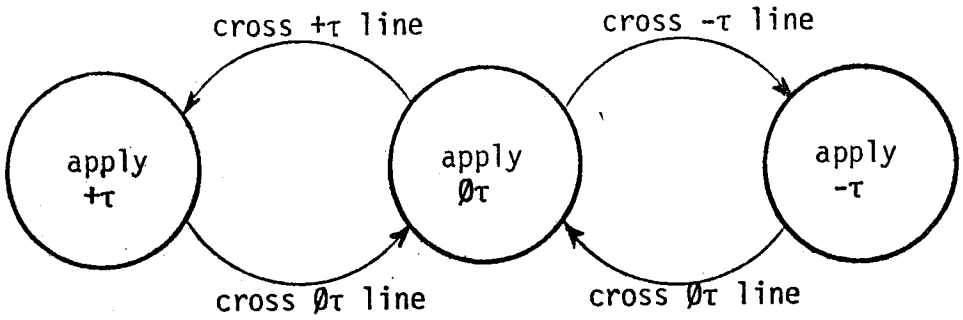


Fig. II.10 Finite-state machine for the three-torque logical controller.

II.D Overcoming Oscillations

This control methodology was used to control a single degree of freedom arm, with one modification. Heretofore we have ignored the issue of stability. Any system controlled by a bang-bang style controller can enter into a limit cycle around the origin. Even with our zero-torque zone, there exists the possibility of oscillations, because the positive and negative torque switching lines converge at the origin. The obvious way to reduce any oscillation is to employ reduced torque levels in the vicinity of the origin. This is reasonable since the joint velocity must be reduced upon approaching the origin. The only consideration is that the torques remain strong enough to drive the system completely to the origin. (In practice, the torque levels were halved when the system state was within $\pm$0.10

radians and ±0.15 radians per second of the origin.)

The next step to eliminate limit cycling is to separate the positive and negative switching lines (at the origin) and have them intersect the abscissa of the state-space graph on opposite sides of the origin. This widens the zero-torque zone around the origin. Slight displacements of these intersections (on the order of ±0.03 radians, in our experience) eliminate or greatly reduce the amplitude of the limit cycles. But displacing the switching lines also introduces a corresponding inaccuracy in the controller; increased separation of the switching lines may thus not always be the best way to quench limit cycles. Therefore the final step in eliminating limit cycles is to define a dead zone around the origin that contains the minor limit cycles still occurring with the diminished torques and split switching boundaries. In practice, this dead zone had dimensions on the order of ±0.005 radians by ±0.06 radians per second. The state-space decision boundaries are now:

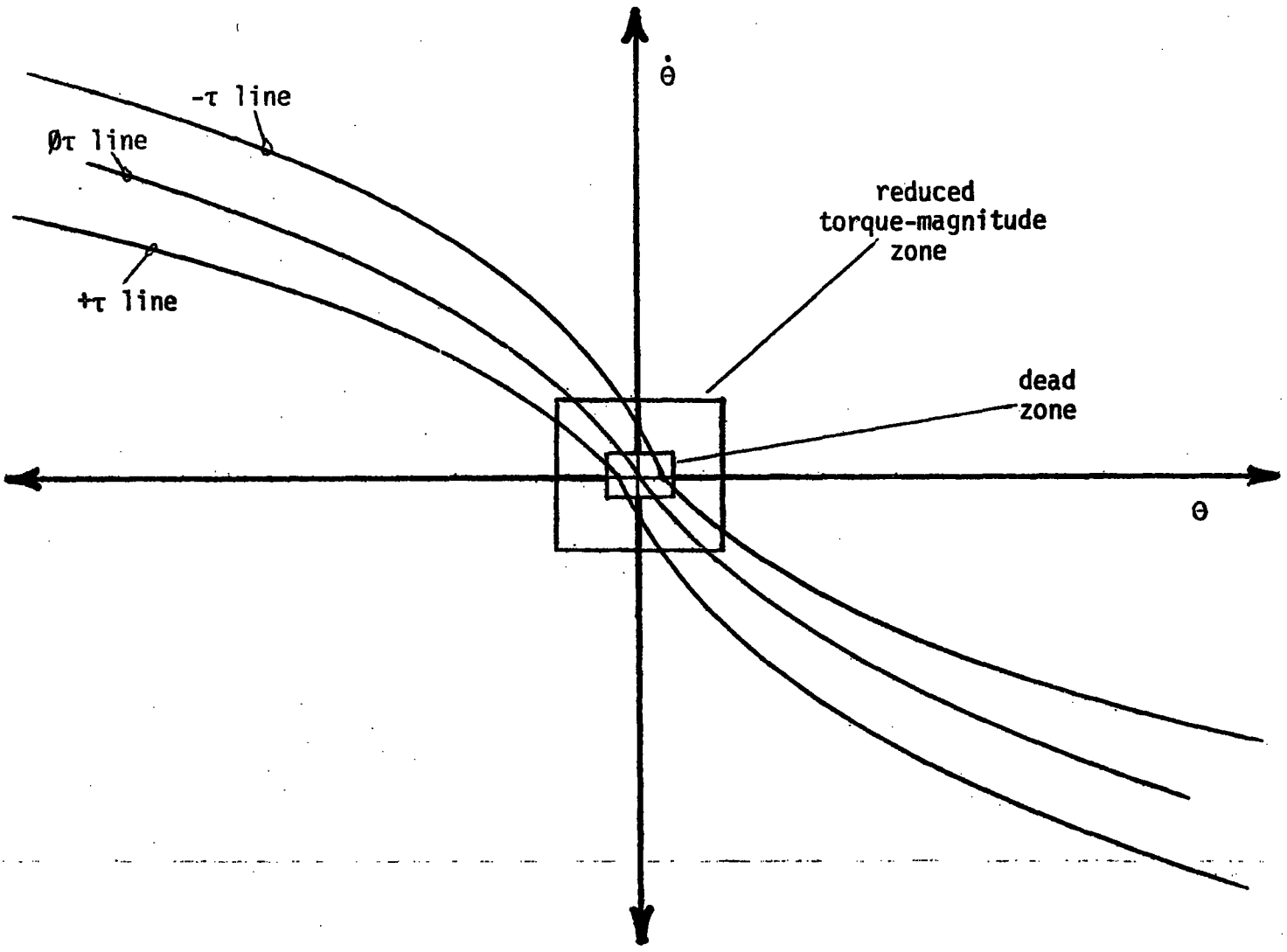


Fig. II.11 Complete set of switching lines for three-torque logical controller.

II.E Constraining Intermediate Velocities

Most applications of manipulators require more than the ability to control the final position achieved at the end of each movement. In particular, it is often desirable or necessary to be able to regulate the intermediate velocities of the joints during a movement. Contrast the task of carrying a glass of water with that of pounding a nail. Our logical control strategy fulfills this requirement, without modification. The same three decision lines can be extended in state space to define a trapezoidal, or whatever, trajectory: they are positioned to bound the desired velocity with some allowable deviation (to within $\pm$0.15 radians per second) along the trajectory. The condition detecting, switching criteria, and finite state machine remain exactly the same!

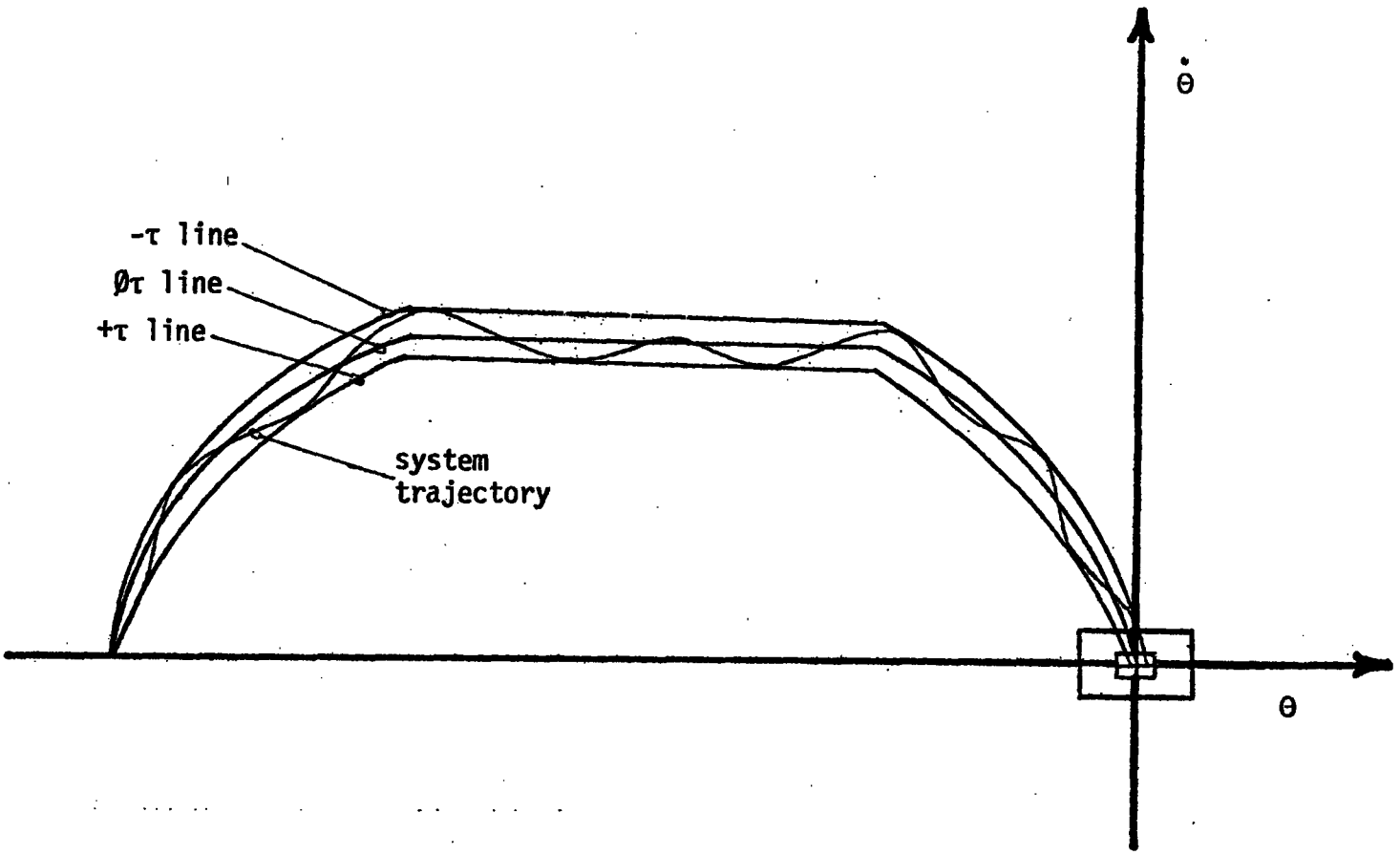


Fig. II.12 Using the three control-torque decision lines to regulate intermediate velocities. Note that the separation of the switching lines defines the allowable range of deviation in the intermediate velocity and thus the smoothness of the movement.

II.F Introducing Gravity

Let us now rotate the axis of the joint 90 degrees such that the link moves in a vertical plane, and consider the effects of gravity. The link is no longer uniformly responsive to the controller's discrete forces over the entire state space. Clearly, the link will accelerate quite differently, under the same input torque, when it is straight out, compared to when it is straight up. And when straight out, the link will be more responsive to a given torque that is forcing it downward than to the same torque forcing it upward. But this is a minor complication since gravity is constant and predictable.

To accomodate these effects of gravity (i.e. to cope with variations in responsiveness of the arm that are a function of configuration), the controller must undergo a refinement. The joint's state space graph, with which the controller makes decisions, is modified to take into account the arm's responsiveness. This is done by partitioning the state space into regions; throughout any one region, the responsiveness of the arm is taken to be the same.

We do not simply divide the 360-degree span of the joint-position axis into several equal-sized regions since the effects of gravity are not in exact proportion to the link's angle of inclination. Rather, the gravitational force is a function of the cosine of the angle of inclination (measured from a horizontal plane). As the link rotates through any 180 degree arc, this force will range over 0 to -1 units: the force is 0 when the arm is straight up or straight down, and the force is -1 when the arm is straight out (arbitrary units normalized for convenience).

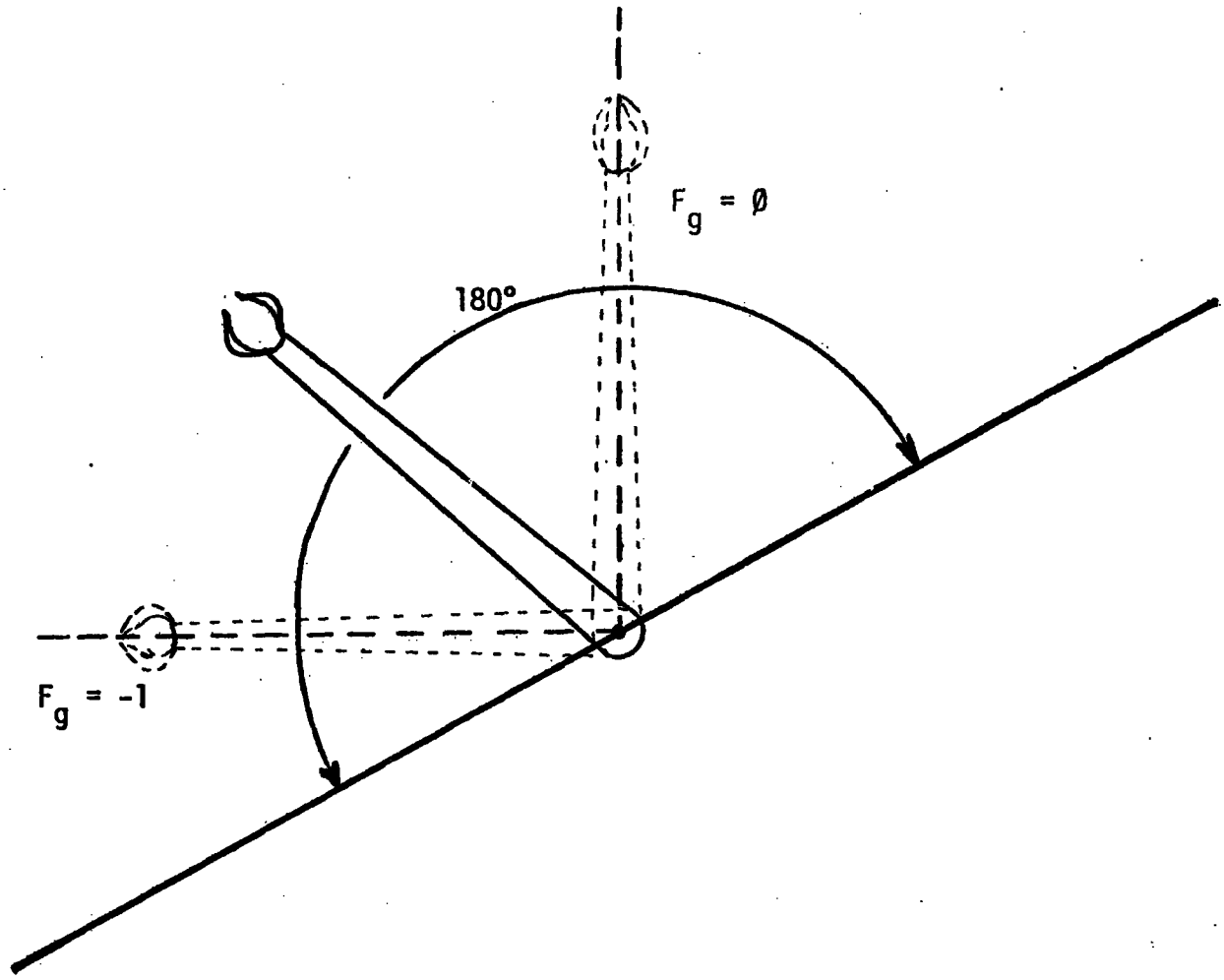


Fig II.13 Range of gravitational force acting on a limb over any 180 degree arc.

