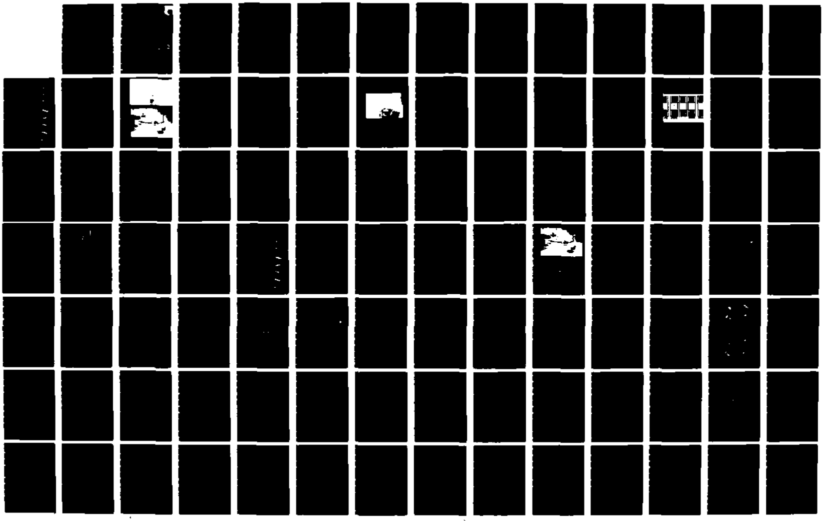
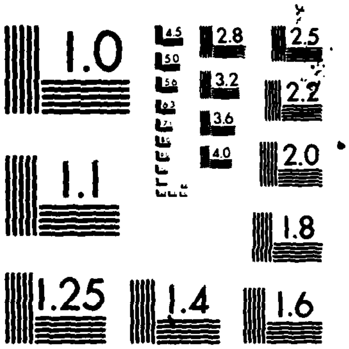
AD-A172 266 CENTER FOR AUTOMATION AND MANUFACTURING SCIENCE(U) 1/3
STANFORD UNIV CA R H CANNON ET AL. SEP 86
SPO-13649-01-00 AFWAL-TR-86-4053 F49620-82-C-0092
UNCLASSIFIED F/G 13/9 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFWAL-TR-86-4053

CENTER FOR AUTOMATION AND MANUFACTURING SCIENCE

Robert H. Cannon, Jr.
Thomas O. Binford
James D. Meindl

Stanford University
250 Durand Bldg
Stanford, California

September 1986

Final Report for Period September 1984 - September 1985

Approved for public release; distribution is unlimited

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

12 Sept 86

TED J. BRANDEWIE
Project Manager
Computer Integrated Mfg Branch

FOR THE COMMANDER

WALTER H. REIMANN
Acting Chief
Computer Integrated Mfg Branch

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/MLTC*, W-PAFB, OH 45433 to help us maintain a current mailing list.*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A172 266

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFWAL-TR-86-4053

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
CENTER FOR AUTOMATION AND MANUFACTURING SCIENCE

5. TYPE OF REPORT & PERIOD COVERED
Final Report
Sept., 1984-Sept., 1985

6. PERFORMING ORG. REPORT NUMBER
SPO 13649-01-00

7. AUTHOR(s)
Principal Investigators: Robert H. Cannon, Jr., Thomas O. Binford, James D. Meindl,

8. CONTRACT OR GRANT NUMBER(s)
F49620-82-C-0092

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Stanford University
250 Durand Bldg.
Stanford, California 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61102F
2306/A3

11. CONTROLLING OFFICE NAME AND ADDRESS
U.S. Air Force
Office of Scientific Research
Building 410, Bolling AFB, D.C. 20332

12. REPORT DATE
September 1986

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Air Force Wright Aeronautical Laboratories
Materials Laboratory (AFWAL/MLTC)
Wright-Patterson AFB OH 45433-6533

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Robotics, Vision, Assembly, Automation, Manipulator, Tactile Sensing, Inspection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See next page.

SECURITY CLASSIF

Abstract

Air Force Office of Scientific Research support of a Center of Excellence at Stanford University has provided the impetus and core for a major new entity, the Center for Automation and Manufacturing Science (CAMS). The new center draws from two well-known research groups at Stanford: the Robotics group of Stanford's Artificial Intelligence Laboratory and the Automatic Control Group of Stanford's Department of Aeronautics and Astronautics. Six professors and some 35 graduate students are participating in CAMS activities. CAMS in turn has become the first of a new complex of four centers at Stanford involved in the manufacturing enterprise: the Stanford Institute for Manufacturing and Automation (SIMA). Strong industrial interaction is a primary objective of SIMA.

In our AFOSR program we have focused on robotic aspects of automation. Our goal is to make fundamental contributions to the underlying *set* of technologies that will enable the next generation of industrial robots to be far more capable than today's — will enable them to be lightweight, limber, deft, facile, quick, friendly, low-powered, seeing, sensing, thinking machines that can reason and strategize — can carry out tasks assigned at a high conceptual level.

Specifically, our research focus is on fast, precise control of lightweight (flexible) manipulators, on sensing, especially optical and tactile sensing, on intelligent systems for robot task management, and on computer vision for robot management.

In the context of a gratifyingly large number of advances we have been able to make in robotics over these three years, some major research results achieved under this contract include the following:

Task 1: Survey of Key Problems and Technology Transfer

We have identified, and pursued toward solution in our laboratories, a set of key technical problems whose resolution will enable major advances in the capability of automation using robots. These key problems are in the areas of intelligent robotic systems (including task management, vision, and system integration) of fast, precise control of very lightweight, flexible robotic manipulators, and of advanced tactile sensing; and our contributions to their solution are enumerated below.

We have committed our resources to ongoing technology transfer through vigorous interaction with a set of major aerospace manufacturing companies including the 13 sponsor companies of our formal structure, SIMA.

Task 2: Intelligent Systems in Manufacturing, Inspection and Vision

a. A system of real-time collision avoidance was implemented. The system is based on the use of potential functions around obstacles. Obstacles are described by composition of primitives which are approximately cylinders and blocks. The method requires a small amount of calculation; it allows obstacle avoidance to occur in real time as an integral part of the servo-control. An experimental manipulator programming system "COSMOS" using the method has been designed for the PUMA and demonstrated

with obstacles (including mobile obstacles) detected by an MIC vision module.

b. A dynamic simulator was implemented as a software equivalent of a robot arm. So far, the simulator has been used to test three adaptive control schemes, several dynamic control schemes, and two parameter estimation schemes.

c. A new, nonlinear, and generalizable technique has been developed that will continually monitor the parameters of a robot arm to estimate continuously the inertial forces and the friction in robot joints. This system has also been simulated for a three-link robot, and has been successfully applied to a physical single link robot as well.

d. We have completed joint force sensing for one joint of the PUMA, and have designed a touch sensing finger which senses three components of force.

e. Contributions have been made toward a successor for ACRONYM, an intelligent system developed at Stanford and adopted by about a dozen laboratories and companies. The modeling system of SUCCESSOR is greatly generalized to include multiple naming, holes and set operations on volumes (union, intersection, difference). Work in other areas of computer vision includes architecture of VLSI vision processors, segmentation with edge operators, graphics support, and hardware support.

f. The Automatic Task Level Assembly Synthesizer (ATLAS) was advanced significantly in power and usability. A high level task specification model was developed and demonstrated; advances were made in fine motion synthesis and constraint integration.

g. Several portable versions of LISP have been developed and both utilized at Stanford and, indeed, transported. These include subset SLISP for VAX systems and TAIL for SUN work stations.

h. Both software and hardware support for vision systems include: software for interfacing the Grinnell display; LISP graphics; interface for an expensive TV input system; convolution software; TV time base corrector; an interface for an Optronics drum scanner; work on software for a GTCO digitizing tablet; and an interface for an image hard copy output device.

i. Implementation of a new edge operator, tests of shape from a shading algorithm, and experimentation toward building an active ranging device are underway.

Task 3: Rapid, Precise Control of Nonrigid Manipulators

The goal is to provide manipulator control so good that a whole new generation of manipulators can be developed — manipulators that are much lighter and far more facile than anything today's control systems could stably manage. To do this we are developing a sequential family of new manipulators that are extremely light and flexible, deliberately exaggerating the control problem so that it will have to be solved in much more fundamental ways than it ever has before.

The central control problem for each of these manipulator systems is the problem of controlling the end-point (fingertips) of a manipulator by measuring position or force *at that point* and using *that* measurement to control torque at an actuator at the other end (elbow or shoulder) of the flexible manipulator. This turns out to be, for fundamental stability reasons, *very hard to do.*

a. For very flexible manipulator arms (two-second vibration period), we have succeeded in achieving control that is not only stable but highly robust, and at a speed limited basically by wave propagation times in the manipulators themselves.

b. A new technology, the "quick wrist" was developed for enhancing (by a factor of 10) the within-work-station task speed of a long, very flexible robotic manipulator. Again, high-bandwidth end-point feedback control was accomplished for the first time, and rapid pick-and place and slew-and snatch maneuvers have been demonstrated.

c. A light-weight experimental two-link arm was designed and built with very flexible tendons (representing drive-train flexibility). High-bandwidth control using end-point feedback has now been accomplished for the first time and shown to be more precise and ten times faster than conventional joint control. A whole sequence of key basic developments and experiments are made possible by this equipment, and are now underway.

d. Two sets of fundamental contributions were made to strategy for minimum-time control of generic pick-and-place robot maneuvers, and for chasing and capturing a moving, swinging part.

e. Three new adaptive control schemes have been developed, and two of them demonstrated experimentally at this writing. More importantly, new discoveries have been made about the dominance of noise in real-time dynamic identification, and new means developed and demonstrated for dealing with it.

Task 4: Ultraminiature Tactile Pressure Sensor Arrays

a. A millimeter-sized silicon pressure transducer (CPT) prototype for a tactile array element that is integrated with a silicon integrated circuit system has been designed, and the new micromachining techniques required for its fabrication have been invented and demonstrated.

b. A custom integrated circuit for (and to be integrated with) the silicon capacitive pressure sensor has been built and tested. The accuracy is ± 0.7 mm Hg over a range of 300 mm Hg.

c. An integrated circuit for Tactile Array Interpretation has been designed and its circuit blocks simulated successfully. An Asynchronous Digital Multiplexer (ADM) implements a novel system architecture.

Even more important than the specific research accomplishments listed above is the the achievement of the Ph.D. degree by 10 of our CAMS students (5 with AFOSR support), who now have deep experience and interest in aerospace and manufacturing automation. So far, 8 of the graduates from CAMS have joined 7 major aerospace companies. In the "pipeline" are currently additional students engaged in Ph.D. research in our CAMS laboratories; and we expect most of them to follow the initial ten into the aerospace community.

Table of Contents

Abstract i

Table of Contents iv

Introduction v

Technical Report 1

Task 1. Survey of Key Problems and Technology Transfer 1

Task 2. Intelligent Systems for Manufacturing; Inspection and Vision 6

Overview 6

Planning 7

Inspection and Vision 11

Prediction 12

References and Publications 16

Task 3. Rapid, Precise End-Point Control of Nonrigid Manipulators 17

a. Two-Link Manipulator with Flexible Tendons 23

b. Very Flexible Manipulator with Fast Wrist 55

c. Strategies for Task Command and Control of Two-Link Manipulators . 97

d. Adaptive Control 99

References and Publications 132

Task 4. Integrated Tactile Systems 133

Objective, Design Goals and Development Plan 133

Integrated Circuit for a Capacitve Pressure Transducer 134

An Integrated Circuit for Tactile Array System Management 136

Micromachining 146

Related Publications 149

Appendix A – Personnel 151

Appendix B – Statement of Work 153

Appendix C – Predicting Specular Features 155

Appendix D – On the Design of a Geometric Modeling System 165

Appendix E – Sensor Modeling for Prediction 173

Appendix F – Summary for 85 177

Appendix G – Minimum-Time Control of a Two-Link Robot Arm 187

INTRODUCTION

In 1982 we were privileged to be invited by the U.S. Air Force Office of Scientific Research to propose to establish at Stanford University, with major AFOSR funding, a Center of Excellence in Manufacturing and Automation. This report describes the development of our Center, the Stanford Center for Automation and Manufacturing Science (CAMS), and the accomplishments that the AFOSR funding have already made possible during its first three years.

Most notably, the vision of the original AFOSR creators has been more than fulfilled in the form of a new entity — the Stanford Institute for Manufacturing and Automation (SIMA) — whose scope goes manifold beyond the original AFOSR plan. For, as hoped, the AFOSR funding, complemented by a major grant from IBM for master's-level training in manufacturing management, has stimulated the establishment of not one, but four centers in manufacturing-related high technology, the four being affiliated through SIMA which now counts 13 corporations among its sponsors.