Thus we first partition the range of this gravitational force exected on the line into several equal-size divisions. These divisions are then mapped onto the joint-position axis using the arccos function.

There is a tradeoff to consider when determining into how many regions the

position axis should be divided. Specifically, the accuracy of the compensation for gravity (or consistency of the responsiveness) within each region vs the amount of space required to store sets of torques unique to each region. But note that the link's dynamics are identical above and below a horizonal plane containing the axis of the rotary joint. Thus we must develop and store sets of torques only for a chosen number of n regions spanning some 180 degree arc. These n regions and sets of torques give us the operational equivalent of 2n regions across the entire joint-position axis. For demonstration purposes, we chose n = 2, or 4 regions.

One might expect that this partitioning would require dozens of regions. Yet we found that only four regions allows the controller to perform very well (intermediate velocities could be consistently constrained within $\pm$0.10 radians per second). Doubtless, a few more regions would yield an improvement in controller accuracy. Nevertheless, our four-region approach demonstrates the principle. The partitioning was done as illustrated:

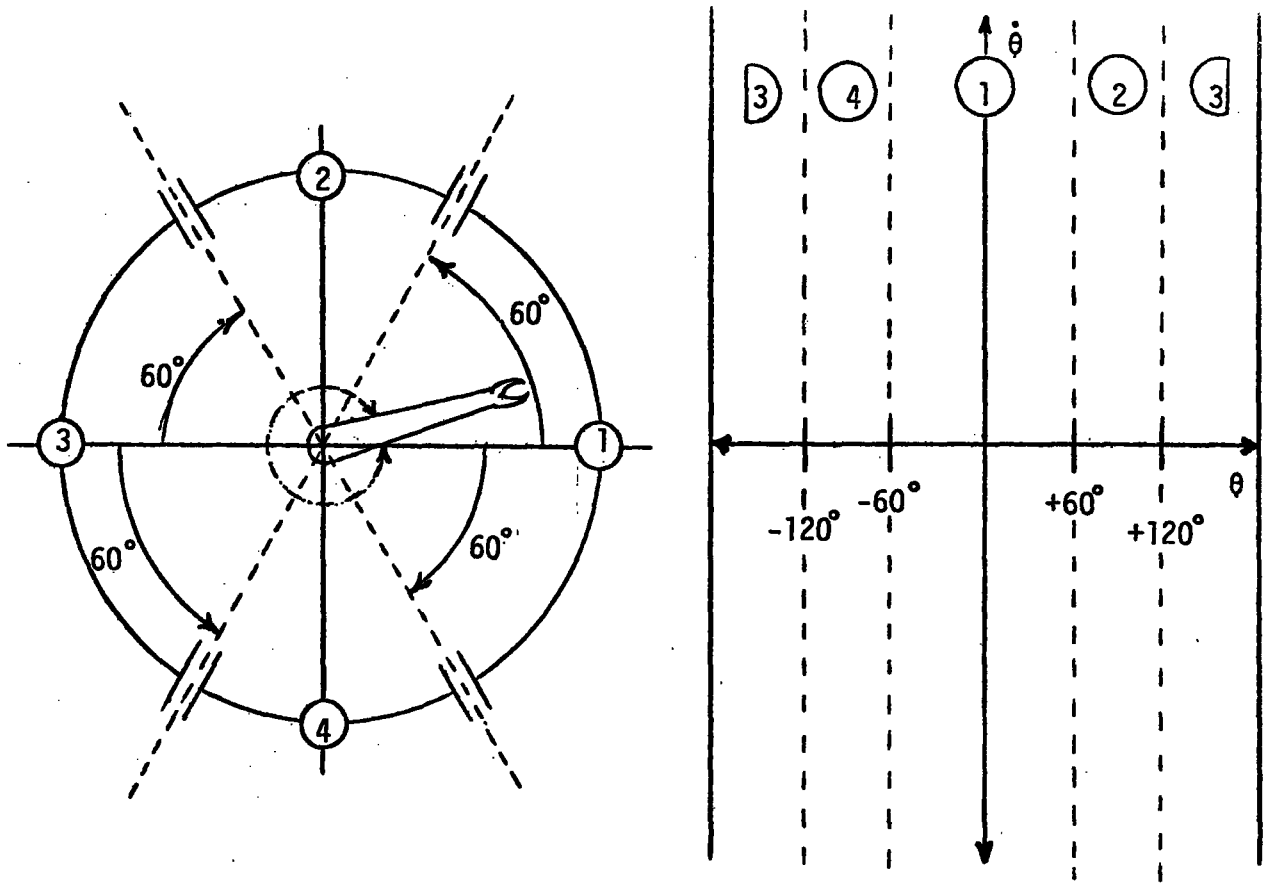


Fig. II.14 Partitioning of system state-space on the basis of partitioning the range of forces affecting the system. Note that regions 1 & 3 are equivalent but opposite, as are regions 2 & 4.

To employ the existing controller structure, we must develop a set of torques for each state-space region that compensate for the effects of gravity in that region. Within each of these regions, the control torques are selected to overcome for the maximum gravitational force that can be experienced in the region, and thus allow the controller to maintain any desired velocity.

The controller uses these state-space region-specific sets of discrete currents which are selected to compensate for the effects of gravity, along with region-specific sets of decision lines that take into account the system's behavior while operating under these sets of currents. When the controller is making a control decision for the joint, it uses only the one set of torques and decision boundaries that is applicable for the region that contains the current state of the joint.

The new state-space trajectories that define decision boundaries are readily developed by modifying the system model to include the force of gravity. (Alternatively, the system is observed while operating with its axis now in a horizontal plane.) Within each state-space region, the torques that will overcome gravity and sustain a desired velocity or acceleration anywhere in that region must first be determined. Then the arm's state-space trajectories in the region can be obtained by modelling or operating the arm in the region under the torques applicable to that region. Finally, the positions of the decision boundaries are established as discussed earlier. This process of selecting driving torques and then developing a set of switching boundaries must be done for each region of the state space.

Note that it is not necessary that decision lines be perfectly continuous across region boundaries. A control decision that is precipitated by the system state suddenly being

in a different torque zone upon crossing a region boundary (e.g. going from positive torque zone to zero torque zone when changing regions) is no different from a decision caused by the system state moving to a different torque zone within the same region. However, if the system state remains in the same type of torque zone upon crossing a region boundary (i.e. no decision to change the control torque is precipitated) the control torque could be of the wrong magnitude. Upon crossing a region boundary it is thus necessary for the controller to reevaluate the joint's control torque to the magnitude applicable in the just-entered region.

Once again, the implemented controller regulates the joint with essentially the same performance statistics, in the same amount of CPU time. We need merely add condition detectors to check for which region the system state is currently in, and control logic to select the set of corresponding control torques and decision lines. Operationally, this amounts to simply computing which side of a region boundary the system state is on, and then using the indicated set of torques and switching lines.

II.G Summary

The bang-bang controller has been developed into a framework for a logical controller. It provides a complete strategy for controlling a single degree of freedom. In applying it to a single arm joint, we were able to constrain intermediate velocities by $\pm$0.10 radians per second, and achieve final positioning accuracy to within $\pm$0.01 radians (approaching the resolution limit of our A-to-D converters). There is no detectable deterioration in performance until the sample/service period approaches 4 ms. Yet the entire calculations for one control decision and D-to-A outputting of a control torque takes less than

0.2 ms on the PDP 11/45 computer used in our implementation.

III. A Multi-Link Arm

III.A Introduction

In this section the system is extended to an arm with a plurality of coupled links. We begin with the controller developed in the previous section and find that it cannot handle the system's complex dynamics arising from the interaction of the links. The logical controller is then refined to manage the multi-link arm. We will observe that with our approach, the problems and their solutions are the same for a two-link arm as for a three-link arm, etc. The discussion is therefore in terms of controlling a general multi-link manipulator.

III.B Interacting Joints

The limitations of the controller developed in section II become manifest when we apply it to a multi-link manipulator. The dynamics resulting from the interaction of two links invalidates the modelling that goes into designing decision boundaries for a joint. Since the links do not operate independently, the state-space trajectory for each of the joints is not necessarily parabolic. The significance of this is that coupling torques, Coriolis forces and varying configurations (which we shall collectively call interaction forces) contribute to irregular dynamics that can no longer be effectively regulated by the controller. A typical set of trajectories is presented below.

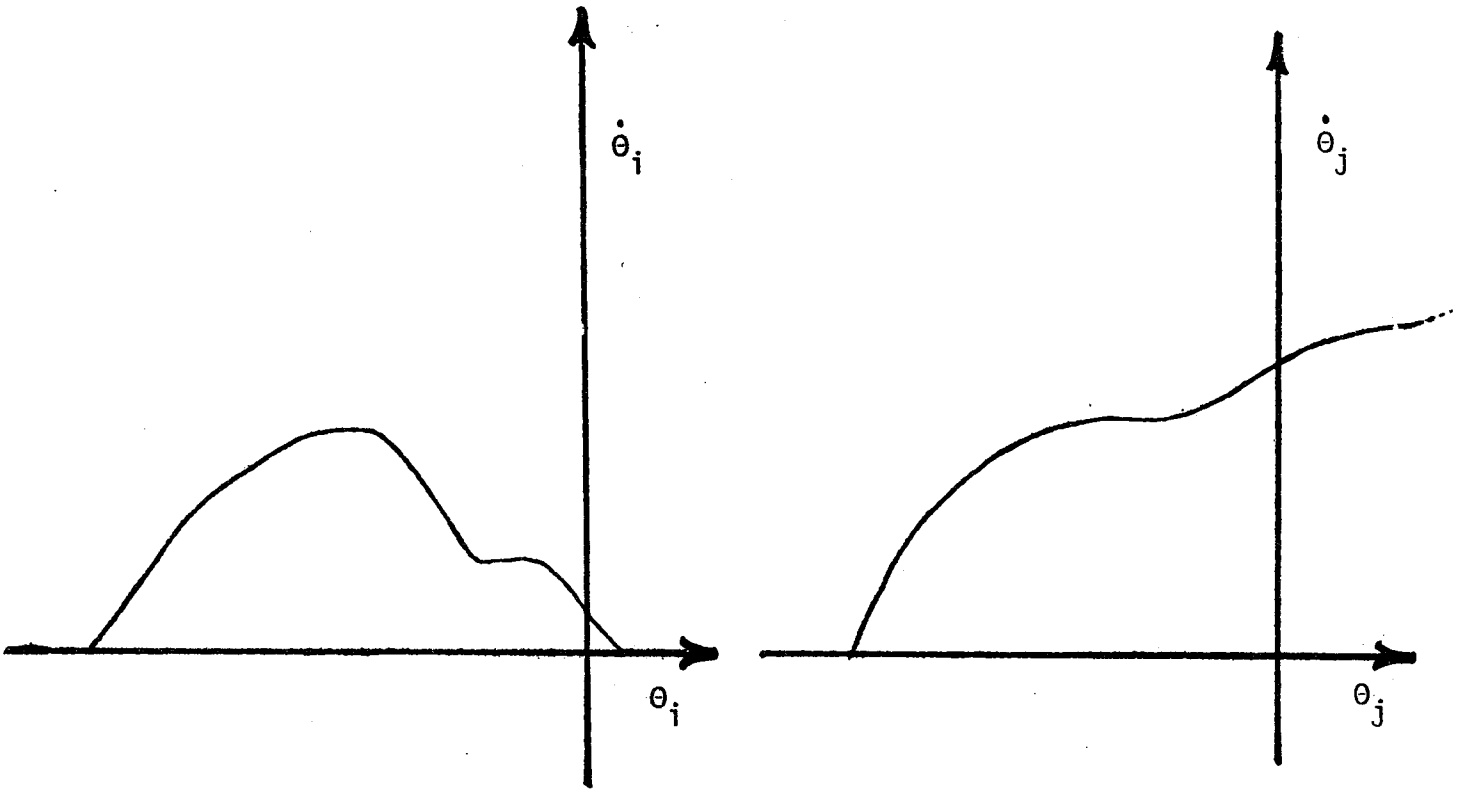


Fig. III.1 Representative system state-space trajectories for two interacting degrees of freedom.

To cope with such dynamics and still be able to maintain desired velocities, the controller must be capable of selectively applying larger torques. It is computationally prohibitive to calculate specific appropriate torques for handling every "disturbance" (i.e. interaction forces) that occurs during a movement. Instead, the controller should have a set of predetermined discrete torques that can be applied in the same manner as the regular control torques.