Thus, while the purpose of the present report is to summarize the first three years of AFOSR sponsored accomplishments in CAMS, we do want to call attention first to the larger scope of SIMA which the AFOSR sponsorship has helped engender, and which was equally part of the AFOSR vision. The structure and scope of SIMA are described under Task 1, along with a list of its 13 corporate members, and the quite vigorous interactions we are having with them.

To focus now on CAMS, per se: the goals of this center, started by AFOSR, are to contribute a series of key advances to the technology areas that will be central to the next generation of intelligent machines and robots — basic advances that will in fact be *enabling* to the next generation.

The new Center draws from two internationally known research groups at Stanford: the Robotics Group of Stanford's Artificial Intelligence Laboratory, and the Automatic Control Group of Stanford's Department of Aeronautics and Astronautics. A common objective that we are able to address together, with this major Air Force support, is to advance the effectiveness of automation in manufacturing by mounting research concurrently — and synergistically — into a set of the primary, pacing technologies in automation, as we outline below.

In the laboratories of CAMS, some 35 graduate students carry out Ph.D. level research on selected projects in intelligent machines and robotics, working with 6 professors and 8 professionals and technical staff.

In our Air Force program we have decided to focus on robotic aspects of automation. With their requirements for greater flexibility of use and rapid task redirection, the robotic aspect of automated manufacturing will draw upon more of the new technologies, and more deeply, than any other aspect.

If the right *set of technologies* is developed, we believe the next generation of robots can (by comparison with today's) be lightweight, limber, deft, facile, quick, friendly, low-powered, seeing, sensing, thinking machines. Above all, they will be capable of reasoning and strategizing — of carrying out tasks assigned at a high conceptual level by "thinking through" the best way to carry out any given task. Robotic devices with such characteristics and capability can provide the flexible automation that will be so important in achieving higher levels of productivity.

What are the underlying technologies that will be needed as the base for robots with such capabilities? They can be described in four categories: manipulator control, sensing, thinking, vision. Among us, in our Center, we are working to make useful contributions in all four technical areas. There is, of course, much interaction between, and synergism among the four areas; and that is the exciting thing about the level of effort that the AFOSR program makes possible. Specifically, fast, precise manipulator control is the primary focus of Task 3 of the program, tactile sensing of Task 4, and computer-based thinking and vision of Task 2. But these depend upon each other altogether as diagramed in Fig. 1, and draw upon one another in many ways. We feed back signals from many sensors — optical and eventually perhaps acoustic, as well as tactile and force — to effect good end-point control of manipulators. New, more competent manipulators, with their multiple sensors, will be utilized avidly by task-management systems to produce new assembly sequences that are quicker, more precise, and more efficient.

More acute robot vision, together with more rapid visual *perception* (scene analysis) are very important basics for more effective task planning, and may even someday be used in real time by the fast manipulator controllers themselves.

And of course in an ubiquitous way, our overview task, Task 1, of the program has drawn totally upon — and stimulated — all of the other tasks. For Task 1 is to work with the industrial automation community to survey in a continuing way the problems of automation, and to effect technology transfer to designers of new automation systems as each advance becomes demonstrable in our laboratories. Thus while, for convenience, we discuss the four research Task areas separately in the following paragraphs, and as separate sections of this report, they are in fact highly interrelated in our Center.

Under direct AFOSR support, there has taken place in the Stanford Artificial Intelligence Laboratory (SAIL) basic development of more effective languages for managing robot tasks now culminating in the SUCCESSOR system and including new work in extending prediction to a variety of sensors and multisensor tasks, and work on vision that now provides a firm basis for prediction and interpretation of image data.

In the Aerospace Robotics Laboratory (ARL), we have constructed a new facility for developing lightweight, very flexible, multiple-link manipulators; and we have demonstrated experimentally (for the first time) solution to the very difficult problem of controlling the motions of such manipulators quickly and precisely using end-point feedback. We have also demonstrated a new concept, the end-point-controlled quick wrist, which has dramatic potential for a new level of performance by future robots. Also, we have been making important advances in adaptive control, achieving (for the first time) *experimental* real-time high-bandwidth adaptive control of dynamic systems having lightly-damped vibrating natural modes. We hope to make major new strides of a fundamental kind in this area using the experimental systems our Air Force funding has allowed us to develop. We have also made a good start on trajectory management strategies which will be important as we apply our control capability to multiple-link and cooperating robot tasks requiring

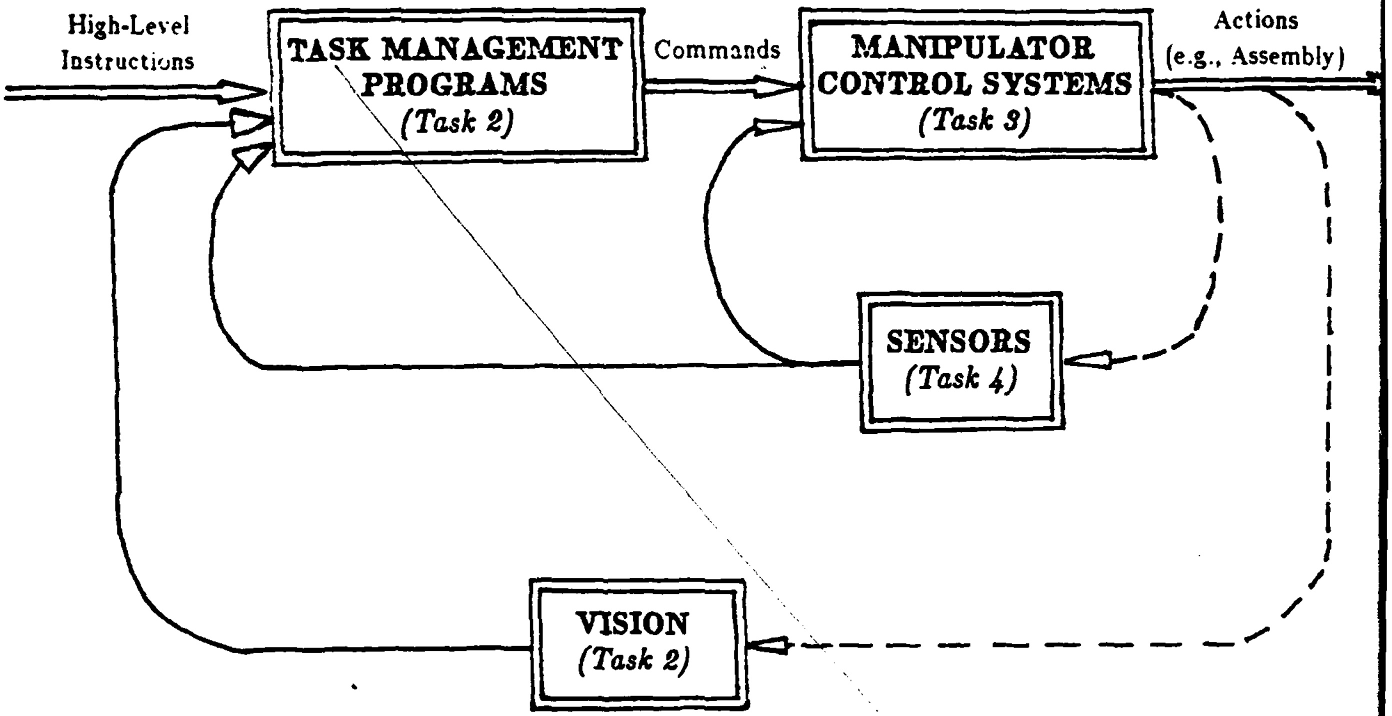


Fig. 1 The Anatomy of a High-Level Robot

high speed and precision.

Rapid Precise Control of Nonrigid Manipulators

Manipulators — arms and hands with their actuators, and the increasingly sophisticated feedback systems that control their movements — are the "business end" of a robot, where parts are mechanically moved, formed, placed, fitted together (or dismantled), and inspected. In our Controls Group we are addressing the question of how to provide manipulator control so good that a whole new generation of manipulators can be developed — manipulators that are much lighter and far more facile than anything today's control systems could stably manage.

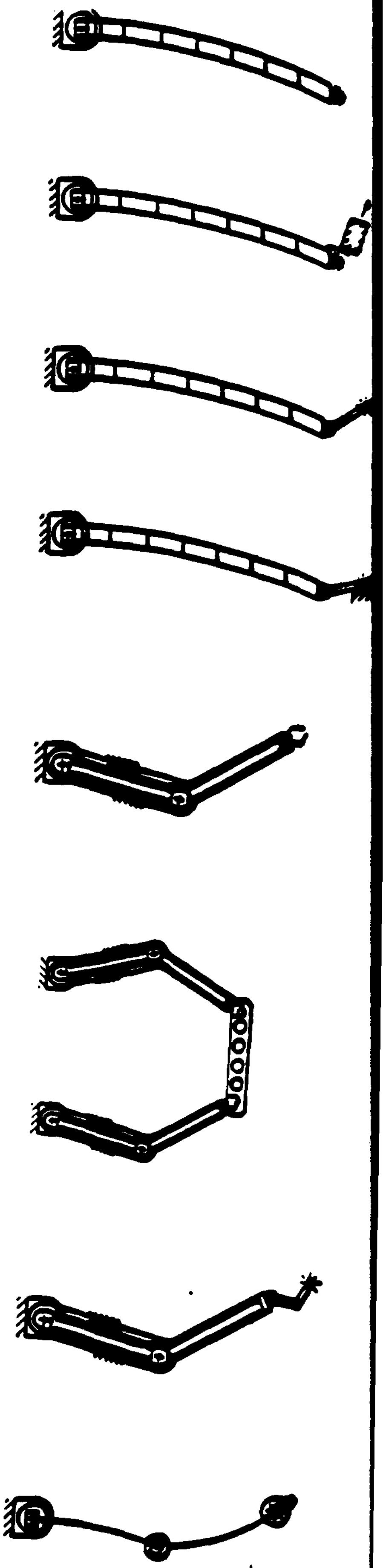
To do this we have begun to develop a sequential family of new manipulators that are extremely light and flexible, deliberately exaggerating the control problem so that it will have to be solved in much more fundamental ways than it ever has before. Some members of our family of manipulators are shown in Fig. 2. The family will also extend in many ways not shown — to three dimensional action, and to mobile-mounted very flexible manipulator systems, for example.

The central control problem for each of these manipulator systems — and the problem we have been the first to solve — is the problem of controlling the end-point (fingertips) of a manipulator by measuring position force *at that point* and using *that* measurement to control torque at an actuator at the other end (elbow or shoulder) of the flexible manipulator. This turns out to be, for fundamental stability reasons, *very hard to do.* Every time someone has tried it (this *non-colocated* control) in commercial robots, the robot system has gone unstable.

Using advanced control methods developed in our laboratory we have already succeeded in achieving, for the first four configurations of Fig. 2, control that is not only stable but highly robust, and at a speed limited basically by wave propagation times in the manipulators themselves. (The achievement for the system in Fig. 2a was by Eric Schmitz, for Fig. 2b by James Maples, and for Fig. 2c Scott Tilley under other funding.)

By proceeding step-wise with the sequence of basic manipulator control problems indicated in Fig. 2, we expect to provide the fundamental new technology for controlling a new generation of lightweight flexible robots.

In the first year of this contract we developed two important new experimental systems: the two-link arm with flexible tendons and the very flexible arm with quick wrist. In the second year we achieved remarkable performance with the latter, and succeeded in our first successful closed-loop control of the former. We also developed two preliminary control strategies for achieving the full potential of the two-link arm. In this final year we have brought the two-link arm close to the very fast, precise, end-point control of which we know it to be capable: with end-point control we have already achieved stable response to position commands that is 10 times as fast as is possible with conventional joint-angle control. These research directions continue: to move into 3-D, we have added a vertical-axis control with gripper to the two-link arm (visible in Fig. 4) with which we will carry out a series of carefully designed pick-and-place and moving-target-capture tasks. We will also pursue extending the valuable quick-wrist capability to two dimensions, both as indicated



(a) Very Flexible One-Link Manipulator
(Rapid pick and place.)

(b) Very Flexible Manipulator with Force Control
(Slew and Touch moving target.)

(c) Flexible Manipulator with Fast Wrist
(Precise snatch and place.)

(d) Flexible Manipulator with Fast Wrist
(Slew and Precision touch.)

(e) Two-Link Arm with Elastic Tendons
(2D pick and place; slew and position touch.)

(f) Cooperating Two-Link Arms
(Long-part handling.)

(g) Two-Link Arm with Double Wrist
(Very fast precise 2D tasks.)

(h) Two-Flexible-Link Arm

Fig. 2 Sequence of Expermental systems,
(A through E built and tested; F through H under development)

in Fig. 2g, and also for the very flexible arm of Fig. 2d. (We will also look at the "R-O" alternative configuration.)

We have begun detailed design for a new apparatus, Fig. 2h, with which to uncover and solve the problems of controlling multiple very flexible arms. In these first fundamental studies we will concentrate on the flexibility problem, "deferring" gravity by supporting the elbow and wrist on air cushions. (We will accommodate gravity subsequently, as we did in going from the systems of Fig. 2a-d, to those of Fig. 2e-g.)

A primary future thrust will be developing adaptive control systems for our physical robot experimental systems, one after another, beginning with Fig. 2d. Adaptive control has received much lip service and a great deal of theoretical treatment and computer simulation. But its use in controlling actual physical systems is so far confined to very slow chemical plants: applications to high speed systems, especially vibrating ones, are essentially nonexistent. There are good reasons why successes in computer simulation have not translated to any real-world success. We intend to make progress in this area that is so important to fast control of *lightweight* robots. We have developed a special experimental four-disk system with very low damping $\zeta < .001$ that provides a most severe test for any adaptive control scheme that uses end-point feedback. We now use this as our first testing ground before proceeding to adaptive control of the various manipulators in Fig. 2. During the current year, we completed the testing of our second and third adaptive control systems on the vibrating-disk apparatus; and we are now applying one of them to the very flexible manipulator.

Fast Wrist on a Flexible Arm

The advance of Fig. 2c, development of a small quick-acting wrist mounted on a very flexible long arm, was accomplished by Wen Wie Chiang entirely with AFOSR funding. This new system, shown in Fig. 3, has several generic implications for future robot systems, which can be inferred from the experimental performance represented in Fig. 5.

The wrist, a short link 17 cm. long, is installed at the tip of a 97 cm. flexible beam, Figs. 3 and 5a. The wrist is light and rigid compared with the flexible beam, and is controlled by a separate DC motor. A lamp is mounted at the tip of the wrist to indicate the end-point position (where various end effectors are to be mounted). A silicon photo sensor mounted above the apparatus senses the position of the lamp, and this tip position is the primary signal for controlling tip position.

Figure 5b is a time sequence of the system moving between two workstations. Generically, when a robot manipulator is used either in a fabrication or assembly job, the area of workstation is small compared with the reachable region of the manipulator, a region that may include several stations. To achieve the most efficient operation, the manipulator has to move rapidly from station to station, but at the same time be able to perform tasks within a station under accurate, very-high-bandwidth control.

A rigid and heavy manipulator cannot achieve high speed and bandwidth with a reasonable amount of power consumption. A lighter large manipulator can be moved faster when its flexibility is under proper control, as we have demonstrated (Fig. 2a); but the maximum bandwidth of the closed loop is still limited by its flexibility, i.e., within the

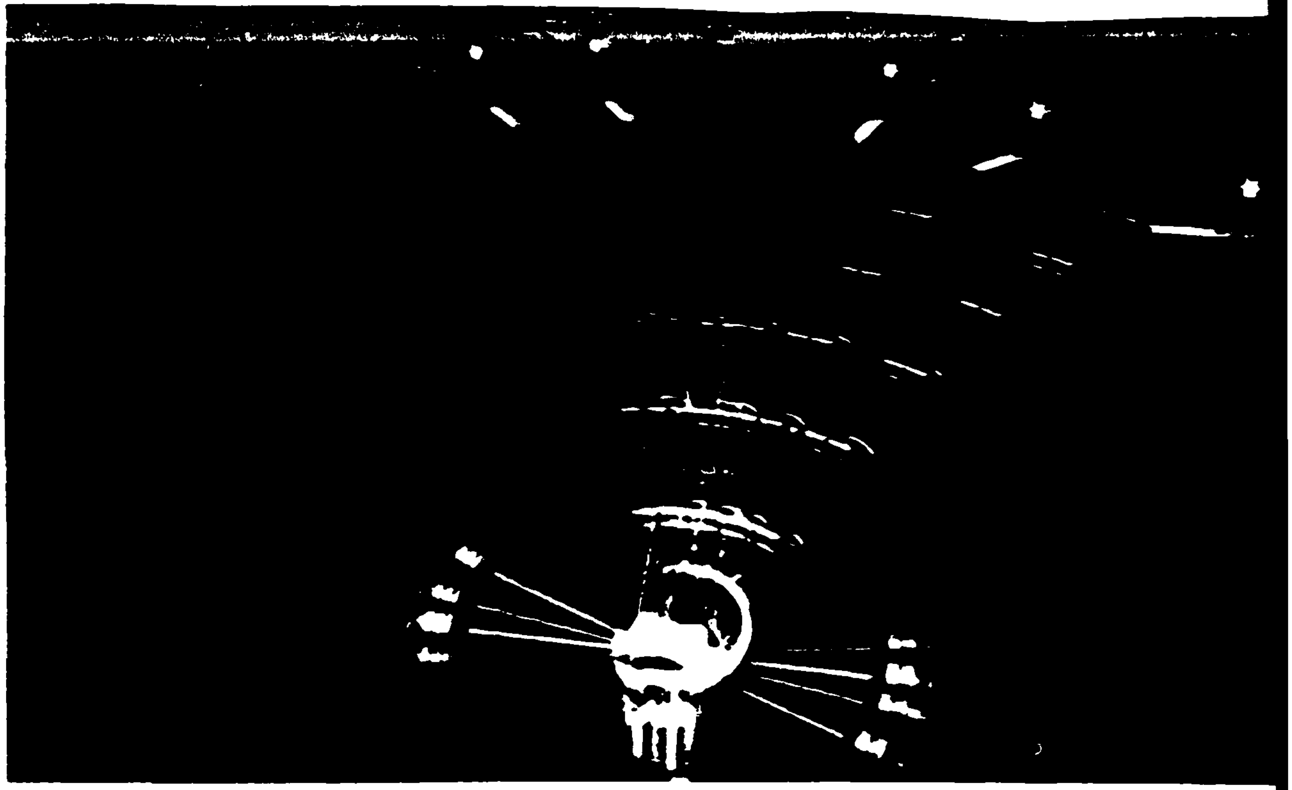


Fig. 3 Very Flexible Manipulator with Quick-Acting Wrist.

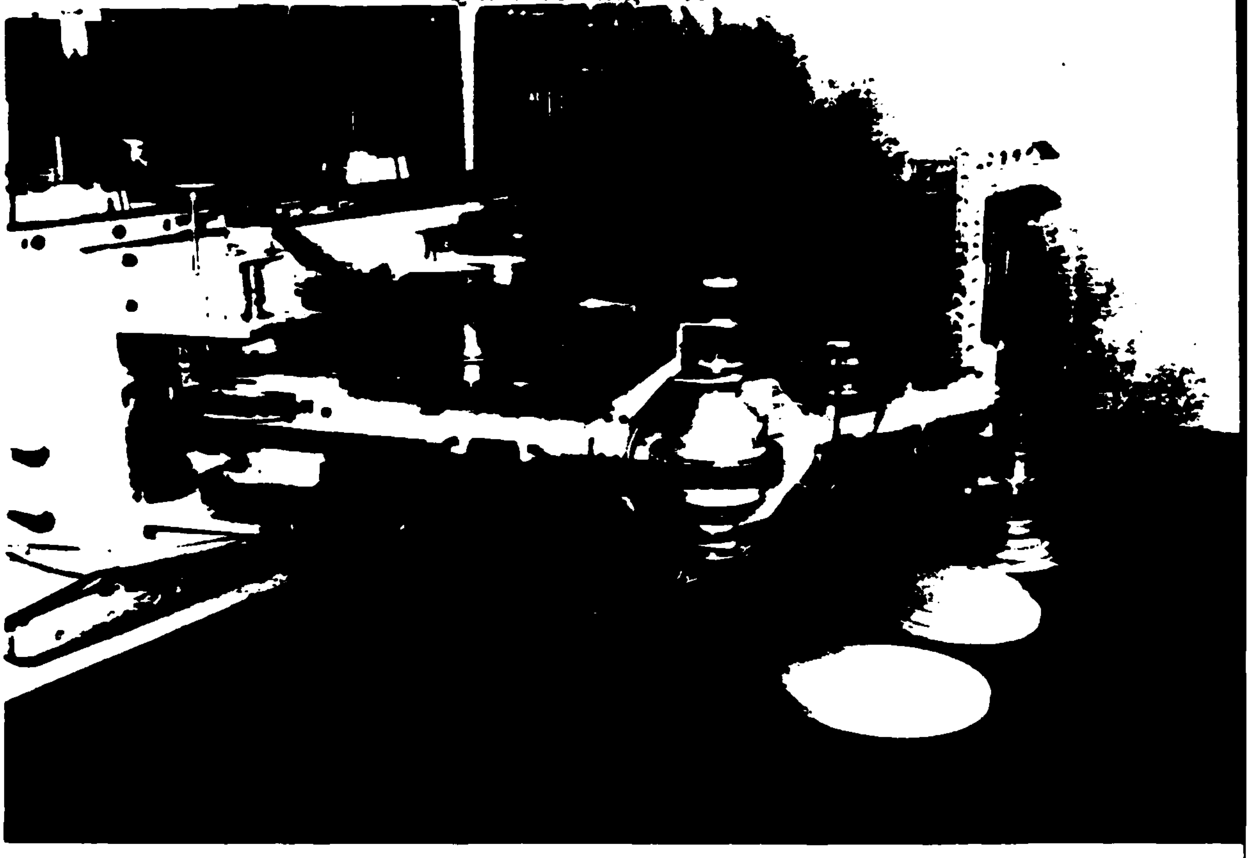


Fig. 4 Photograph of the Stanford Two-Link Manipulator with Flexible Tendons

immediate vicinity of the end of the larger manipulator.

Figure 5 presents an experimental demonstration of how the combination works to get optimum system performance. In Fig. 5b, the long flexible arm moves the wrist from one workstation to the next. In Fig. 5c and d, a commanded change in the tip position within the workstation is accomplished by the fast wrist almost instantaneously (A,B), and that tip position is held rigidly in space while the slower flexible-arm end is brought into alignment (C,D,E). (It is striking to watch the tip obey commands in space so precisely, regardless of motion of its supporting links.)

Two-Link Strategy

Under funding by DARPA we have developed capability for *force* control of flexible manipulators (Fig. 2b and d) using end-point feedback, and are now building a new experimental system with which we plan to develop the control technology for *intimately cooperating* manipulators, Fig. 2f. With funding from NASA we are concentrating on mobile robot control. While our focus is on the simulation of space robots (using air cushion vehicles), the implications for mobile robots in a manufacturing environment are quite direct.

It may be helpful to have a clear picture of how our AFOSR-funded research projects interact with other projects in our laboratory. Such an overview is presented in Fig. 3-1.

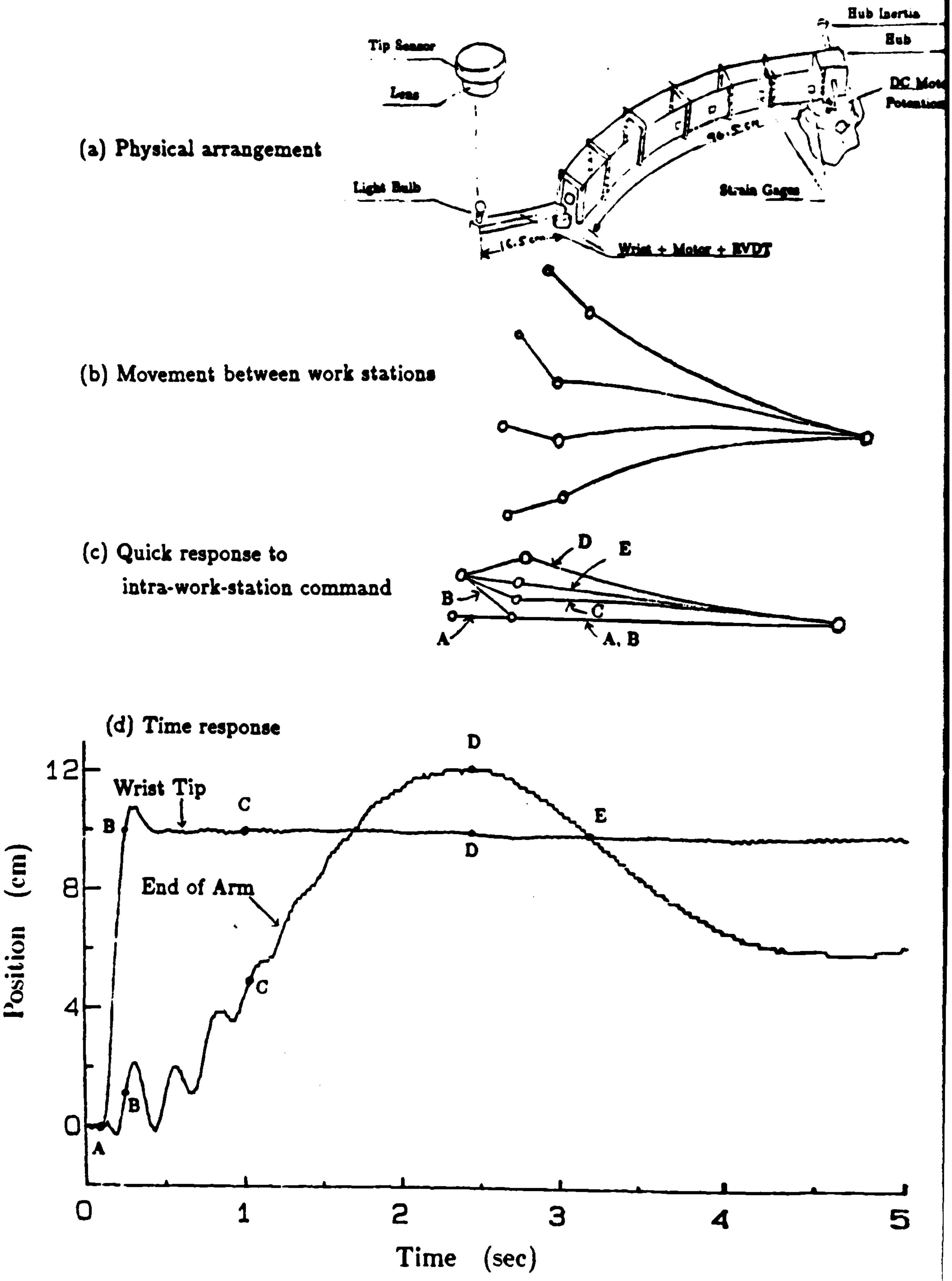
Intelligent Systems for Robot Management and Vision