The most elegant approach to this problem would be to have sets of torques that are applicable in particular hyperregions of the multi-joint state space. These hyperregions and their region-specific torques would be similar to those developed for the single-joint arm affected by gravity (section II.F): sets of currents and associated decision boundaries would be selected in accordance with the responsiveness and interaction of the system in each hyperregion. This is a complex undertaking, something on the order of Raibert's state-space controller, except that for our controller the complete operating torques, rather than just interaction-force correcting torques, must be determined. We shall elaborate on this idea later.

A somewhat simpler approach has been to add, for each joint, two more control torques to the already existing structure. These two torques, "high-positive" and "high-negative" must be selected to regulate (compensate for) any disturbance that may arise from dynamic interaction that the regular torques were not designed to handle.

Determining the magnitude of these torques is facilitated by means of the system model. For each link, we must find (analytically or empirically) the maximum positive and negative interaction forces that can act on the link. The high-level torques are then chosen to compensate for the deleterious effects of these greatest possible interaction force and thus maintain a desired velocity or acceleration.

An important point is that all interacting links must be analyzed simultaneously. It is incorrect to examine the sytem with all interacting links being driven by their normal-level torques because the interaction forces will be greater when the links are being driven by their high-level torques. Thus the high-level torques for all interacting links must

be determined when {1} each joint in the system is operating at the desired velocity, while {2} the arm is in a maximally interactive configuration and {3} is being driven with these same high-level torques.

Once a complete set of torques for a joint has been determined, the decision boundaries are developed in the same manner as were those for the earlier three-line controller. We must acquire (by modelling or observation) benchmark state-space trajectories for the link while it is operating under the high & normal negative torque levels. Note that when the trajectories for the high negative torque is obtained the link must be viewed as though it were under the maximum influence of interaction forces arising when the system is in the current state.

These system trajectories guide the positioning of the decision boundaries, as before. The normal-negative-torque decision line is positioned relative to the normal-negative-torque benchmark trajectory to insure that the controller can normally drive the system to the origin when it is in a configuration corresponding to maximal intertia. The high-negative-torque decision line is intended to correct abberations due to interaction forces, and so ends up being positioned roughly parallel to and outside (beyond) the normal negative torque line. The high-negative-torque benchmark trajectory defines the absolute-worst-case deceleration trajectory. The spacing between the two lines is fixed by how much variation in a desired velocity or deviation from a desired trajectory is tolerable; the limiting case is when these two lines are coincident. The joint-service interval, system velocity and control-response time must be considered when positioning the lines, as illustrated below.

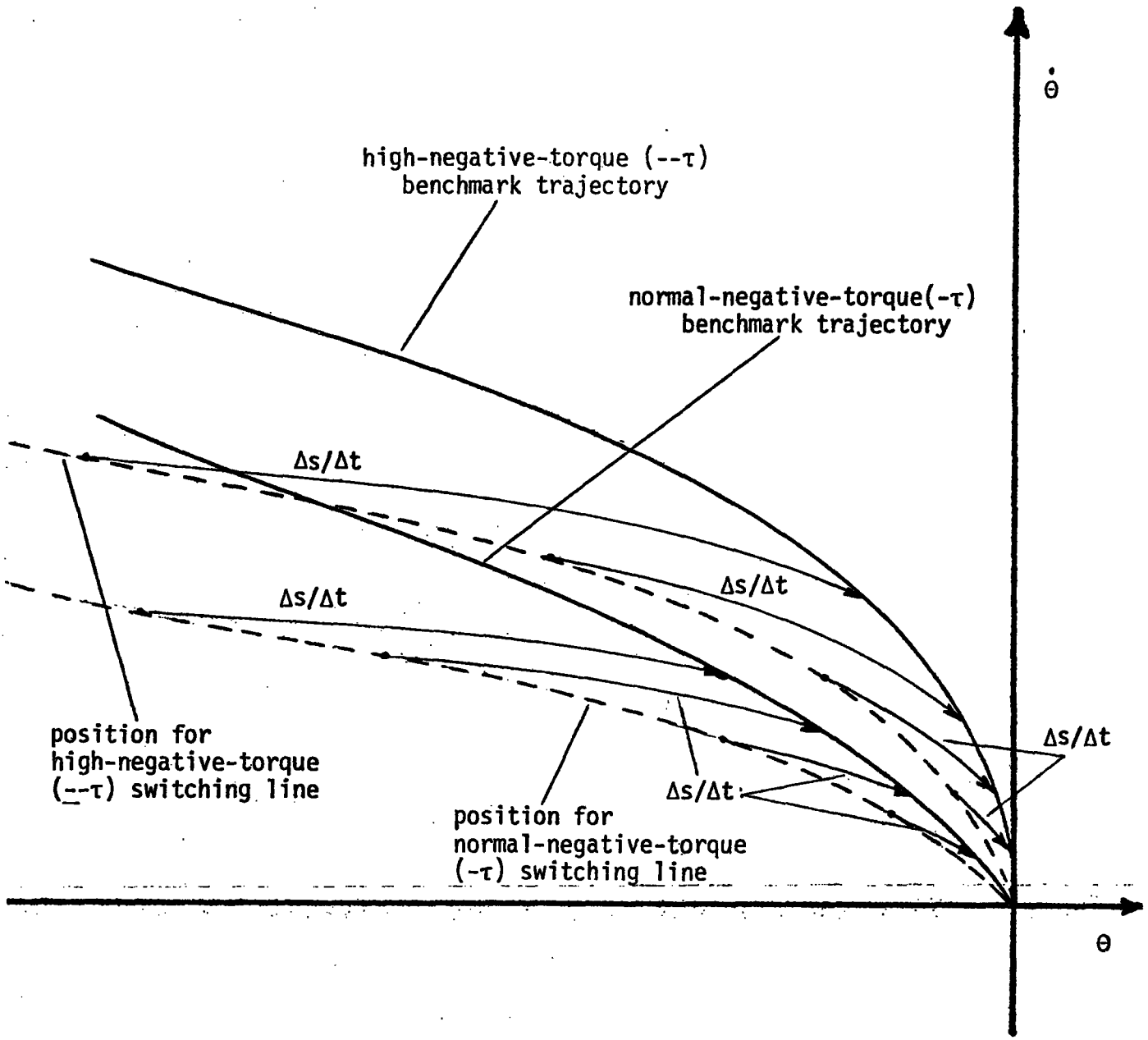


Fig. III.2 Determining the positions of the negative-torque switching lines.

Δs = change in system state (position and velocity) that will occur during service interval plus time taken for controller to affect a control decision, at indicated velocity. Vertical axis represents joint velocity and horizontal axis represents joint position.

The normal-positive-torque switching line is positioned to define a lower bound on joint velocity and on joint deceleration into the origin. The high-positive-torque is intended to correct for interaction disturbances and so is positioned to closely parallel the normal-torque-line, in accordance with tolerable deviations of the system state. The high-positive-torque switching line defines an absolute lower bound on joint velocity and deceleration. This is illustrated below.

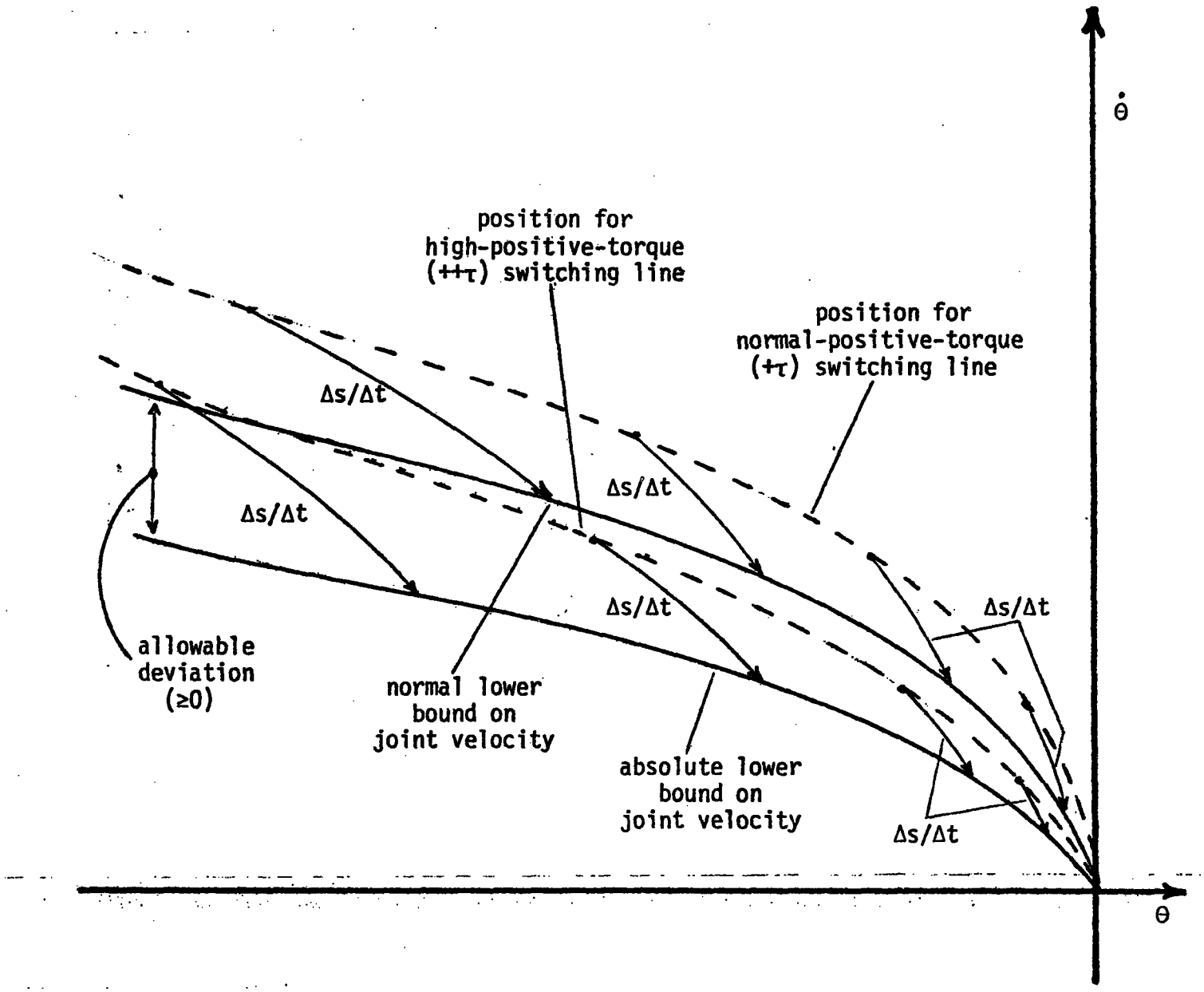


Fig. III.3 Determining the positions of the positive-torque switching lines.

Δs = change in system state (position and velocity) that will occur during service interval plus time taken for controller to affect a control decision, at indicated velocity. Vertical axis represents joint velocity and horizontal axis represents joint position.

The zero torque decision line should approximate the state-space trajectory of the arm as it coasts to the origin when in configurations exhibiting medial intertia. Alternatively, if gravity affects the link, the zero torque line should be positioned between the normal positive and negative torque decision lines such that the system state has an approximately equal tendency to diverge from the zero torque line to either of the normal torque lines. The complete set of switching lines for an individual joint now appears as:

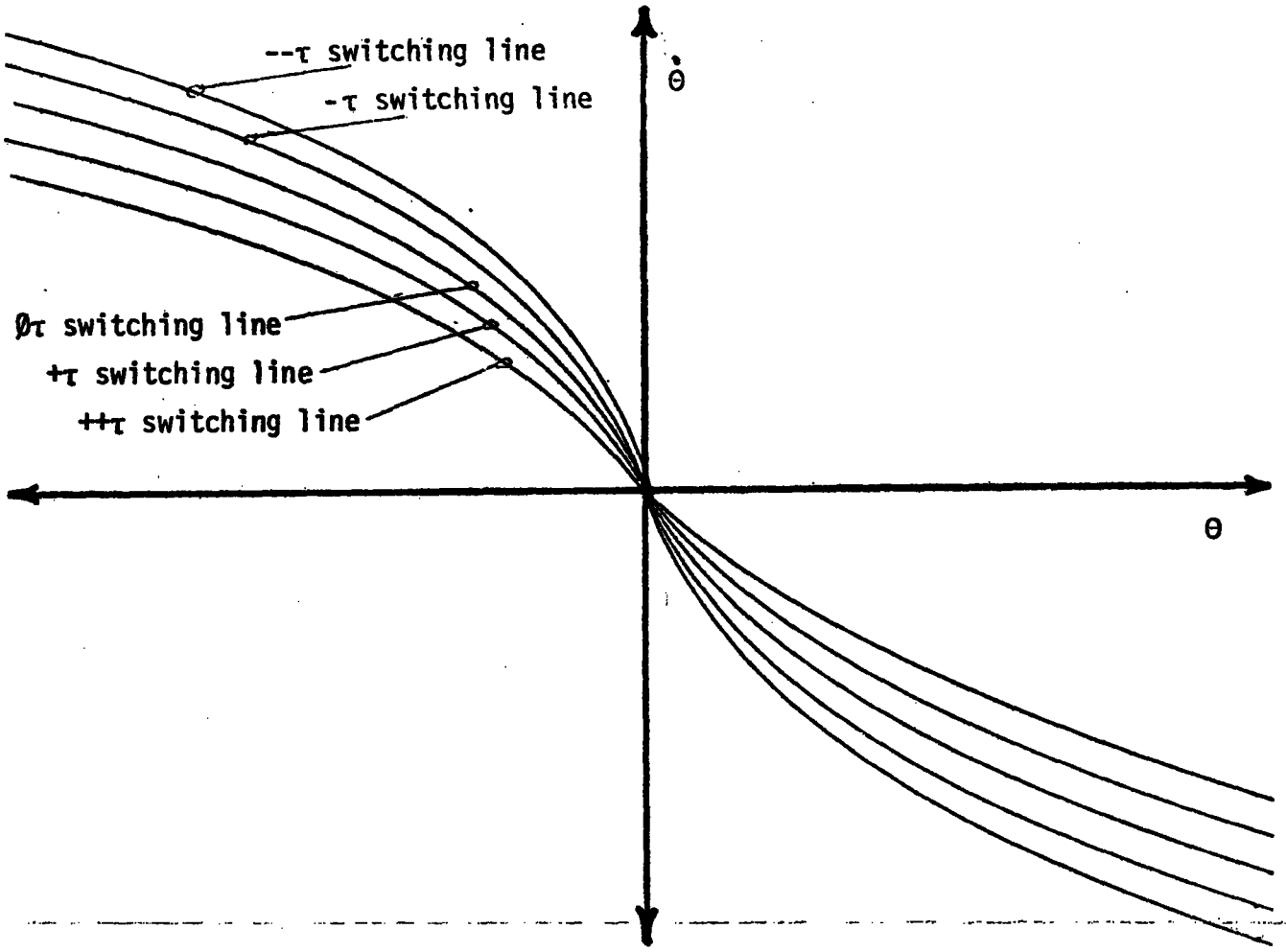


Fig. III.4 System state-space divided with five decision lines for five-torque logical controller.