We developed a new version of AL, a programming system which is portable among computers and between robots, and which demonstrates capabilities for programming manufacturing systems also. Under this contract, a new version of AL was brought into operation and implemented major parts of the AL user interface, a syntactic editor and symbolic debugger. The new version included the following features: (1) graphics for forces and dynamics, new syntax to allow AL programs to make use of recently added abilities in the arm servo code, especially incremental motions, individual joint motions, and joint sensing; (2) reading force vectors from the force wrist in arbitrary coordinate systems; (3) a variety of new motion clauses to specify desired configurations such as straight-line (Cartesian) motions; and (4) a SAY statement to access the speech synthesiser from AL programs. Subsequently, under other support, AL and COSMOS systems have become well developed and advanced.

During this contract, research on programming systems began with intermediate level systems for programming robots, i.e., the AL system and the COSMOS system for force control and collision avoidance.

In the second year, the effort shifted to intelligent systems for planning robot actions and for planning inspection and vision tasks. These systems with geometric models and geometric reasoning have a contribution for general manufacturing, beyond those tasks we investigate here.

The ATLAS system was begun to investigate general methods for planning robot



(a) Physical arrangement

(b) Movement between work stations

(c) Quick response to intra-work-station command

(d) Time response

Fig. 5 Very Flexible Manipulator with Quick-Acting Wrist.

actions. The effort now focuses on synthesis of sensor-controlled fine motion. It is planned to couple ATLAS with Cosmos or AL, putting out fine motion commands for force control.

An intelligent system for planning vision, SUCCESSOR, incorporates geometric models and reasoning about geometry. Progress has been made toward a powerful modeling system with generic and individual models made of generalized primitives linked by set operations; a very general intersection program is being developed. A system for building models was implemented.

Prediction mechanisms for making use of expectations about images and measurements have been developed. Symbolic graphics provide detailed methods for individual objects in fixed states. A sensor modeling system is being developed for sensory integration and multi-sensor tasks. A study of specularity has led to methods for predictions of specularities and for use of specularities to estimate surface properties. Theoretical studies have determined a generic observability model and a family of quasi-invariant observables for prediction and recognition of metric properties of objects.

A system for real-time collision avoidance was implemented by Khatib. The system is based on the use of potential functions around obstacles. Obstacles are described by composition of primitives which are approximately cylinders and blocks. The method requires a small amount of calculation; it allows obstacle avoidance to occur in real time as an integral part of the servo-control.

Previous research in obstacle avoidance has focused on the development of path planning algorithms, aimed at providing a free Cartesian path for the manipulator. A coordinate transformer generates the joint-space path for the servo-control level. The high complexity of computation in such approaches leads to serious problems in real time control of a manipulator meant to act in a complex and evolving environment.

We have developed a new control scheme based on the construction of a dynamic model of a manipulator in operational space (task-space) rather than joint-space. This allows a simple force transformation to replace the difficult conversion of the Cartesian path into joint coordinates. The philosophy of this approach is to say that the manipulator moves in a field of forces. The position to be reached is an attractive pole of the end effector, and obstacles are repulsive surfaces for the manipulator parts. Controlling a given point of the manipulator with respect to several obstacles is resolved directly by the addition of the relevant potentials.

An experimental manipulator programming system "COSMOS" (Control in Operational Space of a Manipulator-with-Obstacles System) has been designed for implementation of the method for PUMA arms. Demonstration of motions with obstacles (including mobile obstacles) detected by an MIC (Machine Intelligence Corporation) vision module have been performed, Fig. 6.

Assembly Robot Tools and Subsystems

A dynamic simulator was implemented by Burdick. The simulator has been used in verifying analyses of control algorithms, and in Burdick's estimation of inertia and friction. The simulator is essentially a software equivalent of a robot arm. The simulator takes in

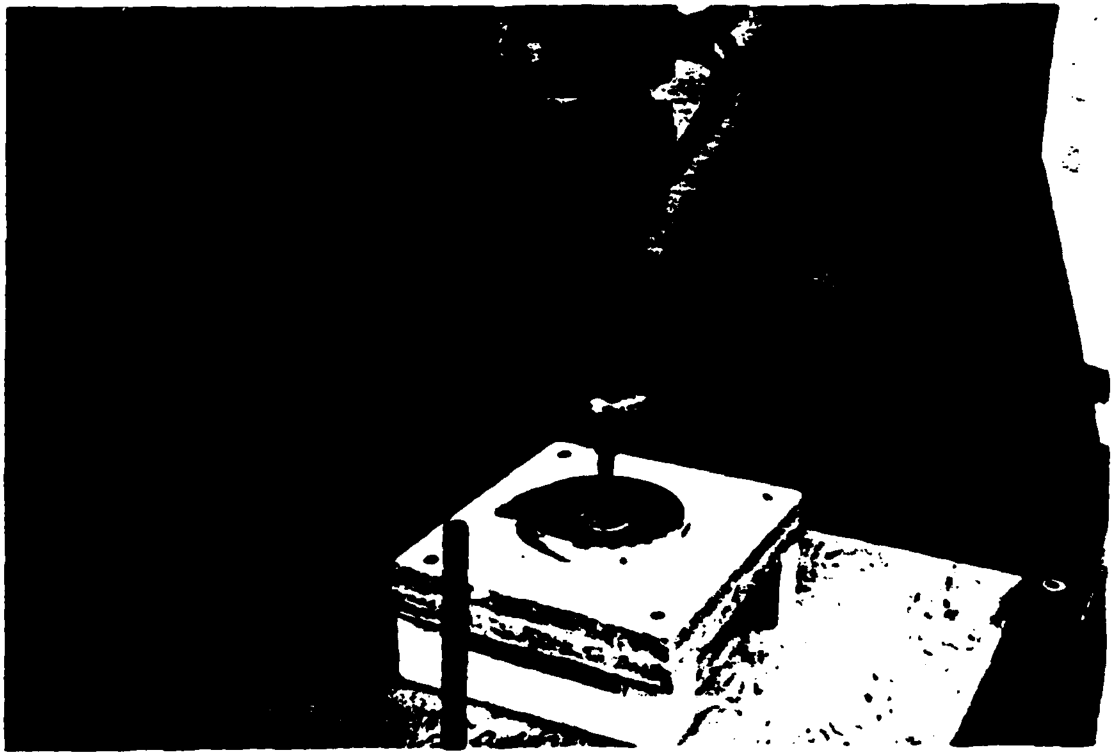


Fig. 6 Demonstration of Assembly Program Written for `the RS-1 Robot: Assembling an Electric Motor.

torque commands from a control system program, and integrates the dynamic equations of motion (which are developed using Lagrangian mechanics) to yield joint accelerations, velocities, and positions. The simulator is currently set to simulate the PUMA 560 robot, since most of the experimental work in controls at Stanford uses this robot.

An interactive graphics package was developed to plot the data generated from simulation. Involving the human mind to find patterns in visual data greatly improves the quality and efficiency of the data analysis. So far, the simulator has been used to test three adaptive control schemes, several dynamic control schemes, and two parameter estimation schemes — with considerable time saving as compared to implementing and debugging these schemes on a real robot arm.

A new, nonlinear, and generalisable technique has been developed that will continually monitor the parameters of a robot arm and update them as they change during operation. The technique measures the amount of torque applied during operation and tracks the robot's response to the input. Using the errors between the actual and desired trajectory and a general nonlinear model of the robot arm, the estimation scheme updates the parameter values. The technique has been successfully applied to a physical single link robot.

Another technique has also been developed to estimate continuously the friction in robot joints. The system has been simulated for a three-link robot, and has been successfully applied to a physical single link robot as well.

Inspection and Vision

Research in vision has a focus on intelligent systems which support not only inspection and vision but the total robotics and manufacturing research program. Contributions have been made toward a successor for ACRONYM, an intelligent system developed at Stanford and adopted by about a dozen laboratories and companies (Brooks 82).

Work has been carried out in other areas of computer vision, including architecture of VLSI vision processors, segmentation with edge operators, graphic support, and hardware support.

Work on extending ACRONYM and developing SUCCESSOR has included several projects. A geometric editor has been designed and implemented by Rublee. Parts of the modeling system of SUCCESSOR have been designed and initial implementation has been begun by Cowan. The modeling system is greatly generalized over ACRONYM to include multiple naming, holes and set operations on volumes (union, intersection, difference). Binford, Brooks, Triendl, Dreschler, and Takamura are involved with the system and design of this system were done under other support.

Implementation of a new edge operator is being done by Triendl. Lim has made tests of shape from a shading algorithm by Pentland to assess its suitability for integration in ACRONYM or SUCCESSOR. Wells has begun design and experimentation toward building an active ranging device.

Both software and hardware support for vision systems have been provided. They include: software for interfacing the Grinnell display; LISP graphics by Selker; interface for an inexpensive TV input system by Imaging Technology Inc. by Wells; TV time base corrector in progress by Rousso; an interface fcr an Optronics drum scanner by Fitzhugh; work on software for a GTCO digitizing tablet; and an interface for an image hard copy output device.

JED is an interactive geometric model editor used for creating and editing three dimensional models of objects. The editor allows one to describe and display a wide range of generalized cones and orient these primitive shapes to form more complex assemblies.

Sensors for Robot Systems

New sensors will of course be extremely important components of new, more capable robots. In our early manipulator research we are using simple sensors — e.g., the light-bulb/optical sensor shown in Fig. 5a and c — in order to move Task 3 forward in parallel with the development of new sensors. At the same time, in Task 4 of this AFSOR program and in other projects, we are also sponsoring and collaborating with sensor development efforts per se.

The sensor system described under Task 4 of this report is for perceiving by touch, the shape or texture of objects.

Spatial information is to be generated by an *array* of capacitive pressure sensors, each as in Fig. 7, data from which are to be multiplexed together for transmission to a remote site. Electrically, a bandwidth of 1 - 500 Hz and a response time < 0.1 sec. were stipulated. Mechanically, a touch sensor must provide a spatial resolution of < 2 mm., and must be reliable, easy to repair, and self-protecting.

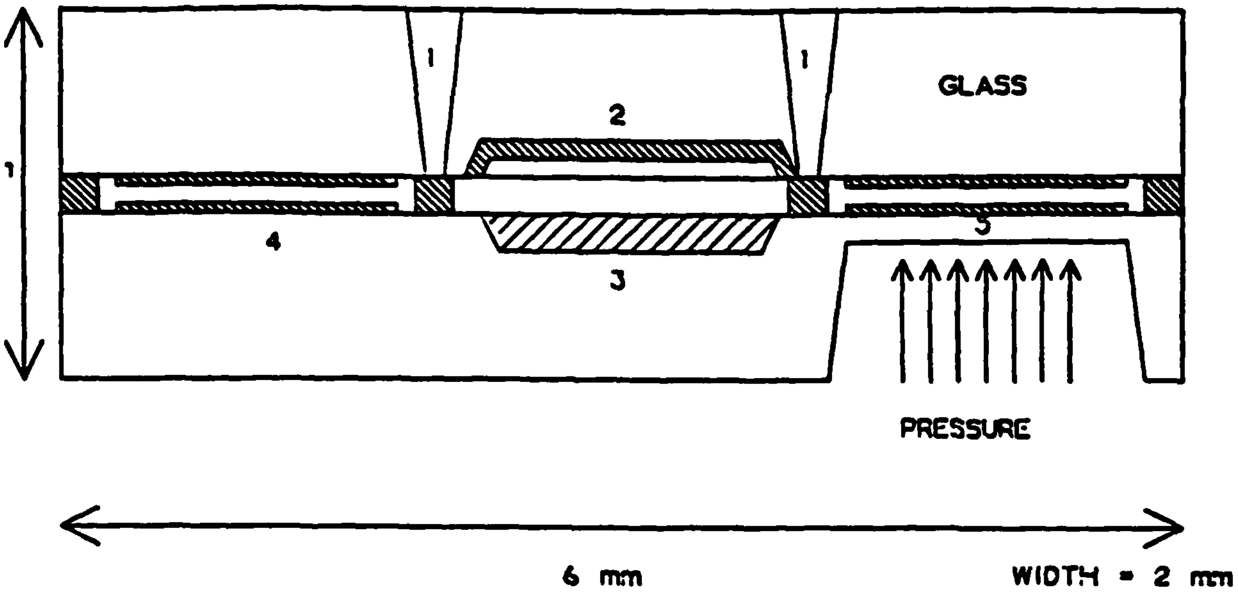
In developing such a touch sensor, such sub-projects include: a) an electromechanical silicon transducer; b) an integrated circuit for signal processing at the site of data collection; c) merger of electromechanical transducer and the on-site processing circuitry into an hermetically packaged integrated silicon sensor; d) an integrated circuit for multiplexing signals from an array of such integrated sensors; and e) incorporation of the integrated array into robotic skin.

During the first year, effort has focused on the development of the required integrated sensors and circuits.

A capacitator was chosen as the transduction element due to its superior sensitivity over piezoresistive pressure sensors. Special micromachining techniques are needed for forming this structure, including chemical etching, laser drilling and welding, micro-sandblasting, and electrostatic bonding. An automated laser workstation for micromachining glass wafers has been developed and used to produce arrays of electrical vias for electrical connection between cables attached to the top of the glass cap and the integrated circuit under the glass cap. In addition, a set of photomasks for a Mechanical Test Chip (MTCHIP) was laid out.

A Mechanical Test Chip was then designed and built to test etching techniques for the pressure sensitive silicon diaphragm, sand-blasting and laser micromachining of the glass wafer, electrostatic bonding of the glass and silicon wafers, and cable connection to the device.

CPT - CROSS SECTION



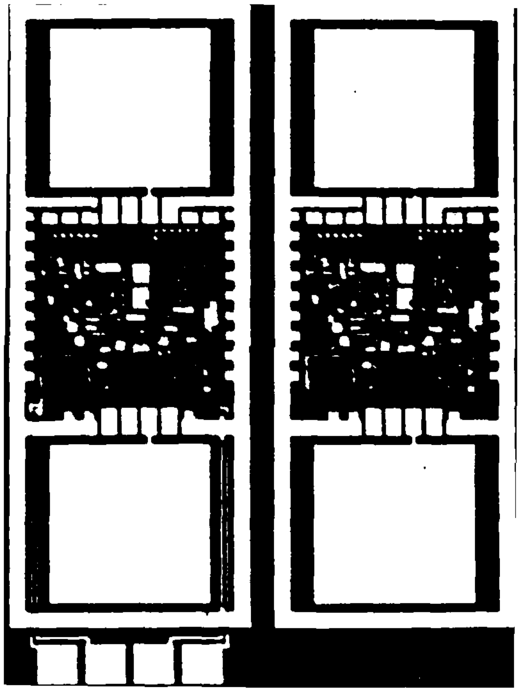
1 - ELECTRICAL CONTACT VIAS
2 - ELECTROSTATIC SHIELD
3 - CUSTOM INTEGRATED CIRCUIT
4 - REFERENCE CAPACITOR
5 - PRESSURE SENSING CAPACITOR

Fig. 7 Structure of Prototype Capacitive Pressure Transducer

Signal processing and output format required for the integrated capacitive pressure sensor were designed to switch the oscillator between reference and pressure sensitive capacitors. In addition to the sensed pressure, the circuit also produces temperature, pressure scale, temperature scale, and zero reference data, which provide all the information required for subsequent signal processing circuitry to calculate capacitance, and thus pressure, independent of temperature and drifts in offset or gain.

This circuit has been designed, simulated, breadboarded, laid out, fabricated, and tested. A photomicrograph of the circuit appears in Fig. 8. Early circuit tests showed the system to be fully functional but more sensitive to temperature and supply voltage than desired. Therefore, the circuit and its fabrication process were refined to reduce these dependencies and fabrication variations. Test results from 6 circuits demonstrate that temperature is measured to an accuracy of ±0.1 degC. As a result, pressure may be measured to an accuracy of ±0.7 mmHg in a full scale of 300 mmHg

In applications such as this, which require many sensors, it is impractical to connect wires to each sensor individually. A significant simplification in wiring results if the sensor output signals are multiplexed in time. A Multisensor Controller/Multiplexer (MC/M) integrated circuit that allows the output terminals from many silicon sensors to be fused together is currently under development. This integrated circuit which will fulfill a variety of specifications, has now been designed and simulated, first block by block, and then as an integrated system.



TECHNICAL REPORT

TASK 1: SURVEY OF KEY PROBLEMS AND TECHNOLOGY TRANSFER

In the first phase of this task — Survey of Key Problems — we have investigated relevant technology based on the experience of this contract, combined with fifteen years of work preceding it, and from the point of view of contacts in the industry. We feel a responsibility to summarize these observations to share research objectives.

The work of this contract is in the perspective of the broader scope of advanced manufacturing stations for aerospace manufacturing and automation. A primary issue is development of integrated technology for reconfigurable, programmable automation. The program for implementation of integrated technology includes necessary component technology and the integration technology to make possible the building of complex, integrated systems. Figure 1-1 shows this goal structure. Inspection and robotics in assembly are among the most advanced problems driving evolution of this aspect of programmable automation, which impacts all areas of manufacturing.

The goals of **component technology** are to make robots and machines with higher performance at a better performance to cost ratio, to implement new capabilities which expand their work, and to make them more programmable by requiring less fixturing, less preparation and less control of the work environment, by requiring less information to specify tasks and the environment, and by implementing modular component technology.

This perspective suggests priorities among modular component technologies:

1. Adaptation to External and Internal Uncertainties:
   In Task 3 we develop technology for compensation and for adaptation to flexibility of drive trains, structures, mountings, end effectors, and loads, and for adaptation to load. Methods also are developed for continuous system estimation and adaptation to changes in configuration.

2. Sensory Control:
   In Task 3 we include the use of end-point sensing in control. In related work funded under other contracts, technology is being developed for using contact sensors (force and touch sensors) to control compliance and exerting of forces in task coordinates for six degrees of freedom manipulators. An aspect of that research is force strategies in parts mating, both open loop and feedback strategies. That work has developed new methods for stability at contact, at the transition from position and velocity control to force control. Visual control of motion was demonstrated in collision avoidance in Task 2. Fine motion under sensory control, especially using small motion devices such as the wrist of Task 3b., demonstrates speed and accuracy.

3. Cooperation Among Manipulators:
   The quick wrist of Task 3b. explores coordinated coarse- and fine-motion strategies. Work is underway in two arm cooperation. Work under other support deals with three finger hands which require coordination of nine degrees of freedom. To summarize, in the area of manipulation, we have come to believe that among technologies that will

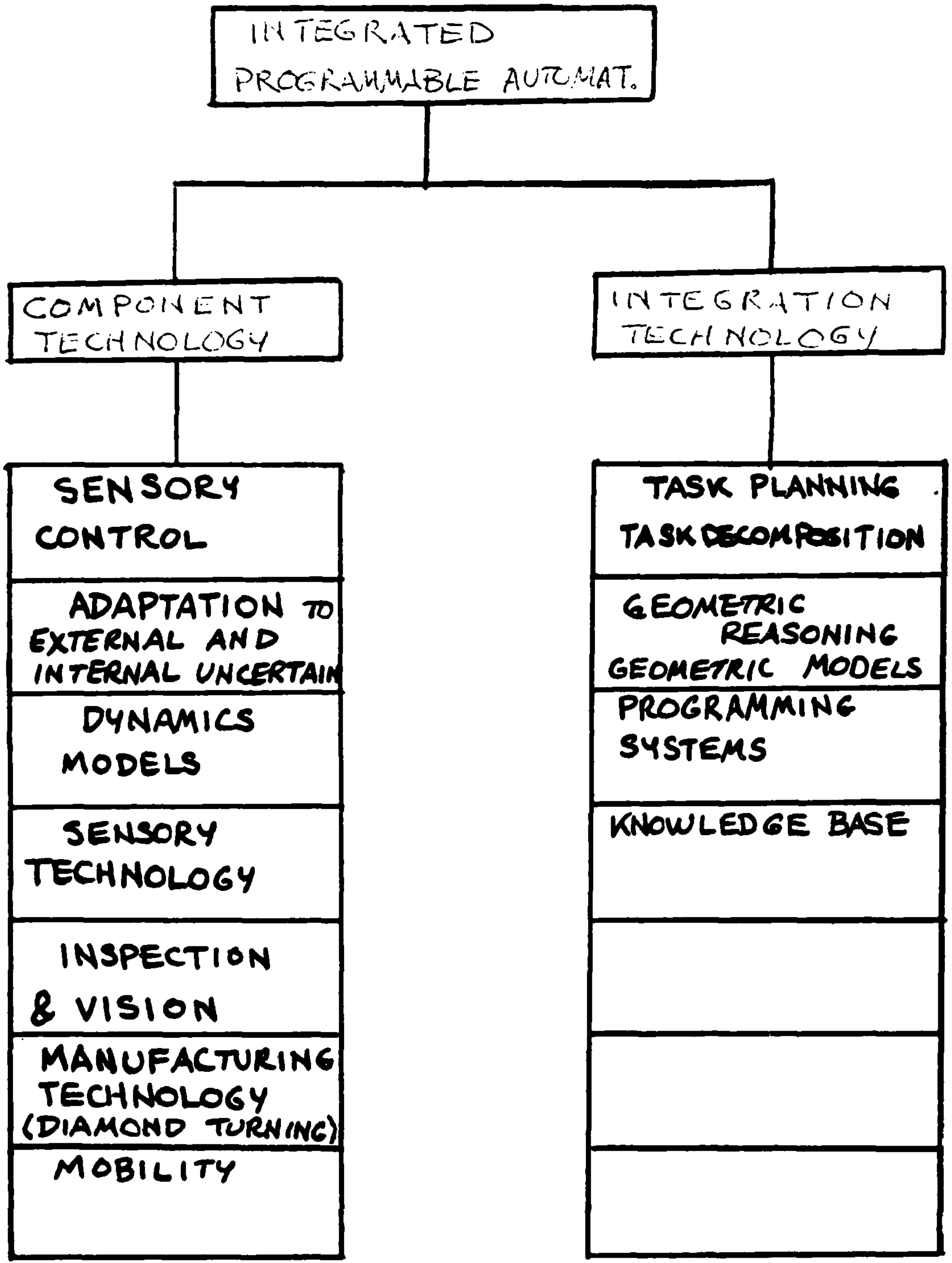

INTEGRATED PROGRAMMABLE AUTOMAT.
COMPONENT TECHNOLOGY
INTEGRATION TECHNOLOGY
SENSORY CONTROL
ADAPTATION TO EXTERNAL AND INTERNAL UNCERTAIN
DYNAMICS MODELS
SENSORY TECHNOLOGY
INSPECTION & VISION
MANUFACTURING TECHNOLOGY (DIAMOND TURNING)
MOBILITY
TASK PLANNING
TASK DECOMPOSITION
GEOMETRIC REASONING
GEOMETRIC MODELS
PROGRAMMING SYSTEMS
KNOWLEDGE BASE

Fig. 1-1

enable the next major steps for industrial robots, the following are of key importance: end-point feedback control that is robust and capable of achieving high bandwidths despite manipulator flexibility; simultaneous control of end-point force and position; high-level control of intimately *cooperating* manipulators; concerted use of quick local manipulators carried by much larger, possibly very flexible manipulators; real-time adaptive control that is fast enough, and sufficiently insensitive to noise to be of practical use. These are the areas in which we have been concentrating our experimental efforts in Task 3 of this AFOSR program, and in some other programs as well.

4. Sensory Technology:
   In Task 4 development of integrated circuit sensors has stressed providing system integration capability for arrays of sensors, especially signal processing and encoding and multiplexing of signals to minimize interfacing difficulties. Other work under DARPA funding dealt with implementing tactile arrays for tasks with the three finger hand.

5. Inspection and Vision:
   Inspection technology is likely to have as big an impact as robots over a very wide domain of manufacturing. In Task 2, we have built up mechanisms for incorporation in vision systems.

6. Manufacturing Technology:
   Under other support, Professor DeBra is building a diamond turning machine incorporating a novel laminar flow hydraulic motor.

7. Mobility:
   Especially in airframe production and machining centers, a robot can work over a large space, e.g., loading and unloading parts for a variety of machines, or working on large airframe parts. Methods for incorporating mobility into the manipulation and control are under development.

Integration Technology includes those disciplines necessary to build large, complex systems. Traditional computer science and artificial intelligence are key elements. The objectives are to provide the science base for integrating modular component technology into workstations, for integrating automation workstations into work cells, the next higher level of structure in manufacturing, and to link workstations to users. The components themselves are complex subsystems. Their integration involves multiple sensors and multiple information sources.