III.C Gravity and Oscillations

The effects of gravity are handled as before. The state space for each joint is dissected into regions in accordance with the magnitude of the gravitational effect. A set of torques is then adapted for each region to compensate for gravity in that region. Sets of decision boundaries are also tailored for each region, in accordance with the trajectories derived from operating the system under that region's specific torques. Stability of the system around the origin is also handled as before: torque levels are reduced near the origin, which is enclosed in a small dead zone.

Finally, the finite state automaton that comprises the heart of the controller is modified to support five control states:

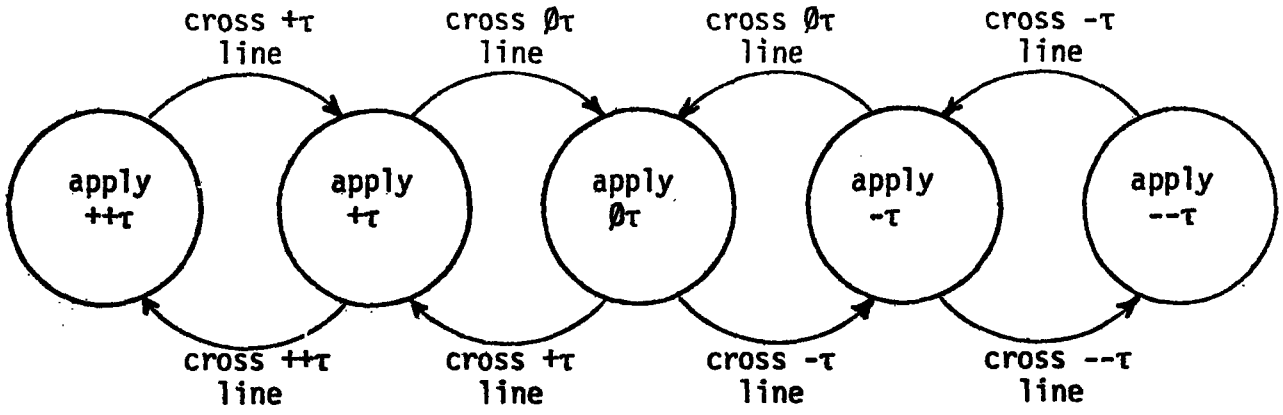


Fig. III.5 Finite-state machine for the five-torque logical controller.

With but a modest increase in complexity, the controller can now readily cope with disturbances arising from coupling torques, Coriolis forces and changes in configurations, and thus control the interacting links. As with the simpler version, this controller is straightforward to design and implement. And the decision lines can be extended in state space to regulate the intermediate velocity of each link.

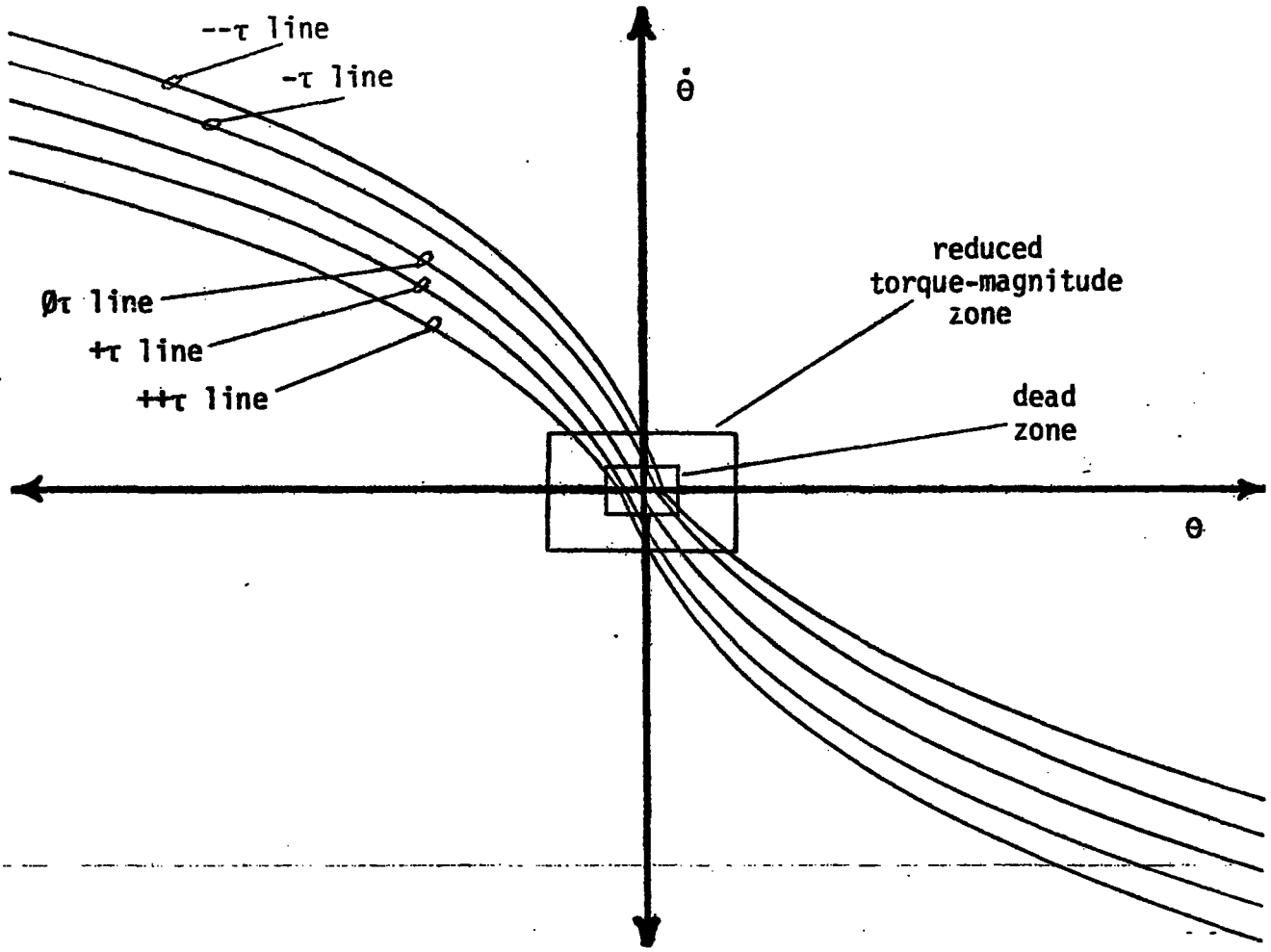


Fig. III.6 Complete set of switching lines for the five-torque logical controller.

III.D Application

This methodology can be applied to the task of controlling manipulators with two or more joints. Using the approach just discussed, we developed a logical controller for the Scheinman arm. For purposes of demonstration, our controller was designed to regulate the three major joints of the arm. (The three joints in the wrist can be regulated as a separate subsystem because they operate at much lower velocities and make minimal contributions to the system dynamics and to the gross motions achieved with the major joints of the shoulder and elbow.) A mathematical model for the three-link arm was used to determine torque magnitudes for each joint and then to develop and postion sets of switching lines. Our model appears in the appendix.

A set of actual state-space trajectories for the Vicarm executing a movement under our logical controller is presented on the next page. In this movement, all three joints were operated at arount 1.5 radians per second. The zero coordinate on the joint-one position axis is arbitrary since the joint rotates in a horizontal plane. The scale on the joint-two position axis is relative to straight up being the zero position. The scale on the joint-three position axis is relative to straight out from link two being the zero position. A picture of the arm appears in the appendix.

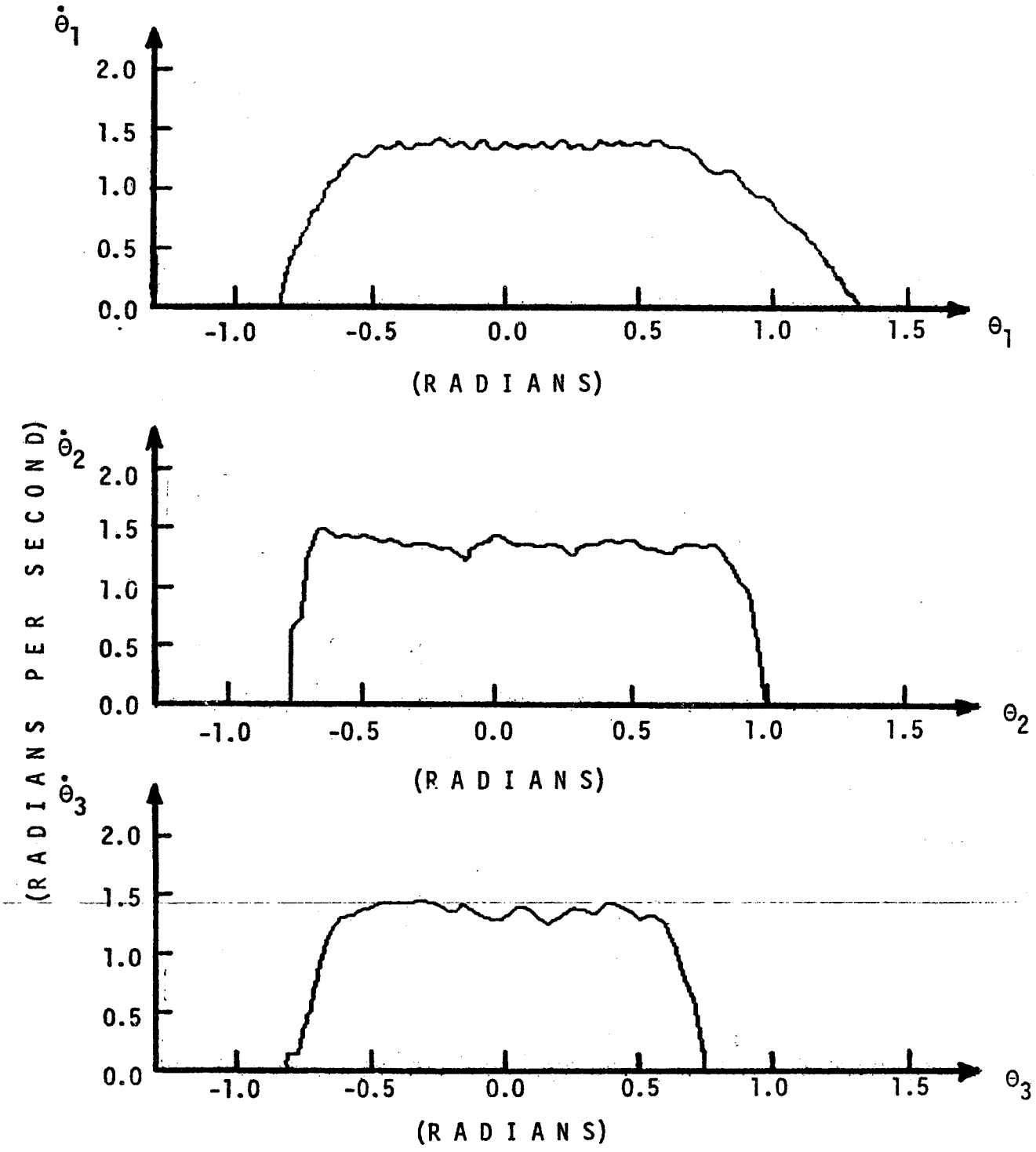


Fig. III.7 Actual system state-space trajectories for the Vicarm executing a movement.

III.E Summary

Here is a methodology for controlling a system with interacting degrees of freedom. This methodolgy was successfully applied to the task of controlling the three major joints of a mechanical arm having six degrees of freedom. With all three joints being driven at around 2.0 radians per second, we were able to constrain intermediate velocities by $\pm$0.10 radians per second in the typical case, and by $\pm$0.15 radians per second in the worst case; and achieve final positioning accuracy to within $\pm$0.015 radians. There was no detectable deterioration in performance in this system with interacting degrees of freedom until the sample/service period approached 3.5 ms. Yet the entire calculations for a control decision for each joint and D-to-A outputting of the three control torques takes about 1 ms on the PDP 11/45.

IV. What We've Done

IV.A Introduction

In this section we assess the development of our "logical" controller. We begin by reviewing the control strategy from a slightly different persepctive, and then compare it with other control strategies. This comparison allows us to highlight the logical controller's virtues but also expose its shortcomings.

IV.B Review of the Logical Controller

This work has examined and demonstrated the applicability of Logical Control Theory to manipulator control. We began with the general form of a logical controller, and described how and why that form was developed into a specific structure for controlling a system such as a mechanical arm. We then presented the important physical factors that must be examined in tailoring the controller to a particular manipulator.

The final product was a controller designed to regulate the arm using discrete knowledge about the dynamics of the arm. This knowledge consists of two components: (1) data on the responsiveness of each joint to particular fixed torques while the arm was in various configurations; (2) the range of potential near-future states of the arm for any current state (i.e. what the near-future state of a joint can be, given a current position, velocity, and input torque).