These integration technology elements are:

1. Programming Systems:
   Software technology is perhaps the technology with the most potential impact in automation. In Task 2 we devoted a small effort to a system for conventional explicit programming of motions and substantial effort to automating programming and concentrating the human interface on building models. A small effort advanced development of the AL system for programming robots.

   It is highly developed now and beginning to be integrated with the hand and with COSMOS, the force control system.

2. Task Planning:
   Task planning can be thought of as highly advanced methods for programming systems. In Task 2, a preliminary version of ATLAS is reaching the stage of planning tactile-controlled fine motion. Work on SUCCESSOR involves several aspects of automated task planning. In work under DARPA support, task decomposition with the three finger hand provides high level task primitives.

3. Geometric Reasoning:
   The basis for integrating modules is a thorough program to express everything in the system in terms of geometry, based on geometric representation, building generic, symbolic geometric models for object classes, representation of geometric constraints, models of sensors and operators, and modeling and prediction for surface properties, especially specularity. Inference rules and quasi-invariants provide new generic interpretation methods along with generic, symbolic prediction.

   Analysis of computational complexity and strategies for efficient execution provide the basis for integrating sequences of operations.

4. Knowledge Base
   Building up the knowledge base for intelligent systems is regarded as a major problem within the AI. Learning is one aspect of building knowledge bases. Our previous work has dealt with learning the relation between structure and function in defining object classes. Geometric representation and generalisation are central issues here.

The second phase of this task — Technology Transfer — has been carried out in two parts; first, technology transfer, and second, setting up mechanisms for sustained interaction with industry personnel concerning industrial applications and technology opportunities.

The Center for Automation and Manufacturing Science, CAMS, was set up under support of this contract, with Professors Cannon and Binford as co-directors. CAMS encompasses research in sensory robotics, computer vision and planning, manipulation for precision manufacturing, precision machining, sensory technology, and related technology topics.

Subsequently, SIMA, the Stanford Institute for Manufacturing and Automation was formed. Professor Cannon serves as Chairman of the SIMA Steering Committee. CAMS is one of four centers in SIMA. The others are: CDR, Center for Design Research, which focuses on the interaction between the designer and the system; CMFPS, the Center for Materials Formability and Processing Science, which investigates advanced processes for materials forming; and CTRIMS, the Center for Teaching and Research in Manufacturing Systems, which offers a master's degree program in manufacturing systems management and conducts research in manufacturing management. SIMA brings together the centers and industry sponsors. Two special mechanisms appear. The SIMA industrial advisory board meets twice a year. The board is made up of the principal responsible person for each corporation, typically a vice president. They are having a major effect on SIMA policy. They also are helping to establish contact between technical people in their organisations and centers within SIMA. This will provide valuable guidance to our university research and teaching in manufacturing, facilitate transfer of the technology we develop, and connect our manufacturing graduates with industrial career opportunities (by far, the most effective means for technology transfer). It will likely lead to specific contracts on topics of mutual interest. The current corporate members of SIMA are Apple, Boeing, FMC, Ford, General Electric, General Motors, Hewlett Packard, IBM, Lockheed, McDonald Douglas, Sperry and 3M.

Specific technology transfer, the other half of this task, has already made an impact. A major area has been transfer of force control technology. Initially, under the Intelligent Task Automation project (ITA), Adept Technology implemented our old force control technology, the Salisbury system. Subsequently, they have been developing techniques based on our current force control system in task space, by Khatib.

A major part of technology transfer is through research personnel. A total of about 10 students from CAMS have joined companies in this area of technology, including Adept, ADS, Boeing, Hughes, IBM, Martin Marietta, and SRI.

The Common Lisp development which supports our work in intelligent systems has been commercialized. Lucid Common Lisp is based on our implementation of TAIL, a Common Lisp for the SUN. The commercial version is attracting much attention on SUN 3 workstations and promises to be a low cost LISP workstation with good performance.

The ACRONYM system has been ported by a dozen or more laboratories and recently a handful of new requests have appeared. It is now being distributed both by the DARPA Image Understanding Testbed and by the Software Distribution Center at Stanford on a very low cost basis.

The AL system has affected the design of robot programming languages substantially. Copies have been made at Grenoble, France, at Karlsruhe, Germany, and at Tokyo. AL has had a very strong influence on standardization efforts for robot programming systems. Several companies have programming systems which look very much like AL: Hitachi Production Engineering Lab, Matsushita, ITMI, and Hewlett Packard. However, it has been difficult to find support for AL development in the USA.

We have now close contact with a number of companies, both long term relations and recently initiated ties through SIMA. Special relations are ongoing with Adept, ADS, Apple, Boeing, FMC, Ford, General Electric, General Motors, Hewlett Packard, Honeywell, IBM, Lockheed, Martin Marietta, McDonald Douglas, and 3M.

TECHNICAL REPORT ON TASK 2: INTELLIGENT SYSTEMS FOR MANUFACTURING; INSPECTION AND VISION

Overview

Throughout computer science, the programming problem is dominant. This is also true of programming robots and other programmable automation. For robot tasks which seem simple to program at the high level of specifying how parts fit together, in specifying detailed programs required by existing robots, programs expand through several substantial levels of program length and complexity. In the second year of this contract, the ATLAS system was undertaken to investigate general methods for programming at this high level of fitting parts together. Design and implementation of ATLAS included: 1) a simple geometric model representation of a small set of generalized cylinders; 2) a task specification module with representations of assembly graphs, constraints, and uncertainty; and 3) a skeleton matcher which decomposed tasks into a graph of component operations, identified constraints, and inferred goal positions. The current work in ATLAS is in two parts, 1) synthesizing fine motions when canned routines are inadequate, and 2) integrating a constraint solver. We plan to couple ATLAS as a program generator to the Cosmos force control system for robots to plan fine motion tasks for real robots.

In our work on intelligent systems, software portability is a major concern. Progress was made toward development of a practical Common Lisp for small machines. This work led to a commercial implementation which has high initial technical reviews. Separately, a real-time garbage collection algorithm was created.

In work under other support, the Intelligent Task Automation project (ITA), the intelligent system ACRONYM was extended and tested in recognition of 15 industrial parts. This experiment and analysis led to major criteria for the next system, SUCCESSOR. Under this AFOSR contract, extensions to the modeling system were designed; those extensions being implemented include: adding surface models and surface markings; implementing intersection for combining primitives by set operations; and automated generation of polyhedral approximation to generalized cylinder primitives. In the second year, a model-building program was implemented which a user could learn to operate in one hour and build models in fifteen minutes.

For the intelligent system, extensive work was done for prediction. In the ITA experiments, including more image features became important, especially specularity. Analysis and experimentation were performed to relate specularities to intrinsic surface properties, i.e., orientation and curvature. Substantial results show prediction of image specularity features from models and estimation of surface properties from images. Symbolic graphics of image features, i.e., symbolic edges and vertices, was analyzed and implemented for a large subset of generalised cylinder primitives and partially implemented for compound objects made of these primitives. These methods are relevant for recognizing individual objects in fixed positions.

Under this AFOSR support in the third year we have extended the earlier work in important ways. To extend prediction to a variety of sensors and multi-sensor tasks, analysis, design, and implementation have proceeded toward a system to incorporate models of sensors and perceptual algorithms, to make predictions for images and measurements, and to determine effective strategies for sensor-based tasks.

Theoretical work has led to a generic observability model for visual perception in the visible spectrum. This provides a firm basis for prediction and for interpretation of image data. Related work defines methods for predicting and estimating metric properties of objects, quasi-invariants which are insensitive to viewpoint or object class.

During the first year of the contract work was done on programming systems for sensory robots. Substantial development was performed on the AL system, especially in the user interface, i.e., structure editor and debugger, graphics, and voice synthesizer. New motion commands were added and commands for reading forces in arbitrary coordinate systems. A distributed version of AL was implemented. Since that time, under other support, AL has been extensively debugged, a VAX version has been implemented, and extensions have been added for three finger hand and other sensor-controlled capabilities.

Also during the first year, the COSMOS system was extensively developed. One portion of the work was development of a collision avoidance scheme based on fictitious repulsive potentials around obstacles. Substantial analytic improvements were made, culminating in a solution for the entire arm as a set of extended links. This method makes use of force control using fictitious and actual forces. Its development was coupled to that of force control in the COSMOS system. Thus, this work contributed also to substantial improvements in force control. Under subsequent ITA support and NSF support, force control was extended to make dramatic performance, e.g., the control of forces of about 15 gms, an improvement from 200 gms previous force control level.

Assembly Planning

ATLAS — Automatic Task Level Assembly Synthesizer — is a developing system for automated robot programming. Building application programs for robots is typical of the massive software problem which users face with modern computer-based technology. The software problem is accentuated by the difficulty of robotics and the special knowledge required. In this effort, we investigate automatic, model-based generation of programs. ATLAS is intended to synthesize sensor-based programs from high level specifications, e.g., instructions at the level of an assembly manual. Task level planning aims to simplify the task of the user in specifying only goals for how parts fit together, physical relationships among objects, not the motions of machines to achieve those goals. This is natural and compact to specify for users and CAD systems, in contrast to programming robot motions. A task-level specification is intended to be robot-independent.

ATLAS integrates the planning operations involved in building programs. A user specifies a task, e.g., attaching a lid to a box with four screws:

Place LID on BOX (against (BOTTOM-FACE LID) (TOP-FACE BOX))
Tighten BOLT1 (into (HOLE1 BOX) (HOLE1 LID))
Tighten BOLT2 (into (HOLE2 BOX) (HOLE2 LID))
Tighten BOLT3 (into (HOLE3 BOX) (HOLE2 LID))
Tighten BOLT4 (into (HOLE4 BOX) (HOLE2 LID))

The task specification is used to generate an Assembly Graph which represents physical states with objects as nodes and interactions as links. Links are time-tagged to indicate when they are valid. Objects interact by contact relations among features, i.e., surfaces, edges, and points of one object against those of another. Interactions may be first order, involving a single feature from each object, or they may be higher order. Relations can be categorized by their residual degrees of freedom.

The task planner expands the task into planning elements: parts feeding, layout, fixturing, grasping, coarse motion, and fine motion. The task planner generates a sequence of nominal operations like placing the box and placing the lid on the box. This sets up a constraint on the location of the lid relative to the box; the bottom surface is against the top surface of the box. Inserting bolt 1 into hole 1 of lid and hole 1 of box imposes another constraint, that hole 1 of lid is aligned with hole 1 of box, a constraint on the location of the lid relative to the box. The same is true for inserting the other three bolts.

If we expand the program to include robot motions, each step expands into about six motions. For each motion, a path must be calculated from positions of parts. The assembly will probably be unsuccessful because of uncertainties in positions and dimensions of parts. Typical robot programs expand a factor of five to ten in incorporating contingent actions, in analyzing sensor results, in specifying sensor states and strategies in sensor-controlled fine motion, or in using fixturing. In addition, data base references and calculations expand programs greatly. What was a simple program at the high level becomes more difficult and more technical at the level of robot motions. It becomes very difficult and very technical when uncertainties and contingencies must be included, as they almost always must be. These are the steps that ATLAS aims to automate.

As we saw, each step of the task imposed constraints on relative locations of parts. Different steps interact because they impose different constraints on the same variable, e.g., each bolt puts a constraint on the location of the lid relative to the box.

There are many uncertainties in task environments, i.e., uncertainties in part dimensions, in feeders and fixtures, in manipulators, and in sensors. Uncertainties can be analyzed to model the task environment, they can be designed to be small, they can be corrected by sensing and fine motion. They must be represented and dealt with in this system.

A high level task specification module was developed. It included PLACE with constraint AGAINST and INSERT (cylinder) with constraint relation INTO. A simple geometric modeler was developed which included circular cylinders and blocks, implemented as a small subclass of generalized cylinders. The assembly graph representation was developed. Constraints on uncertainties were developed. Operations in task specifications were decomposed into a set of skeleton operations which were: 1) Gross motion; 2) Grasp; 3) Ungrasp; 4) Bolt in hole; 5) Stacking: vertical align without sensing; 6) Synthesized Fine Motion. Bolt in hole, stacking, and synthesized fine motion are all called fine motion operations. A module called the skeleton matcher was implemented. It acted to: 1) identify which skeleton operations were needed; for example, INSERT involves gross motion, grasp, gross motion, bolt in hole, ungrasp; 2) infer the goal position for each operation; 3) identify applicability constraints for use of the operations and identify constraints which must be propagated; 4) build up the graph of skeleton operations; 5) identify time of operations, using one time unit per operation; 6) find all interactions among constraints on the current object.

The current work in ATLAS is in two parts, 1) fine motion synthesis, and 2) integrating a constraint solver.