This combined knowledge is used to make the control decisions. The controller repeatedly selects a control torque that is based on the current state of a joint and the goal

state for that joint, such that the two states will continue to converge. The latest control torque will not necessarily drive the joint exactly to the goal. Rather, this torque is applied because it is best suited to regulating the joint at that moment. The controller reexamines the joint about once every millisecond to revise the control input if needed. The carefully tailored decision lines and frequent status checks insure that each joint is satisfactorily regulated.

The quality of the condition detectors (the position and velocity sensors) proved to be a major problem. The outputs from these devices, especially the tachometers, were discouragingly noisy. Noise in the velocity signals was roughly in proportion to joint velocity, but often ranged over ± 0.25 radians/second. This was compounded by the observable effects of backlash in the gear trains. Some of the noise was smoothed over with a triangle filter applied to the latest three sensor readings. Improved sensors would be necessary in any real application.

These problems notwithstanding, the controller achieved performance statistics that compare favorably with systems using competing control methodologies. An important point is that this performance was realized on original (non-rehearsed or preprogrammed) movements with a controller executing simple and rapid control calculations in real time. It is possible that the controller decision boundaries could be designed with more sophistication to improve the general performance with little or no increase in the complexity of the actual controller and it's calculations. This task will be discussed section V.

Our original logical controller utilized three different torques (positive, zero, and negative) to drive the arm. The revised version utilized five different torques (high-positive,

low-positive, zero, low-negative, and high-negative). We predicted and demonstrated improved performance by the controller as we went from two to three to five torque levels. We also encountered some increase in controller complexity, corresponding to the increase in the number of torques. It becomes tempting to propose that there exists a continuum of performance (and complexity) between the simple two torque controller, and a controller that utilizes an "infinite" number of different torques (i.e. the continuously variable torques applied by a PID or modern controller). To a limited extent, this is true, but there is not an infinite improvement in performance when the controller utilizes an infinite number of torques. Of course neither is there an infinite increase in controller complexity as one (changes control strategy and) utilizes an infinite number of torques.

Within reason, a controller can apply only a finite number of resolvably different torque levels --many fewer than is suggested by the range of a typical digital-to-analog interface. And, in fact, a controller need be able to apply only a finite number of torques. The difference in the effect which two different torques can have on a link is often undetectable, particularly when the chosen torque is applied for only a few milliseconds. This is particularly characteristic of a logical controller applied to an arm, since the manipulator acts as a low-pass filter, smoothing out much of the transient behavior caused by rapid switching of control torques.

IV.C The Classical Controller

Both a PID controller and our logical controller *operate* as though each joint were independent. It is important to note that we have not made any preliminary assumptions or assertions that the joints are truly independent. Rather, we have considered the manipulator as a system of interacting degrees of freedom and examined how affecting each degree of freedom can have an effect on the others. (Again, it is convenient, though not completely accurate, to call these interaction effects "disturbances".) In particular, we had to determine, for each link, the maximal interaction disturbance that could arise from all other links. The controller's decision criteria and torque levels were then based on the sum of all maximum disturbances that could act on each link. The controller was endowed with a set of control torques for each joint that was a function of the desired dynamics for that joint, plus the possible disturbances arising from other joints being driven by their sets of torques. Each joint could thus be operated independently since the controller was designed to overcome all interaction disturbances that might arise.

IV.D The Modern Controller

Both a modern-control-theory (MCT) controller and our logical controller treat the arm as though it were a discrete system. The controllers periodically examine the state of the system by means of position and velocity sensors and from these observations determine a new set of control torques. These control torques are then applied to the system for some predetermined interval of time where upon the system state is rechecked. This cycle is repeated until the goal state is achieved.

The MCT controller uses a mathematical model of the system to compute each set of control torques. This model is sufficiently general that the controller can calculate a new set of control torques for any sensed system state. Our logical controller uses a fixed set of control torques that is developed in advance. A mathematical model of the system is initially used to determine what torques must be applied to regulate the arm within some reasonable range of states. The torques are actually selected to insure that the expected extreme states of the system can be handled. These same torques are used, in judicious combinations, to regulate the system throughout the range of states. That is to say, since this repertoire of torques is designed to handle all extreme cases over some range of system states, we know that a solution for all control problems within that range can be approximated with a series of torques from that set. It is this piecewise combination of fixed torques that the logical controller dynamically determines.

In fact, both controllers approximate the solution to the control problem with a series of discrete torques. The difference is that our logical controller selects from a fixed set of torques rather than compute a unique torque for each successive control decision, and that it typically applies that torque for a much shorter period. Since our logical controller's approximate solution to arm-control problems appears to provide regulation that is equal to that provided by any controller using a more exact solution, it cannot be discounted for using a fixed set of predetermined torques. It is entirely possible that the logical controller will often be, averaged over time, closer to the exact solution to a control problem than will a MCT controller, since it updates control decisions approximately once every millisecond, which is generally five to ten times more frequent than is done by a MCT controller.

IV.E The State-Space Controller

Both Raiberts's state-space (SS) controller and our logical controller utilize a state-space that is parameterized around the arm's degrees of freedom. Raibert's state-space is one hyperspace whose axes correspond to the positions and velocities of every joint in the arm. Our state-space is actually treated as a collection of projections onto phase-planes. Each joint is operated independently and so the controller uses independent phase planes in regulating each joint.

With both methodologies, the control space is dissected into regions. Associated with each region is a predetermined torque level (or set of torque levels) that is a function of the dynamics of the arm when the state is in that region. The controllers select which of these predetermined torques to apply partly on the basis of which region the arm's state is in at that moment. (Raibert's SS controller also performs a simple calculation using Newton's law to select a control torque; our logical controller uses hysteresis in selecting a control torque.)

Raibert dissected his state-space along axes corresponding to both position and velocity. This was apparently done by arbitrarily dividing each axis into several equal segments. Our phase planes were divided only along the axes corresponding to position. We dissected each axis by mapping equal divisions in the range of the link's responsiveness onto that link's position axis (see section II.6).

One other important difference in our controller is that it has hysteresis: the controller output is a function of the current state as well as how we got to that state.

Consider the two histories of the state of some joint, diagrammed below.

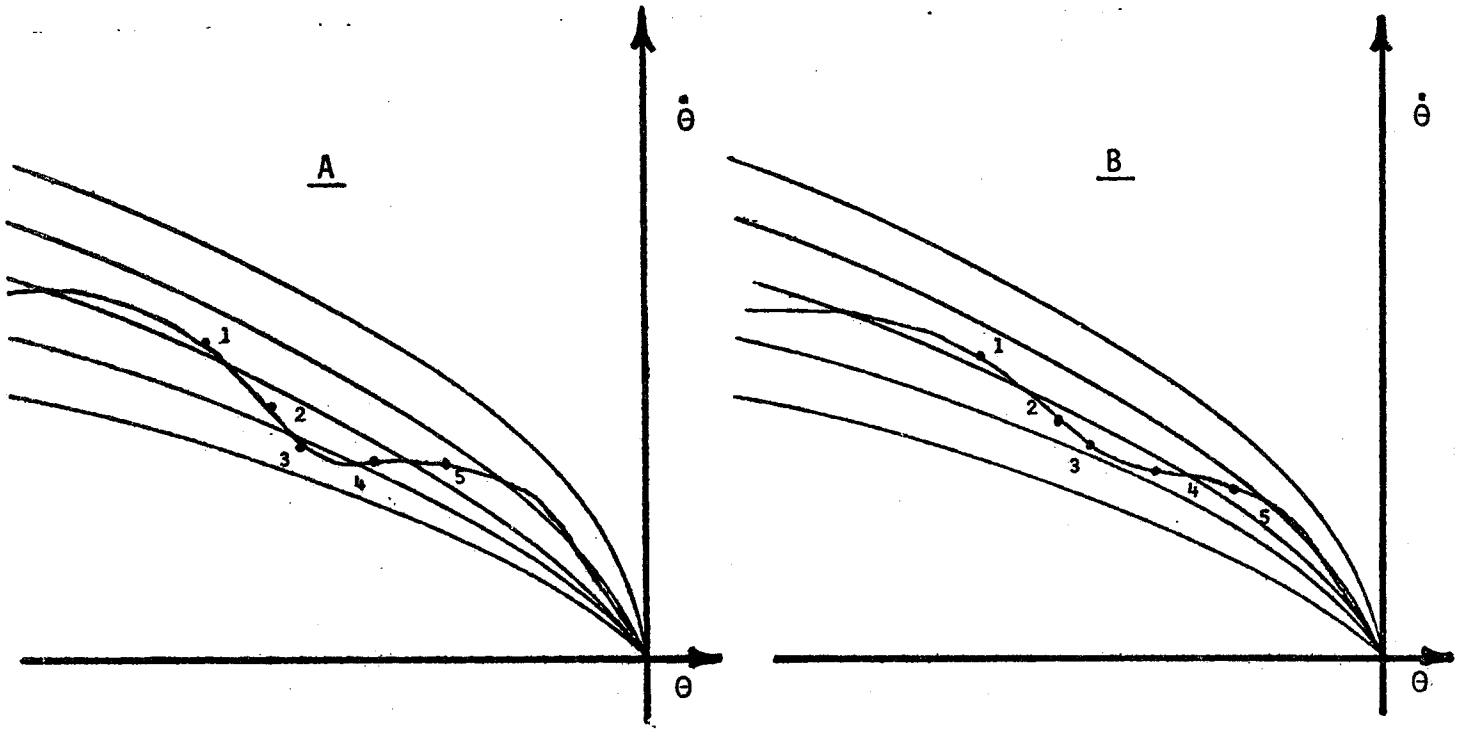


Fig. IV.1 The effects of hysteresis on the controller's behavior for two nearly identical system state-space trajectories.

In history A, the controller applies a low positive torque when the joint state is at points 3 and 4, whereas in history B, the controller applies a zero torque when the joint state is at points 3 and 4. (Control torques at points 1,2 & 5 are the same in both histories.)

Possibly, the performance of our controller could be improved by adopting one comprehensive hyperspace dissected along both position and velocity axes. In essence, this might equip the controller with more extensive knowledge about the arm's dynamics and thus might allow it to make more enlightened control decisions. However, this appears to be a near intractable design problem.

IV.F Variable-Structure-System Controller

A variation of the bang-bang control methodology has been proposed by K.D. Young in [Young, 78] for manipulator control. This approach uses the theory of variable structure systems to design the controller. The design strategy is to insure that when the system is in the so-called "sliding mode," its state-space trajectory lies in the controller's switching surface. When in sliding mode, the controller is said to be insensitive to parameter variations and disturbances in the system.

Both the variable-structure (VS) controller and our logical controller are designed by examining boundary conditions of system parameters. Both controllers are implemented to handle the worst-case disturbances arising from nonlinear interaction of the links, Coriolis forces, gravity, frictions and loading. As such, both yield a non-ideal solution to the control task.

The VS controller and our logical controller use a finite set of torques to control the arm. The VS controller uses two torques in accordance with the bang-bang structure, but the magnitude of the torques is computed as a linear function of the current distances of the state variables from the origin. The coefficients in the linear torque-equation fix the velocities at which the arm joints can be operated with the VS controller. They are selected

to optimize a tradeoff between the maximum speed of movements and the controller's final positioning accuracy. These coefficients must be set to insure that the system state will always converge on, and then stay in the controller's switching surface.

Our logical controller uses a set of five fixed-magnitude torques (with reduced-magnitude torques near the origin) that allow the arm to be operated over a wide range of joint velocities. Analogously, our torques are selected to insure that the sytem state converges on the switching envelope, and our decision lines are positioned to insure that the system state remains in the switching envelope.

The VS controller employs a bang-bang switching strategy with a single decision line in the system state space. This gives rise to the chattering phenomena that was discussed in section II.2. The introduction of a fixed switching delay was proposed to overcome this problem. Our controller uses a grid of five switching lines and an automatically-determined hysteresis to make control decisions. This eliminates chattering without introducing into the regulation any inaccuracy that is inherent in fixed-delay schemes.

Application of the VS control methodology has been examined in a hybrid simulation of a two-joint manipulator being regulated by a VS controller. Working with the assumptions that Coriolis and centrifugal effects, and interaction torques were negligable, the variable-structure approach was shown to be applicable to manipulator control design. This work has asserted that logical control theory is applicable to manipulator control, and the assertion has been successfully demonstrated in our real-world implementation of a logical controller.

IV.G Summary

The logical controller is designed to regulate a system using what amounts to discrete knowledge about that system. Applying this knowledge simplifies the nature of the control decisions that the controller must make without compromising the controller's ability to regulate the system. Comparing this strategy to other control strategies pinpoints its merits and its deficiencies, and thereby suggests possible improvements to our controller.

V. Where Do We Go From Here

V.A Introduction

In this section we discuss the issues that our work has suggested as meriting further investigation. The most pervasive issues concern the knowledge about the arm's dynamics that is used in making control decisions. If we assume that the general nature of our Logical Controller is to remain the same (and thus that the nature of the control decisions remains the same) then there are two major aspects of this knowledge to consider: (1) how comprehensive the knowledge is, and (2) how fine-grained the knowledge is. Once these issues are understood, the performance of a suitably refined logical controller should be rigorously compared with that of other control techniques applied to the same plant. Accessorial to these issues are other things the controller might be able to deal with, including collisions, dropped objects, and failing bearings and actuators.

V.B Dynamics Knowledge

By "comprehensiveness," we mean how many different factors that contribute to the arm's dynamics are accounted for in the controller's knowledge. These factors include simple dynamics, gravitational forces, frictional forces, interaction torques, Coriolis forces, and actuator and sensor dynamics. Our strategy for selecting the control-torque levels and positioning the switching lines incorporates complete information about simple dynamics into the knowledge used to make control decisions. To a limited extent the strategy has also incorporated information about gravitational, interaction, and Coriolis forces. We say

"limited" because of the fact that the compensation for the maximum effects of these forces was quantized along position parameters.

Likewise, knowledge about frictional forces has been included to a limited extent: stiction is taken into account by the dead zone around the origin, and in the torque levels; coulomb friction is approximately constant and so is also accounted for in the torque levels.

The maximum effects of viscous friction and Coriolis forces, for a given maximum velocity, are also taken into account in the torque levels. However, these forces can vary dramatically over a reasonable range of joint velocities. It is thus incorrect to parameterize the compensation for these forces strictly according to the position of a joint. One cannot freely place the horizontal segment of a set of decision lines arbitrarily far from the position axis (corresponding to selecting an arbitrary intermediate velocity) because the arm dynamics can vary due to the change in viscous friction and Coriolis forces. This complicates the regulation of intermediate velocities since the switching lines and torque levels can become inaccurate if we operate at some velocity radically different from the range for which we originally designed.

Comprehensiveness of the controller's knowledge about the arm's dynamics is thus an issue that merits further research. Further investigations should incorporate information about arm dynamics that is parameterized according to joint velocities. Though forces such as gravity and coulomb friction are more or less independent of these velocities, viscous friction and Coriolis forces are very dependent on joint velocities. These dependencies could be included in sets of decision lines and control torques assigned to regions in a joint state space that is partitioned along <u>both</u> position and velocity axes. This

more comprehensive knowledge might significantly improve the controller's performance.

Of course, the controller will incur an increase in control-decision complexity, and in the amount of memory space required to store the greater number of region-specific sets of torques and detison lines. The increase in decision complexity would be minimal because of the elementary nature of the calculations necessary to locate the region containing each joint's current state. Likewise, the increase in storage space should be a minor problem since each set of torques and decision lines requires approximately 200 bytes of memory.

The logical control of the arm as a system of coupled degrees of freedom might also be investigated. Thus far the controller has regulated each joint as an independent degree of freedom. The dynamics knowledge for each of the joints in the arm has been disjoint. Our justification for this approach was presented in section III.B. However, we know from both intuition and the principals of mechanics that the joint dynamics interact. A casual glance at the system equations (presented in the appendix) shows dozens of terms that are sums or products of state variables. It is thus appropriate to consider structuring the knowledge about the arm's dynamics accordingly. In this approach, the controller would use one all-inclusive hyperspace whose axes correspond to the positions and velocities of every joint in the arm. The hyperspace would be partitioned into hyperregions. Associated with each hyperregion would be a set of control torques and decision lines, with one set of control torques and decision lines per arm joint. Similar to the previous strategy, the controller selects sets of torques and decision lines to apply to each joint on the basis of which hyperregion currently contains the system state.

Problems arise in marking out the hyperregions and determining their

associated sets of torques and lines. As discussed earlier, to mark out regions one should first determine the range of effect the variation of a state variable can have on the dynamics of its joint. The range of that effect should then be evenly divided into several parts, and those divisions mapped onto the axis for that state variable. This is in contrast to blindly dissecting the axis into several equal segments.

The sets of control torques and switching lines applicable in a hyperregion would also be determined as before. The system model is used to assess the dynamics at each joint in the arm while the system is in that hyperregion. Control torques are selected to compensate for the disturbing forces (interaction torques, Coriolis and friction forces, etc.), and maintain a desired velocity or acceleration at the joint. Finally, decision lines are positioned around the system state-space trajectories through the hyperregion. It is important that the set of torques and switching lines for any hyperregion be determined for all joints simultaneously to account for the degrees of freedom interacting. This is necessary to insure that the magnitude of interaction torques will be correctly assessed.

This exhaustive dynamics knowledge would result in still more control-decision complexity, and would use more memory space. How much more depends upon the number of hyperregions. This brings us to the other major aspect of the controller's dynamics knowledge.

By "granularity", we refer to the fineness of the partitioning of the system state space. The assumption behind partitioning the state space into regions is that the dynamics of a joint are the same throughout the region (or hyperregion, if working with one composite hyperspace). This assumption is completely valid only in the extreme case of each

region being an individual point in the state space (which is essentially the division of state-space embodied in traditional control methodologies).

The extent of partitioning of the state space becomes an important issue to investigate. A simple trade-off is at the heart of this issue. The amount of memory space required to store dynamics knowledge for each region and the amount of computation required to determine exactly which region the state is currently in reflect the accuracy of the quantization of the joint's dynamics throughout a region.

The quantizing of the system dynamics is an application-dependent problem. In situations where precise regulation is vital, the implementation should favor a finely partitioned state space. This would provide relatively accurate representation of joint dynamics in each region. In situations where computation-time or memory is at a premium, the implementation should favor a coarsely partitioned state space. This would reduce the amount of data that must be stored to represent joint dynamics throughout the state space, and the amount of time required to locate some segment of that data.

Another problem is the number of torque levels used by the controller. As a finite state automaton, the controller can use only a finite set of discrete torques to regulate the arm. And this means there is another trade-off to be analyzed. A small number of torque levels minimizes the amount of data that must be stored on the dynamics at each joint in each hyperregion, and also the complexity of the controller's task of selecting what torque to apply to each joint at any moment. (A bang-bang controller is an extreme example of the use of few torque levels.) Conversely, a large number of torque levels facilitates smoother and more precise operation of the arm. (A controller that could apply an infinite number of

different torque levels would be an example of this other extreme.) Further investigations should examine the size of the controller's repretoire of discrete torques.

Since the number of torque levels affects the amount of data that must be stored as well as the agility of the controller in regulating the arm, this issue is closely related to the preceding two: comprehensiveness and granularity. In fact, these issues must be examined collectively in any general research or implementation effort. Our work concluded with the controller using a complete set of five torque levels in each region for each joint, which yielded decent performance.

As an aside, we note that when the system is in particular states, the controller will never apply certain control torques. For example, the controller would not require very high torques in a region of low velocity near the origin. Similarly very small torques are useless in a region of high velocity far from the origin and working against gravity. Thus, it may not be necessary that a complete set of control torques be specified for every joint in every region (or hyperregion). In any event, based on our experience, we expect that a set of less than 10 torques would be adequate for most applications.

V.C Comparison Testing

Relative to other control methodologies, logical control theory is still in its infancy. This is particularly true with regards to its application to complex mechanical systems. Yet the work reported herein has demonstrated that locigal control theory has real potential in such applications.

Before extensive real-world application of logical control theory can be

advocated, its qualities must be rigorously compared with those of other control methodologies. Controllers for a variety of benchmark systems must be designed using the competing theories. The benchmark systems should cover a range of test criteria; certain certeria will be important in the regulation of individual systems. For a mechanical arm, we expect the following will be important (this is offered as a partial list, in no particular order): accuracy of the regulation, range and limits of velocities at which the joints can be operated, ease and rapidity with which new movements can be specified, amount of resources required for the implementation of the controller, amount of effort required to design the controller and tolerance to system degradation.

Every controller must be carefully designed to fairly assess its capabilities for regulation the benchmark systems. The design procedure for some of the more popular control strategies (e.g. PID control, modern control) has been fully systematized. Currently this is not the case for designing and implementing a logical controller. While we have presented various factors that must be considered in the design process, these factors have not been organized into a comprehensive framework. This organization will depend largely upon the results of an investigation into the issues cited in the previous section. Therefore, performance testing should be undertaken only after the questions that were discussed earlier are formally resolved.

V.D Other Things in State-Space

The normal state-space trajectory for any joint being to the origin is approximately parabolic. Any radical deviation from a parabolic trajectory could indicate to the controller that some abnormal situation has arisen. A few of these are discussed below.

V.D.1 Collisions

If some joint state should suddenly diverge from the expected parabolic state-space trajectory and jump to the position axis, or simply move unrelentingly towards the position axis (i.e. the joint velocity suddenly drops drastically, or drops and cannot be raised again) then some link supported by this joint has possibly collided with an obstruction. When the controller observes such an occurrence, it might back off and shut down the arm, and then signal for help. It might also determine the approximate location of the obstruction from perturbations in all joints' state-space trajectories and the current position of all joints, and then make note of that object.

V.D.2 Dropped Objects

If some joint state should suddenly diverge from the expected parabolic state-space trajectory and move rapidly away from the position axis (i.e. the joint velocity suddenly increases drastically) then the arm has possibly dropped its payload or is falling apart. When the controller observes such an occurrence, it might shut down the arm and then signal for help. It might also estimate the flight path of the dropped object from the current position and velocity of each joint (i.e. assume it had just thrown the object) and offer this information to efforts to relocate the dropped object.

V.D.3 Failing Bearings and Actuators

If some movements become frequently characterized by erratic jumping of the system state, or undershooting of the origin, then the bearings or actuators in the arm may be failing. When the controller observes such occurrences, it might shut down the arm and signal for repairs. Alternatively, the supervisory controller or planner might experiment with adjusting the magnitudes of the set of fixed torques in order to compensate for the deteriorating mechanisms.

V.E Summary

This work has developed a basic framework for what appears to be a promising control methodolgy. Our development and implementation efforts point out a number of issues that require further investigation. Among these are: the nature of the dynamics knowledge that the controller uses in making control decisions, especially with respect to its comprehensiveness and granularity; the regulation capabilities, relative to controllers designed with other methodologies, of a logical controller applied to complex mechanical systems; the handling of anomalies in the system's behavior. Resolution of these issues is vital to realizing the potential of logical control theory.

Bibliography

[Bejczy, 76]
Antal K. Bejczy, "Issues in Advanced Automation for Manipulation Control," Proceedings of the 1976 Joint Automatic Control Conference, West Lafayette, Indiana, July 1976, pp. 700-711.

[Blanchard, 76]
David C. Blanchard, Digital Control of a Six-Axis Manipulator, M.I.T. AI Lab Memo no. 129, B.S. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, August 1976.

[Corwin, 75]
Merton Corwin, "The Benefits of a Computer Controlled Robot," Proceedings of the Fifth International Symposium on Industrial Robots, Chicago, Illinois, September 1975.

[Cosgriff, 58]
Robert L. Cosgriff, Nonlinear Control Systems, McGraw-Hill Book Company, New York, 1958.

[Deutsch, 69]
Ralph Deutsch, System Analysis Techniques, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.

[Dertouzos, 73]
Michael L. Dertouzos, Control Robotics; The Procedural Control of Physical Processes, Technical Memorandum 11, Engineering Robotics Group, Project MAC, M.I.T., November 1973.

[Ernst, 61]
Heinrich A. Ernst, A Computer-Operated Mechanical Hand, ScD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, December 1961.

[Finkel, 76]
Raphael A. Finkel, Constructing and Debugging Manipulator Programs, Stanford AI Project Memo no. 284, Stanford Computer Science Report STAN-CS-76-567, PhD Thesis, Computer Science Department, Stanford University, August 1976.

[Goertz, 63]
R. C. Goertz, "Manipulators Used for Handling Radioactive Materials," chap. 27 of Human Factors in Technology, McGraw-Hill Book Company, New York, 1963.

[Greenwood, 65]
Donald T. Greenwood, Principles of Dynamics, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1965.

[Hill & Peterson, 74]
Frederick J. Hill and Gerald R. Peterson, Introduction to Switching Theory and Logical Design, 2nd ed., John Wiley and Sons, Inc., New York, 1974.

[Hopkins & Quagliata]
Albert L. Hopkins, Jr. and Louis J. Quagliata, State Transition Synthesis for Sequential Control of Discrete Machines, (paper received from author, undated).

[Horn, 75]
Berthold K. P. Horn, Kinematics, Statics, and Dynamics of Two-D mainpulators, M.I.T. AI Lab Working Paper no. 99, June 1975.

[Horn, 77]
Berthold K. P. Horn, Ken-ichi Hirokawa and Vijay V. Vizirani, Dynamics of a Three Degree of Freedom Kinematic Chain, M.I.T. AI Lab Memo no. 478, October 1977.

[IITRI, 75]
IIT Research Institute, Proceedings of the Fifth International Symposium on Industrial Robots, Chicago, Illinois, September 1975.

[Kahn & Roth, 71]
M. E. Kahn and B. Roth, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," ASME Journal of Dynamic Systems, Measurement, and Control, September 1971, pp. 164-172; see also M. E. Kahn, The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains, PhD Thesis, Stanford University, December 1969.

[Kalman, Falb & Arbib, 69]
R. E. Kalman, P. L. Falb and M. A. Arbib, Topics in Mathematical System Theory, McGraw-Hill Book Company, New York, 1969.

[McGhee, 67]
Robert B. McGhee, "Finite State Control of Quadruped Locomotion", Simulation, September 1967, pp. 135-140; see also Rajko Tomovic and Robert B. McGhee, "A Finite State Approach to the Synthesis of Biological Control Systems", IEEE Transactions on Human Factors in Electronics, June 1966, pp. 65-69.

[Ogata, 70]
Katsuhiko Ogata, Modern Control Engineering, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1970.

[Paul, 72]
Richard Paul, Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm, Stanford AI Project Memo no. 177, Stanford Computer Science Report STAN-CS-72-311, PhD Thesis, Computer Science Department, Stanford University, November 1972.

[Pieper, 68]
D. L. Pieper, The Kinematics of Manipulators Under Computer Control, Stanford AI Project Memo no. 72, Stanford Computer Science Report STAN-CS-68-116, October 1968.

[Raibert, 77]
Marc H. Raibert, Control and Learning by the State Space Model: Experimental Findings, M.I.T. AI Lab Memo no. 412, March 1977; see also Marc H. Raibert, A State-Space Model for Sensorimotor Control and Learning, M.I.T. AI Lab Memo no. 351, January 1976.

[Roderick, 76]
Michael D. Roderick, Discrete Control of a Robot Arm, Engineers Thesis, Stanford University, August 1976.

[Thaler & Brown, 60]
George J. Thaler and Robert G. Brown, Analysis and Design of Feedback Control Systems, 2nd ed., McGraw-Hill Book Company, New York, 1960.

[Tomovic, 66]
Rajko Tomovic, Introduction to Nonlinear Automatic Control Systems, trans. Paul Pignon, John-Wiley Sons, Ltd., London, 1966 (copyrighted by NOLIT Publishing House, Belgrade, Yugoslavia).

[Tou, 59]
Julius T. Tou, Digital and Sampled-Data Control Systems, McGraw-Hill Book Company, New York, 1959.

[Waters, 73]
Richard C. Waters, Mechanical Arm Control, M.I.T. AI Laboratory Vision Flash no. 42, March 1973.

[Whitney, 72]
D. E. Whitney, "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", ASME Journal of Dynamic Systems, Measurement, and Control, December 1972, pp. 303-309.

[Young, 78]
Kar-Keung D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure Systems", IEEE Transactions on Systems, Man and Cybernetics, February 1978, pp. 101-109.