Fine Motion Synthesis

The use of active compliances enables a robot to execute tasks under uncertainty. Compliant motions are difficult to program or specify. The geometry required is detailed, force strategies are complex and require specialized knowledge, and programming effort is large. These factors make automatic fine motion synthesis an important part of automatic robot programming.

[Losano-Peres 83] discussed factors of uncertainty and friction and used a pre-image method of backward chaining for synthesis of fine motion programs. They subsumed the work of [Mason 81] on hybrid position and force control for the motion of bodies under contact.

As a tool for automatic synthesis of hybrid control strategies, the freedom of bodies under contact has been related to their C-spaces using screw theory. One of the bodies represented as a point can contact the C-space of the other body under configurations that have different natural position constraints. A sensed change in position constraints carries information about the state of the object and the manipulator. This information is used to restrict the set of possible positions under sensor uncertainty. The change is also used as a guard-condition for shifting the control strategy for a different geometrical configuration. The nodes in the "reachability graph" from [Losano-Peres 83] correspond to a point body touching one feature of the C-space. Each of those nodes corresponds to many nodes, each corresponding to different freedoms of motion between the bodies. A program to implement the strategy is in progress.

Constraint Solver

Constraints are propagated between plan steps. Initially plan steps are modeled by instantiated plan skeletons and later refined into more detail. Constraint propagation continues as the plan steps are refined, propagating the inter-relating constraints back and forth between plan steps. Thus, each plan module acts as a generator of success/restriction pairs. As it is re-invoked it generates another solution or eventually fails.

The basic constraint solver has now been provided with "generator", "escape-generator", "try-next", and "add-possibility" macros to implement the generator-iterator environment.

To keep track of the associations of constraints and the corresponding modules that introduce each, each set of constraints is tagged to the frame that it belongs to. The generators are in fact local to the frames. Supporting macros to create frames, associate variables with each frame, pop and flush frames have been created. These are the "constraint posting" functions.

Assembly and Sub-assembly Inference

As a part of intelligent inference from task level specification, a tool is required to infer that a subset of components in the assembly is a sub-assembly. Such sub-assemblies can be assembled independent of the main assembly and transported to the main assembly position.

This problem has been posed as a sub-graph search problem. The essential tool to search the graph using the screw theory of "form-closure" has been studied. We do not know about the complexity of its computation.

Plans

We plan to implement fine motion synthesis as a program generator for the COSMOS system developed at Stanford. COSMOS has position and force control in task space which is required to carry out the fine motion operations dictated by the planning system.

Common Lisp Development

A major aim in system development was portability of software. Hardware is evolving rapidly; this provides an opportunity to move to systems with better performance or cost. Hardware evolution provides a risk of being tied to obsolete hardware. Hardware availability has changed substantially over the life of this project and will evolve dramatically over the life of a major software system. Software is the major investment in a computer system, typically estimated at 90% of the cost. From these considerations, it was decided to tie our work in intelligent systems to Common Lisp.

A subset of Common Lisp called SLisp was defined and implemented in Frans Lisp on VAX systems. SLisp has been used during the last two years in much of our work. It has meant that moving programs from the VAX to Common Lisp has been simplified. SLisp was used by some researchers at AI&DS, a local AI company, in their work.

A version of Common Lisp, TAIL, was developed for SUN workstations. It has been a major success. A company, Lucid, Inc., was formed to develop and market Common Lisp systems including personnel who worked on TAIL. Lucid Common Lisp is now available on SUN 3 workstations which are attracting attention as high performance Lisp workstations at a $25,000 cost. This may expand the contribution to the research of others that our own internal development could not make. Lucid Common Lisp has strong roots in TAIL. Technical advances in the TAIL compiler included: 1) enhanced portability including a machine independent code generation strategy; 2) improved code optimization and relatively efficient implementation; and 3) operating system portability in window systems.

A new algorithm for real-time garbage collection was developed which supports Lisp systems for robotics [Brooks 84]. This algorithm has significant performance advantages in implementing real-time garbage collection on ordinary computers like the 68000 with moderate cost in computation time and storage. Previous algorithms have large overhead on ordinary computers and required special hardware for efficient implementation.

Model-based Intelligent Systems For Inspection and Vision

The new model-based intelligent system is intended to generalize over the previous system, ACRONYM, to provide enhanced performance, greater speed, and to be more usable by other groups.

It will implement a much richer class of models:

a) more powerful generalized cylinder primitives;

b) boolean set operations (union, intersection, set difference);

c) joints between primitives;

d) merge surface and volume representations in painted surfaces.

It will incorporate more powerful symbolic predictions:

a) specularity and shading;

b) generic observability models;

c) models for sensors and vision operators;

d) expanded features and quasi-invariants;

e) hierarchical, structured prediction graphs.

It will integrate more capable symbolic observations from segmentation/aggregation operators and stereo vision.

It will introduce extended interpretation and matching of models with observation using:

a) coarse-fine models and matching;

b) strategies for matching;

c) hierarchical, structured matches.

Geometric Models

Geometric models are represented by compound volumes and surfaces which are formed from primitive parts, generalized cylinders, combined into structures by set operations, i.e., union, intersection, and set difference. An algorithm is under development by Chien to calculate the key operation, intersection, for a rich class of generalized cylinder primitives.

A key aspect is to make symbolic approximations for intersections in order to make generic prediction, to support geometric reasoning, and to calculate intersections to arbitrary accuracy of approximation. Curves and surfaces of intersection are characterized by analytical basis functions locally to whatever approximation is required and are characterized globally as splines with knots at discontinuities and other knots without discontinuities elsewhere as necessary.

Chien's algorithm uses methods of computational geometry to determine approximate intersection edges, surfaces, and volumes. Its volume representation is a volume tree, an oct-tree with spherical cells, its surface representation is a net of vertices, planar faces, and edges, and its edge representation is a rope of vertices and straight line segments. Surface and edge representations are first order splines which are initial approximations for iterative, analytic solutions for higher order splines.

For the intersection of volumes A and B, intersection curves are the intersections of their boundary surfaces ∂A and ∂B. In the volume tree, these are the joint boundary cells of A and B, which are grey cells for both A and B. They are starting places for iterating to points on boundary curves. The hierarchical volume structure enables efficient computation. Refined estimates for boundary points are intersections of edges of one surface with faces of the other. An iteration approximates points on boundary curves to desired accuracy. Appendix C describes the formulation and implementation of this algorithm.

Building Geometric Models

A system for building models was designed and implemented by [Takamura 84]. The system was designed so that a user could learn to use it in one hour and build simple models in ten minutes. The user interface included voice input selection from menus, combined with keyboard input for names. Models were built manually using stereo pictures; corresponding points were selected with a track ball. Other pointing devices could be used. Soon, automated stereo will be sufficiently robust to select most corresponding points. Previously we built a geometric editor which was text-based and one which used pointing devices to construct geometric forms. Also, a previous system built models automatically from depth data.

The system simplifies the task of the user by using some models of the class of generalized cylinders to minimize the input required. It uses hierarchical object structure to include parts which can be identical to others already specified; symmetry relations can be specified to decrease input. It also generalizes individual models to generic models (object classes). For simple objects, only four points are required to specify a model. For relatively complex objects, about ten points are necessary. The user defines cross section type and the few points to specify it, e.g., three points to specify a circle or rectangle. For a constant sweeping rule, another point specifies the object. Affixment relations of one part relative to another must be specified. The relations: Align, Coplanar, and Flush can be specified. Subpart structure can also be specified. The system provides a stereo display of the geometric model which enables the user to verify that the model is reasonable as the user constructs it. The object writes out a textual model of the part. It also determines generic models by generalizing constraints which determine the object class. As the user specifies members of the class, the system generalizes the constraints of the object class. This is subject to overgeneralization which the user must watch for.

Prediction

Specularity

Specularity was a very prominent feature of the industrial parts recognized in the Intelligent Task Automation project sponsored jointly by DARPA and Air Force. Specularity is an especially important problem in the multi-sensor integration domain; i.e., specularity is more important outside the visible spectrum than in it. For radar and acoustic imaging specularity is a principal property of images. Prediction and analysis should include shading effects. A generic model for reflectivities includes specular and matte components. Healey Appendix C has done analytic and experimental research to characterize specular reflection. The results show that useful symbolic predictions can be made for model-based vision. Further, specularities provide local estimates of the projected surface curvature. Appendix E includes details of this work.

Specularities provide intrinsic information about surface material and surface geometry. They are usually the brightest regions in an image and often have very high contrast. Thus they are easy to locate. The information they carry is very meaningful. If specularities are not properly understood, they are merely nuisances, like shadows, rather than valuable evidence. For a generic, model-based vision system, the issue is to include a symbolic characterization of specularities for prediction and for analysis, characterizations which will hold in varied situations.

We adopted the Torrance-Sparrow specular model. It assumes that the surface has a distribution of surface orientation, depending on roughness. A part of the micro-surface will satisfy the specular condition for angles in the vicinity of the specular peak. For glancing incidence there will be surface shadowing. The model includes a Gaussian distribution of intensity vs. angle about the specular peak, i.e., an exponential in the square of the angle; the rougher the surface, the wider the Gaussian. The specular peak is such that the ray to the source, the surface normal, and the ray to the observer are co-planar, and that the surface normal bisects the rays to the source and the observer.

The apparent width of a specularity is the projected width of the Gaussian. This allows estimating the rate of change of surface angle with projected angle, i.e., projected curvature. Expressed as a fraction of part dimension, the ratio is an intrinsic property of the surface. The two principal curvatures can be estimated. Careful measurements with an experimental setup on an optical bench confirm the calculated behavior.

Symbolic Graphics for Geometric Models

Graphics is aimed at display of a known, single view or sequence of known views of known, individual objects. Graphics is useful for the user in building geometric models, debugging calculations programs, and demonstrating programs. Graphics computations can be useful in a weak class of model-based vision systems for individual objects from individual views. To be useful, the graphics system should produce symbolic output instead of pixel output, e.g., edges, apparent edges, vertices, and surfaces. This is a limited form of symbolic graphics. A graphics system was begun at this level of symbolic graphics [Scott 84].

Our aim is a vision system which is viewpoint-insensitive for classes of similar objects. The system uses geometric models to make predictions which are generic with respect to object class and with respect to viewpoint. An investigation was made of extending the graphics system to generic prediction. Extension was only partially successful. This system makes use of Whitney's result that the only generic singularities of projection for a single, continuous surface are folds and cusps. Folds are apparent edges, i.e., limbs. This result is widely used in classification of image features. Multiple surfaces introduce obscurations at Ts; surface discontinuities introduce edges and vertices.

Limbs and cusps are found by an iterative, second-order Newton-Raphson procedure which finds one point and tangent on a limb. The program extrapolates for a starting approximation for a nearby point on the limb, using tangent and curvature and choosing the step length to make the expected number of iterations constant. This gives a fast solution with uniform accuracy of the limb. Obscurations are found by placing surfaces in an image plane quad tree. The quad trees are used to hold pointers to symbolic information about limbs. Hidden surfaces are obtained and Ts are calculated. Surfaces are ordered front to back.

Generic Symbolic Graphics

The symbolic graphics methods described above are effective for a part of the prediction for individual objects and particular viewpoints. In many problems, there are few objects and they appear in a few stable states. It is reasonable in those cases to store views of all stable states of all objects. Some attention has been paid in the community to storing all topologically distinct views of an object. [Koenderink 79] describe a subdivision of viewing space into cells, each of which has a distinct view of the object. Cells are linked in a graph structure. The cells are determined by ruled surfaces which divide up the viewing space. In our work on symbolic graphics we have investigated this approach. The symbolic structure which we generate has infinitesimal stability, i.e., it is stable under infinitesimal transformations of object or viewpoint.

However, there are severe barriers to practical use of this method for generic prediction: 1) it is dependent on individual objects; 2) there are many distinct views of even simple objects, i.e., a polyhedron of n faces has 2^n distinct views; each time a face is crossed there is a distinct view; 3) the graph is not stable under large deformations, i.e., is not quasi-invariant. These difficulties have proven difficult to overcome. No effective generic method of prediction has come out of our investigation of this approach.

Models of Sensors and Algorithms

Without an explicit model of sensors, an intelligent, model-based system, e.g., ACRONYM, has an implicit model of typical TV cameras for the visible spectrum. The system is limited to that source of sensor data. Its implicit model of the data is not very good, hence it cannot make detailed use of sensor data. Chelberg is building a subsystem and analytic models of sensor and perception algorithms to enable the model-based system to work with a variety of sensors, to integrate multi-sensor information, to plan efficient strategies for perception, and to make accurate predictions of observables obtained from perception. Appendix E presents a status report.

We intend to model sensors including: cameras for the visible and infrared; triangulation ranging utilizing stereo, motion, "structured light" ranging, and position-sensitive diode measurements; high resolution radar; phase measurement ranging; acoustic ranging; and force and tactile sensing. It is necessary to model illumination, also. We intend to model algorithms including: edge detection; edge linking; curve segmentation; curve grouping; curve intersection and vertex estimation; region segmentation; and texture grouping. Together with geometric models of objects and their surfaces, we must model reflectivities of surfaces for various sensors.