APPENDIX

Our logical controller was implemented to regulate the MIT Scheinman Vicarm. This device, pictured on the next page, is scaled to two thirds human proportions. Measurements for important parameters are listed below. (Our work considered only the three major joints of the arm.)

	Length	Mass
$link_1$	l_1 = 3.75 cm	m_1 = .45 kg
$link_2$	l_2 = 25.5 cm	m_2 = 1.36 kg
$link_3$	l_3 = 21.0 cm	m_3 = .90 kg

To simplify the derivation of dynamics equations for the arm, we use a generalized form of Langrange's equation [Greenwood, 65]:

$$Q_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}$$

Q_i is a generalized force, q_i is a generalized coordinate, and L is the Langrangian Function, defined as $L = K - P$ where K is the kinetic energy of the system, and P is the potential energy of the system. There must be exactly one such equation for each degree of freedom in the system.

Our approach is similar to that taken in [Horn, 75] and, to a lesser extent, in [Horn, 77] and [Paul, 72]. The major differences are in how the link offsets are handled and how the distributions of

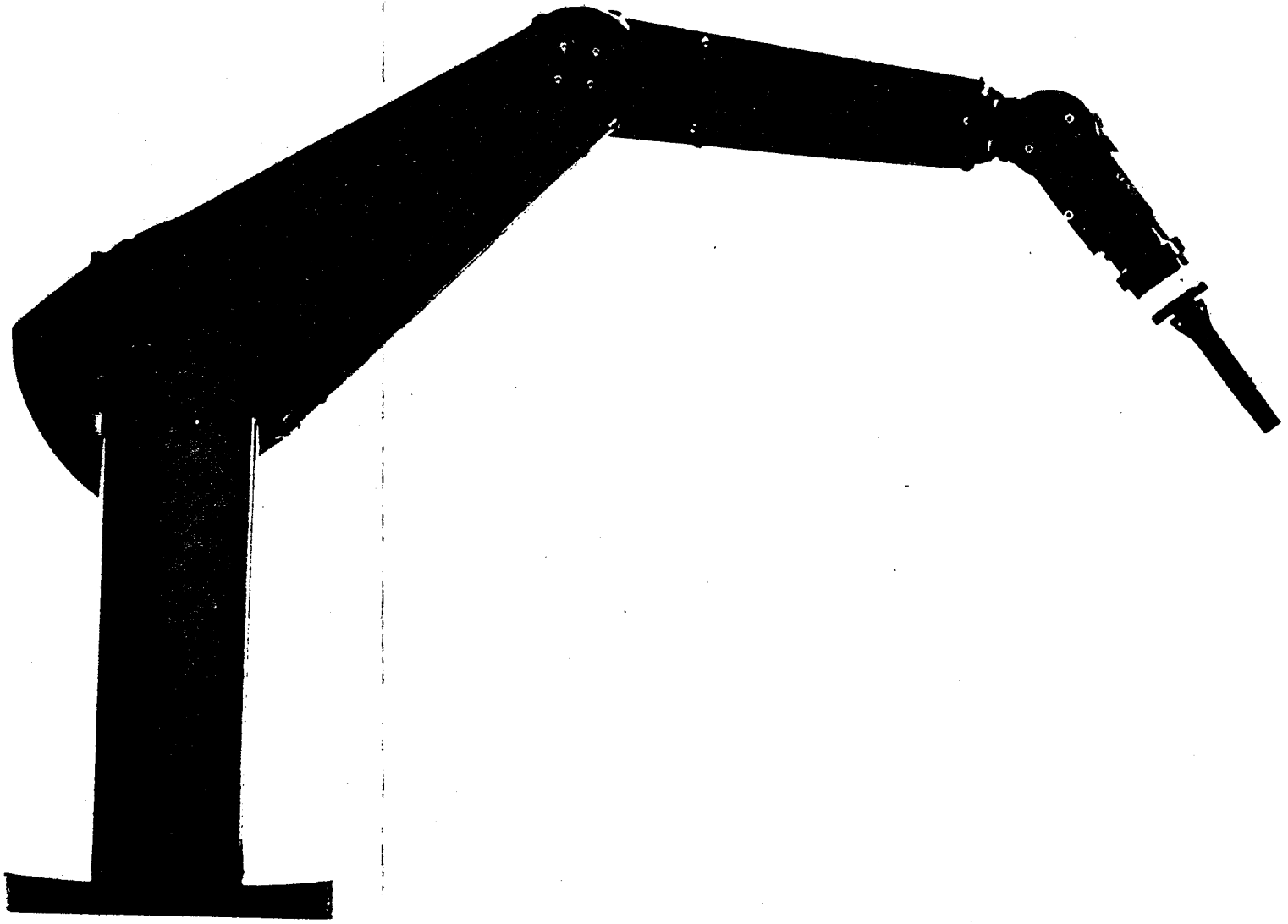


FIG. A.1 PHOTOGRAPH OF THE M.I.T. VICARM

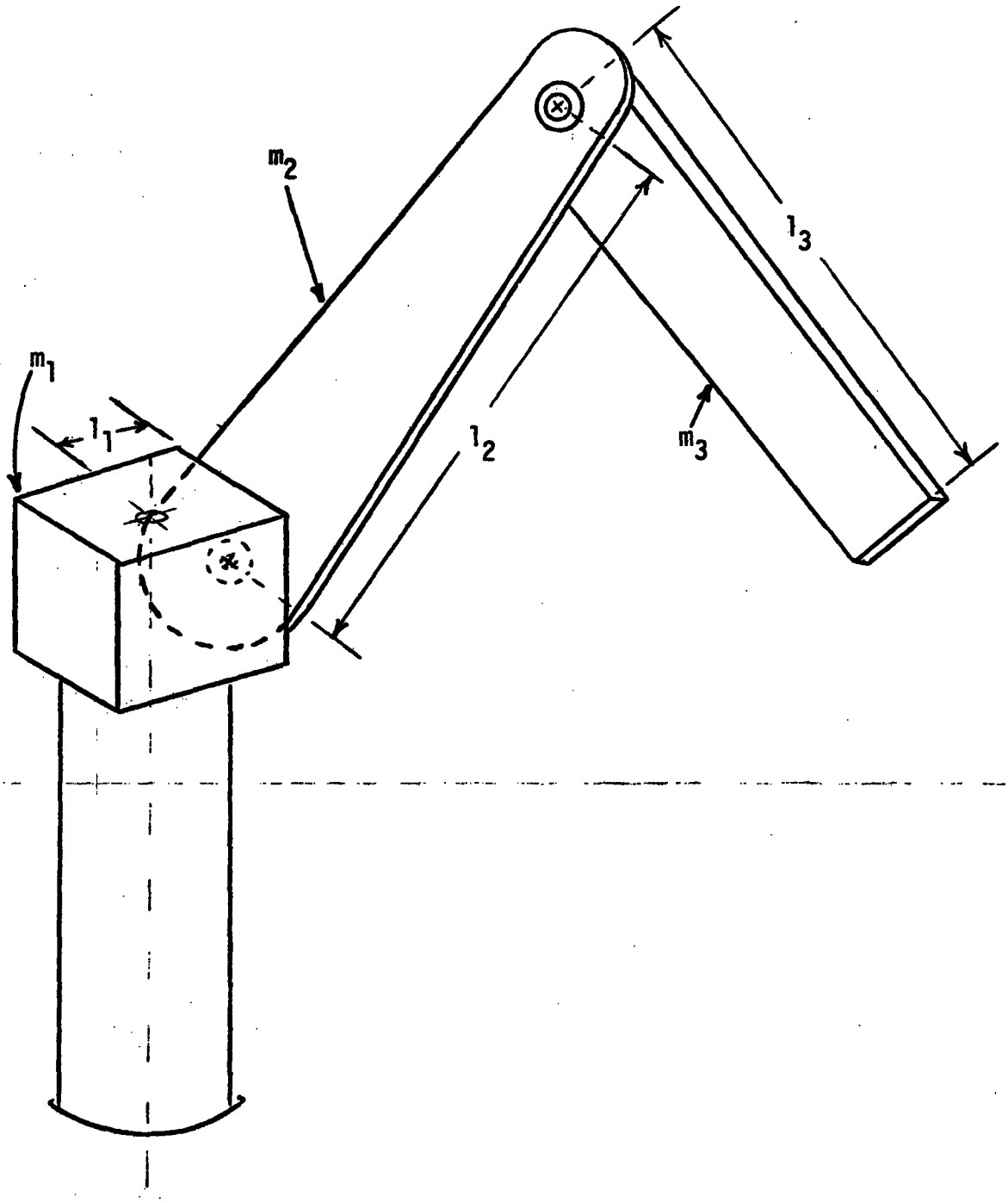
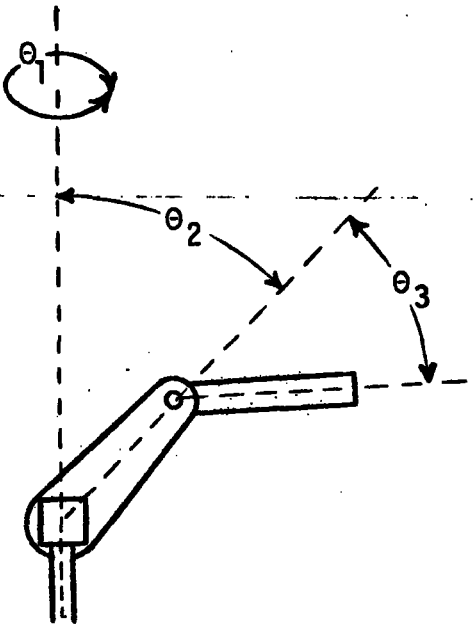


FIG. A.2 DRAWING OF THE M.I.T. VICARM

mass are approximated. The net result of these differences is that the kinetic energy of the system is represented in different expressions with slightly different coefficients.

For the Vicarm, there are three equations, corresponding to the three major joints we controlled. Each q_i corresponds to the i^{th} joint angle. The Q_i's correspond to actuator torques applied at the joints. The kinetic energy of our system occurs as links' masses with rotational and translational velocity, and the potential energy occurs as mass elevated above some plane. In the equations that follow, the angles of the arm joints are measured as illustrated below. (The angle of joint 2 is measured with respect to straight up being the zero position; the angle of joint 3 is measured with respect to straight out from link 3 being the zero position.)



Kinetic Energy

Rotational kinetic energy of an object is defined as $KE = \frac{1}{2}I\omega^2$ where I is the moment of inertia of the object, and ω is its angular velocity. For the geometry of the Vicarm, we can compute the kinetic energy in two components:

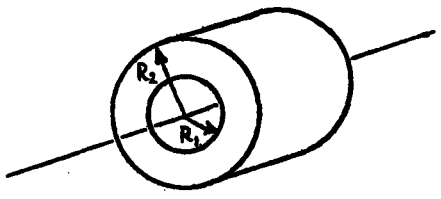
The first component comes from rotation about the axis of joint 1 (the arm can be viewed as carving out a conic surface centered at the joint-one axis). Let K_{1i} be this first component of the kinetic energy of $link_i$.

For $link_1$ $K_{11} = \frac{1}{2}\frac{2m_1l_1^2}{5}\dot{\theta}_1^2$ where $\frac{2m_1l_1^2}{5}$ gives the rotational inertia of $link_1$ (with mass m_1), approximated as a sphere: Link 1 is essentially a cubic solid that rotates atop a stationary pedestal; its distribution of mass can be adequately approximated as a sphere of uniform density.

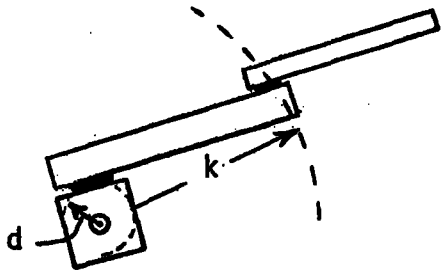
To compute K_{12} and K_{13}, we approximate links 2 and 3 as annular cylinders, with uniform distributions of mass (m_2 and m_3, respectively). The formula for the rotational inertia of an annular cylinder is

$$I = \frac{m}{2}(R_1^2 + R_2^2)$$

where R_1 and R_2 are as illustrated below.



For link 2, R_1 and R_2 are approximated as illustrated below.

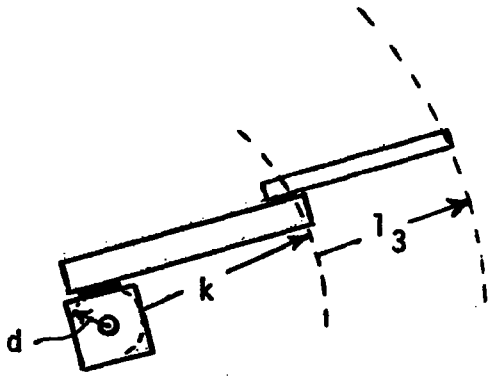


(overhead view of arm)

Note that d is the offset of link 2 from the joint-one axis and is, in fact, equal to l_1. This offset is significant in the arm's dynamics. We use "d" simply to denote this displacement of link 2; it remains constant, independent of all joint positions. And k marks off the rest of link 2 —roughly l_2 minus d. The term $(d + k\sin\theta_2)$ is the external radius for the cylinder approximating link 2, and will also be the internal radius for the link 3 approximation.

Now
$$K_{12} = \frac{1}{2}[\frac{m_2}{2}(d^2 + (d + k\sin\theta_2)^2)]\dot{\theta}_1^2$$

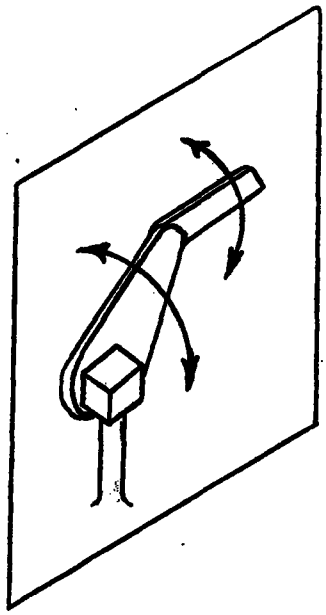
For link 3, R_1 and R_2 are approximated as illustrated below.



(overhead view of arm)

Thus $K_{13} = \frac{1}{2}[\frac{m_3}{2}((d + k\sin\Theta_2)^2 + (d + k\sin\Theta_2 + l_3\sin(\Theta_2+\Theta_3))^2)]\dot{\Theta}_1^2$

The second component of the arm's kinetic energy comes from rotation of the links through an arbitrary vertical plane, as illustrated below.



Let K_{2i} be this second component of the kinetic energy of $link_i$. Since link 1 does not rotate in a vertical plane, $K_{21} = \emptyset$.

To compute K_{22} and K_{33}, we approximate the masses of links 2 and 3 as rods with uniform distributions of mass m_2 and m_3, respectively. (Thus the center of mass of each rod is assumed to be at its physical center.) For link 2 we simply use the formula for the rotational inertia of a rod rotating about an end $I = \frac{ml^2}{3}$ where l is the length of the rod. Since the angular velocity of link 2 is $\dot{\Theta}_2$, we have

$$K_{22} = \frac{1}{2}(\frac{m_2 l_2}{3})\dot{\Theta}_2^2$$

Since link 3 experiences both rotation and translation in our vertical plane, the expression for its kinetic energy is more complex. We consider rotation of a rod about its center (axis perpendicular to its length), and linear translation of the rod's center.

The rotational inertia of a rod about its center is given by $I = \frac{1}{12}ml^2$. The angular verocity of link$_3$'s center is $\dot{\theta}_2+\dot{\theta}_3$. Thus the rotational component of K_{23} is

$$\frac{1}{2}(\frac{1}{12}m_3l_3^2)(\dot{\theta}_2+\dot{\theta}_3)^2$$

$$= \frac{1}{2}m_3\frac{1}{12}l_3^2(\dot{\theta}_2^2 + 2\dot{\theta}_2\dot{\theta}_3 + \dot{\theta}_3^2)$$

Translational kinetic energy is given by $KE = \frac{1}{2}mv^2$. The linear velocity of the center of link 3 is given by

$$\vec{r}_2\dot{\theta}_2 + \frac{1}{2}\vec{r}_3(\dot{\theta}_2+\dot{\theta}_3)$$

where $\underline{r}_2 = l_2(\sin\theta_2,\cos\theta_2)$

and $\vec{r}_3 = l_3(\sin(\theta_2+\theta_3),\cos(\theta_2+\theta_3))$.

(Vectors $\vec{r}_2$ and $\vec{r}_3$ must be used to correctly describe the linear velocity of link 3.) Squaring the linear velocity expression gives

$$l_2^2\dot{\theta}_2^2 + l_2l_3\cos(\theta_3)\dot{\theta}_2(\dot{\theta}_2+\dot{\theta}_3) + \frac{1}{4}l_3^2(\dot{\theta}_2+\dot{\theta}_3)^2$$

$$= l_2^2\dot{\theta}_2^2 + l_2l_3\cos(\theta_3)\dot{\theta}_2^2 + \frac{1}{4}l_3^2\dot{\theta}_2^2 + l_2l_3\cos(\theta_3)\dot{\theta}_2\dot{\theta}_3 + \frac{1}{2}l_3^2\dot{\theta}_2\dot{\theta}_3 + \frac{1}{4}l_3^2\dot{\theta}_3^2$$

Thus the translational component of K_{23} is

$$\frac{1}{2}m_3[l_2^2\dot{\theta}_2^2 + l_2l_3\cos(\theta_3)\dot{\theta}_2^2 + \frac{1}{4}l_3^2\dot{\theta}_2^2 + l_2l_3\cos(\theta_3)\dot{\theta}_2\dot{\theta}_3 + \frac{1}{2}l_3^2\dot{\theta}_2\dot{\theta}_3 + \frac{1}{4}l_3^2\dot{\theta}_3^2]$$

Combining the expressions for the rotational and translational components of K_{23} gives

$$K_{23} = \frac{1}{2}m_3[(l_2^2 + \frac{1}{3}l_3^2 + l_2l_3\cos(\theta_3))\dot{\theta}_2^2 + (\frac{2}{3}l_3^2 + l_2l_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3 + \frac{1}{3}l_3^2\dot{\theta}_3^2]$$

The total kinetic energy of the system is now

$$KE = K_{11} + K_{12} + K_{13} + K_{21} + K_{22} + K_{23}$$

$$= \frac{1}{2}(\frac{2m_1l_1^2}{5})\dot{\theta}_1^2 + \frac{1}{2}[\frac{m_2}{2}(d^2 + (d + k\sin\theta_2)^2)]\dot{\theta}_1^2$$

$$+\frac{1}{2}\frac{m_3}{2}[(d + k\sin\theta_2)^2 + (d + k\sin\theta_2 + l_3\sin(\theta_2 + \theta_3))^2]\dot{\theta}_1^2$$

$$+ \quad \emptyset \quad + \quad \frac{1}{2}(\frac{m_2l_2}{3})\dot{\theta}_2^2$$

$$+ \frac{1}{2}m_3[(l_2^2 + \frac{1}{3}l_3^2 + l_2l_3\cos(\theta_3))\dot{\theta}_2^2 + (\frac{2}{3}l_3^2 + l_2l_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3 + \frac{1}{3}l_3^2\dot{\theta}_3^2]$$

Potential Energy

The potential energy of the arm exists in the link masses elevated above some plane. The elevation of link 1 is constant and so makes no contribution to the Lagrangian for the system. The elevation of the center of mass of link 2 is $\frac{1}{2}l_2(1 + \cos\theta_2)$. And the elevation of the center of mass of link 3 is $l_2(1 + \cos\theta_2) + \frac{1}{2}l_3(1 + \cos(\theta_2+\theta_3))$. (These expressions provide for the potential energy of the system to be zero when the arm hangs straight down.) Thus the potential energy of the system is

$$PE = g[\tfrac{1}{2}m_2 l_2(1 + \cos\theta_2) + m_3(l_2(1 + \cos\theta_2) + \tfrac{1}{2}l_3(1 + \cos(\theta_2+\theta_3)))]$$

Combining the expressions for the kinetic energy and potential energy gives the Langrangian for the system: $L = KE - PE$

$$L = \tfrac{1}{2}\left(\frac{2m_1 l_1^2}{5}\right)\dot{\theta}_1^2 + \tfrac{1}{2}\left[\frac{m_2}{2}(d^2 + (d + k\sin\theta_2)^2)\right]\dot{\theta}_1^2$$

$$+ \tfrac{1}{2}\tfrac{m_3}{2}[(d + k\sin\theta_2)^2 + (d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))^2]\dot{\theta}_1^2$$

$$+ \emptyset + \tfrac{1}{2}\left(\frac{m_2 l_2}{3}\right)\dot{\theta}_2^2$$

$$+ \tfrac{1}{2}m_3[(l_2^2 + \tfrac{1}{3}l_3^2 + l_2 l_3\cos(\theta_3))\dot{\theta}_2^2 + (\tfrac{2}{3}l_3^2 + l_2 l_3\cos(\theta_3))\dot{\theta}_2\dot{\theta}_3 + \tfrac{1}{3}l_3^2\dot{\theta}_3^2]$$

$$- g[\tfrac{1}{2}m_2 l_2(1 + \cos\theta_2) + m_3(l_2(1 + \cos\theta_2) + \tfrac{1}{2}l_3(1 + \cos(\theta_2+\theta_3)))]$$

Now we derive the partial derivatives of L with respect to θ_1, θ_2, θ_3, $\dot{\theta}_1$, $\dot{\theta}_2$, and $\dot{\theta}_3$.

$$\frac{\partial L}{\partial \theta_1} = \emptyset$$

$$\frac{\partial L}{\partial \theta_2} = \frac{m_2}{2}(k\cos\theta_2(d + k\sin\theta_2))\dot{\theta}_1^2$$

$$+ \frac{m_3}{2}[k\cos\theta_2(d + k\sin\theta_2) + (d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))(k\cos\theta_2 + l_3\cos(\theta_2+\theta_3))]\dot{\theta}_1^2$$

$$- g[m_2\frac{l_2}{2}\sin\theta_2 + m_3 l_2\sin\theta_2 + m_3\frac{l_3}{2}\sin(\theta_2+\theta_3)]$$

$$\frac{\partial L}{\partial \theta_3} = \frac{1}{2}m_3[(-l_2 l_3\sin\theta_3)(\dot{\theta}_2^2 + \dot{\theta}_2\dot{\theta}_3)]$$

$$+ \frac{m_3}{2}[(d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))(l_3\cos(\theta_2+\theta_3))]\dot{\theta}_1^2$$

$$- gm_3\frac{l_3}{2}\sin(\theta_2+\theta_3)$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (\frac{2m_1 l_1^2}{5})\dot{\theta}_1 + \frac{m_2}{2}(d^2 + (d + k\sin\theta_2)^2)\dot{\theta}_1$$

$$+ \frac{m_3}{2}[(d + k\sin\theta_2)^2 + (d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))^2]\dot{\theta}_1$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = (\tfrac{1}{3}m_2 l_2^2)\dot{\theta}_2 + m_3[(l_2^2 + \tfrac{1}{3}l_3^2 + l_2 l_3 \cos\theta_3)\dot{\theta}_2 + (\tfrac{1}{3}l_3^2 + l_2 l_3 \cos\theta_3)\dot{\theta}_3]$$

$$\frac{\partial L}{\partial \dot{\theta}_3} = m_3[(\tfrac{1}{3}l_3^2 + l_2 l_3 \cos\theta_3)\dot{\theta}_2 + \tfrac{1}{3}l_3^2\dot{\theta}_3]$$

Lastly, we derive the derivatives of the last three expression, with respect to time.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = \left(\frac{2m_1 l_1^2}{5}\right)\ddot{\theta}_1 + \frac{m_2}{2}[d^2 + (d + k\sin\theta_2)^2]\ddot{\theta}_1$$

$$+ \frac{m_3}{2}[(d + k\sin\theta_2)^2 + (d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))^2]\ddot{\theta}_1$$

$$+ m_2(d + k\sin\theta_2)(k\cos\theta_2)\dot{\theta}_1\dot{\theta}_2$$

$$+ m_3[(d + k\sin\theta_2)(k\cos\theta_2)\dot{\theta}_2 + [d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3)][(k\cos\theta_2)\dot{\theta}_2 + l_3\cos(\theta_2+\theta_3)\dot{\theta}_2]]\dot{\theta}_1$$

$$+ m_3[(d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))(l_3\cos(\theta_2+\theta_3))\dot{\theta}_3]\dot{\theta}_1$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = \left(\frac{1}{3}m_2 l_2^2\right)\ddot{\theta}_2$$

$$+ m_3\left[\left(l_2^2+\frac{1}{3}l_3^2+l_2 l_3\cos\theta_3\right)\ddot{\theta}_2 + \left(\frac{1}{3}l_3^2+l_2 l_3\cos\theta_3\right)\ddot{\theta}_3 + \left(-l_2 l_3\sin\theta_3\right)\left(\dot{\theta}_2\dot{\theta}_3 + \frac{1}{2}\dot{\theta}_3^2\right)\right]$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_3}\right) = \frac{1}{2}m_3\left[\left(\frac{2}{3}l_3^2 + l_2 l_3\cos\theta_3\right)\ddot{\theta}_2 + \frac{2}{3}l_3^2\ddot{\theta}_3 + \left(-l_2 l_3\sin\theta_3\right)\dot{\theta}_2\dot{\theta}_3\right]$$

These expressions can now be assembled according to Lagrange's equation.

$$Q_1 = \left[\left(\frac{2m_1 l_1^2}{5}\right) + \left(\frac{m_2}{2}+\frac{m_3}{2}\right)(d+k\sin\theta_2)^2 + \frac{m_2 d^2}{2} + \frac{m_3}{2}(d+k\sin\theta_2+l_3\sin(\theta_2+\theta_3))^2\right]\ddot{\theta}_1$$

$$+ \left[m_3\left[(d+k\sin\theta_2+l_3\sin(\theta_2+\theta_3))(l_3\cos(\theta_2+\theta_3)(\dot{\theta}_2+\dot{\theta}_3) + (k\cos\theta_2)\dot{\theta}_2\right)\right]\right.$$

$$\left. + (m_2+m_3)\left[(d+k\sin\theta_2)(k\cos\theta_2)\dot{\theta}_2\right]\right]\dot{\theta}_1$$

$$Q_2 = [(\tfrac{1}{3}m_2 + m_3)l_2^2 + m_3(\tfrac{1}{3}l_3^2 + l_2 l_3 \cos\theta_3)]\ddot{\theta}_2 + [m_3(\tfrac{1}{3}l_3^2 + l_2 l_3 \cos\theta_3)]\ddot{\theta}_3$$

$$+ (-m_3 l_2 l_3 \sin\theta_3)(\dot{\theta}_2\dot{\theta}_3 + \tfrac{1}{2}\dot{\theta}_3^2)$$

$$- [(\frac{m_2}{2} + \frac{m_3}{2})[k\cos\theta_2(d + k\sin\theta_2)]$$

$$+ \frac{m_3}{2}[(d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))(k\cos\theta_2 + l_3\cos(\theta_2+\theta_3))]]\dot{\theta}_1^2$$

$$+ g[m_2\frac{l_2}{2}\sin\theta_2 + m_3 l_2 \sin\theta_2 + m_3\frac{l_3}{2}\sin(\theta_2+\theta_3)]$$

$$Q_3 = [m_3(\tfrac{1}{3}l_3^2 + l_2 l_3\cos\theta_3)]\ddot{\theta}_2 + [\tfrac{1}{3}m_3 l_3^2]\ddot{\theta}_3$$

$$+ (\tfrac{1}{2}m_3 l_2 l_3 \sin\theta_3)\dot{\theta}_2^2$$

$$- \frac{m_3}{2}[(d + k\sin\theta_2 + l_3\sin(\theta_2+\theta_3))(l_3\cos(\theta_2+\theta_3))]\dot{\theta}_1^2$$

$$+ gm_3\frac{l_3}{2}\sin(\theta_2+\theta_3)$$

Recall, Q_i represents the torque applied at joint$_i$. Thus these last three equations are used to compute torques that must be applied to produce desired accelerations. Alternatively, these three equations are manipulated to allow computation of accelerations resulting from given applied torques.