Sensors give drastically different response; the world looks very different in radar or infrared compared to the visible spectrum. A major aspect of the system is generic prediction, stable over a range of viewpoints and stable among similar objects in a class. Making analytic quantification of sensors and algorithms in perception is a step toward including multiple sensors in generic prediction and automatically determining the limits of generic prediction under different sensors.

We model statistical and systematic properties of sensors and algorithms. A camera is characterized by angular resolution of pixels, MTF, pixel sampling function, intensity resolution, noise characteristics, uniformity of response, dynamic range, spectral response and color, intensity quantization, bandwidth and sample time, lag, blooming, and other properties. Edge operators are characterized by diameter, angular accuracy vs. contrast, transverse position accuracy vs. contrast, false positives and false negatives vs. contrast, where contrast is the signal to noise ratio, computation cost, and others properties.

The system has objectives and sub-objectives to measure object properties with prescribed reliability, accuracy, and efficiency. It can only make efficient strategies if it has a way of characterizing cost, reliability, accuracy, and ways of relating measurements. These are important properties for models.

TASK 2. REFERENCES

[Brooks 83] Brooks, R., Lozano-Perez, T., "An Approach to Automatic Robot Programming", Internal Memo, MIT AI Lab, 1983.

[Brooks 84] Brooks, R., "A Real-time Garbage Collection Algorithm for Stock Hardware", Proc Lisp Conference, 1984.

[Koenderink 79] Koenderink, J., van Doorn, A., "The Internal Representation of Solid Shape with Respect to Vision", Biological Cybernetics 32, 1979.

[Lozano-Perez 83] Lozano-Perez, T., Mason, M., Taylor, R., "Automatic Synthesis of Fine-Motion Strategies for Robots", Int Journal of Robotics Robotics Research v3, 1984.

[Mason 81] Mason, M., "Compliance and Force Control of Computer-Controlled Manipulators", IEEE Trans Systems, Man and Cybernetics, 1981.

[Scott 84] Scott, R., "Graphics and Prediction from Models", Proc ARPA IU Workshop, 1984.

[Shostak 77] Shostak, R., "On the SUP-INF Method for Proving Presburger Formulas", JACM 24 p529, 1977.

[Takamura 84] Takamura, J., Binford, T., "Stereo Modeling System: A Geometric Modeling System for Modeling Object Instance and Class", Proc ARPA IU workshop, 1984.

TASK 2. PUBLICATIONS

Brooks, R., "A Real-time Garbage Collection Algorithm for Stock Hardware", Proc Lisp Conference, 1984.

Healey, G., Binford, T., "Predicting Specular Features",

Proc DARPA IU Workshop, 1985.

Scott, R., "Graphics and Prediction from Models", Proc ARPA IU Workshop, 1984.

Scott, R., "An Algorithm to Display Generalized Cylinders", Proc ARPA IU Workshop, 1983.

Takamura, J., Binford, T., "Stereo Modeling System: A Geometric Modeling System for Modeling Object Instance and Class", Proc ARPA IU workshop, 1984.

TECHNICAL REPORT ON TASK 3:
RAPID PRECISE CONTROL OF NONRIGID MANIPULATORS

Overview

The underlying objective here is to develop the sequence of technologies that will enable future generations of robots to move much more quickly, more deftly, than today's robots, achieving much higher levels of precision, while at the same time removing the need for robots to be the heavy, rigid, power hungry machines that today's robots are.

Toward this objective, we are pursuing a sequence of specific projects, each with specific capability goals to be demonstrated, that will provide key elements of the desired new robot technology base. The AFOSR Center of Excellence level of funding has made it possible for us to make major advances toward this goal, both in technological capabilities achieved and in the development of experimental facilities and expertise needed to carry on advanced research in this area.

Typical of new facilities is the experimental two-link arm with flexible tendons, which was completed in 1984 with AFOSR and DARPA funding, and on which we are now well underway in a comprehensive sequence of increasingly difficult experiments that will lead to demonstrations of more sophisticated robot control capabilities.

The flow and interaction of our projects is indicated in Fig. 3-1 and 3-2. The different facilities, indicated by the boxes in Fig. 3-1, are depicted more graphically in Fig. 3-2. *

To provide a helpful overview, Fig. 3-1 includes early manipulator projects that preceded the AFOSR Center of Excellence Funding as well as mutually synergistic projects that are funded from other sources. Each box in Fig. 3-1 (e.g., "Flexible Arm with Wrist") indicates a system that has been (or will be) developed. The blue double boxes and all of the Adaptive Control activity are AFOSR funded projects. Each new capability (e.g., "End Point Control") achieved with a given configuration is invariably based on extending what was learned first with an earlier, simpler configuration; and each of these new capabilities will be crucial to several that follow. Thus, for example, the Center of Excellence funding allowed us to bring the flexible manipulator with wrist configuration from first conception through design and fabrication and assembly to a working system on which early successful experiments with end-point control have already been accomplished. But this rapid progress was based firmly on the earlier step-by-step development of the single very-flexible manipulator, with its optical tip-position sensor and its pioneering end-point feedback control. (And that control, in turn, was able to draw on the earlier achievement of non-colocated control of the very-low-ζ disk system.)

Currently, the capabilities—of end-point control using two actuators in concert and of obtaining fine precision—that were developed first for the flexible arm with wrist, now form the technical base for the major series of developments under way with the new two-link manipulator facility: experiments with control in two dimensions, with first optical, then force control; experiments with two arms plus double wrist; experiments with target tracking and rendesvous leading to a *pair* of two-link manipulators cooperating to perform higher-level tasks. Note that both well-developed force control (of a flexible manipulator

* *Fig. 3-2 is not comprehensive. It does not include our AFOSR sponsored fundamental experiments in adaptive control, nor our NASA sponsored work on mobile robots using air cushion vehicles.*

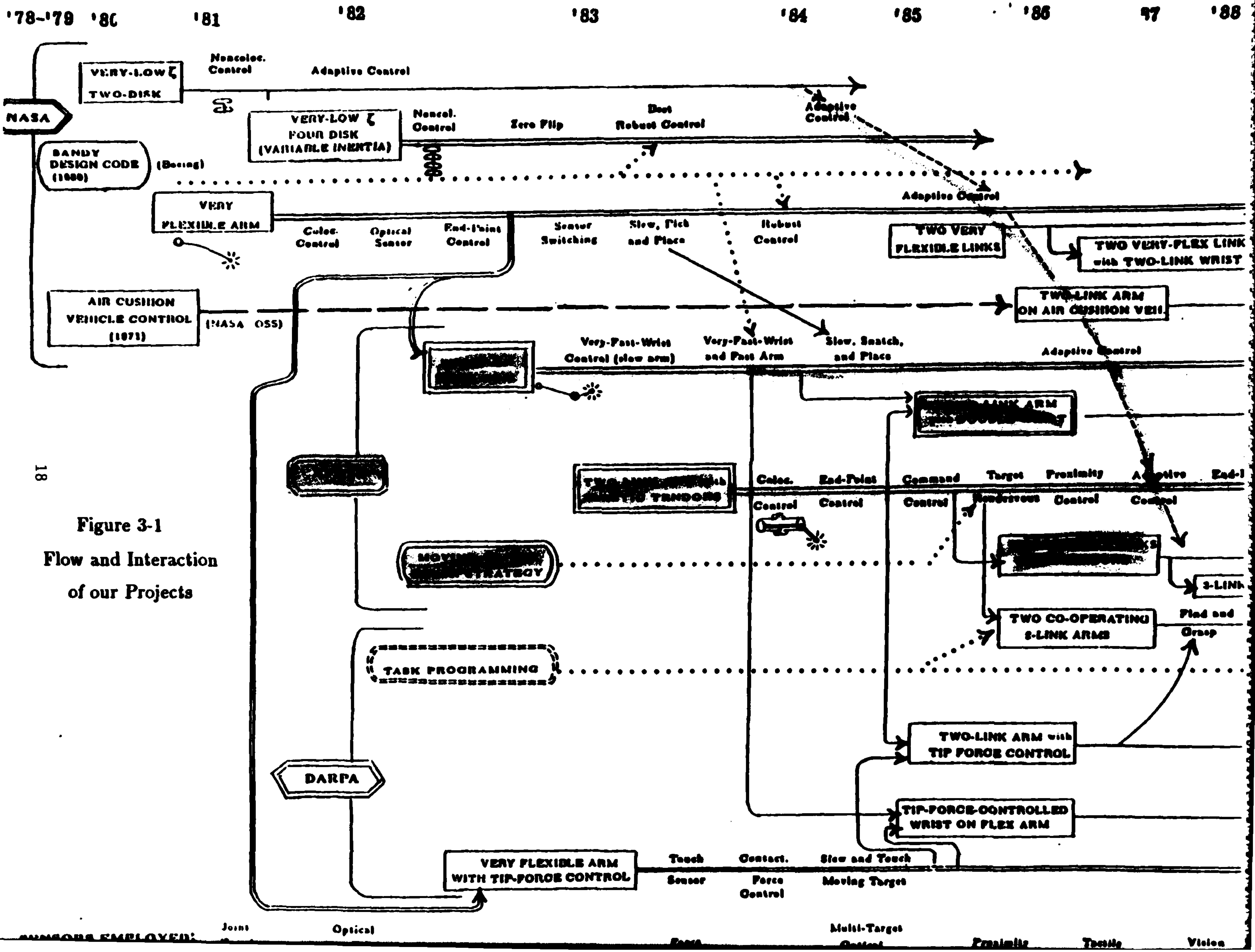


Figure 3-1
Flow and Interaction
of our Projects

been carried out to fulfill work statement items n and o. Results of this research are described in Section a. below.

(See particularly the Summary at the end of Section 3a.)

b. Another experimental arm consisting of a long, very flexible inner link and a short, quick outer link (wrist) has been built, (see Fig. 3-4) and definitive tests have been carried out to fulfill items p and q of the work statement. In particular, the system has been operated in a pick-and-place mode where it has demonstrated that the basic control technology for precise, rapid end-point optical control of tasks within a workstation is now in hand: The system demonstrated precise, flexible pick-and-place tasks independent of motions of the carrier arm, and ten times faster than the dynamic limits of even this lightweight carrier arm. In another demonstration, the carrier arm was slewed rapidly from one workstation to another, and the wrist began and completed pickup or place tasks at the new station well before the carrier could even settle down. These new results are reported on in detail in Section 3b. below.

c. In addition to the above projects, which fulfill the original statement-of-work items (with a body of very interesting and useful results), we have also proceeded to carry out research in two other important areas. First, we have completed two major studies on strategy for task command and control of two-link manipulators, which will be valuable in future phases of our experimental work. These are described briefly in Section 3c. below, and are reported on fully in separate reports as well as in two Ph.D theses.

d. Finally, we have proceeded to attack rather vigorously the generic problem of adaptive control. This has been a fashionable topic for mathematical studies for some decades; but the near non-existence of experimental results has made the promise an unfilfilled one to date. We have now begun, with AFOSR encouragement, an experimental program aimed at robot applications: for the payoff could be great indeed for future, high-performance assembly-type robots. We have achieved some promising preliminary experimental results in three different thrusts. This work is also reported below in Section 3d.

tip) and task programming will also be essential supporting capabilities at this point; and they too will be available from other, concurrent projects, as Fig. 3-1 shows.

As Fig. 3-1 indicates, the Two-Link Manipulator that we designed and built early in this AFOSR program will be (with its derivatives and augmentation) a major facility for a sequence of developements and demonstrations that will make more directly usable, by robot designers, the quite fundamental control advances we have been able to achieve.

A most important fact is that the gratifyingly rapid progress we have been able to achieve in each of the projects in Fig. 3-1 is due in great measure to synergism with the *other concurrent* projects in whose midst it exists. *The major AFOSR Center of Excellence support, together with the other funding indicated in Fig. 3-1, has made possible a critical mass of talented people and new equipment and activity, without which many of the achievements of this program would simply not have occured* at all, *let alone so quickly. The presence* concurrently of all these people is what made these achievements happen.

Again, the basic generic thing that we have been able to do (and be the first to do) is control very light, flexible manipulators in swift, purposeful motions: control position and force at their tips by measuring these quantities directly and feeding them back. As the sequence of projected project milestones in Fig. 3-1 unfolds, we aim to build a deliverable set of experimentally demonstrated fundamental robot-control-system design technologies that commercial designers can apply to the next much more capable generation of robotic systems.

Achieving the Objectives of the AFOSR-Supported Robot Manipulator Research

Guidance for the research supported by AFOSR Contract No. F49620-82-C-0092 is provided in the Contract Statement of Work, [AF 1] Sections B1 n, o, p, q, which are repeated here for convenience:

n. Extend the end-point control laws developed for single link flexible arm operation to two link operation in the horizontal plane and investigate extension to the vertical plane dimensions where gravity affects the system.

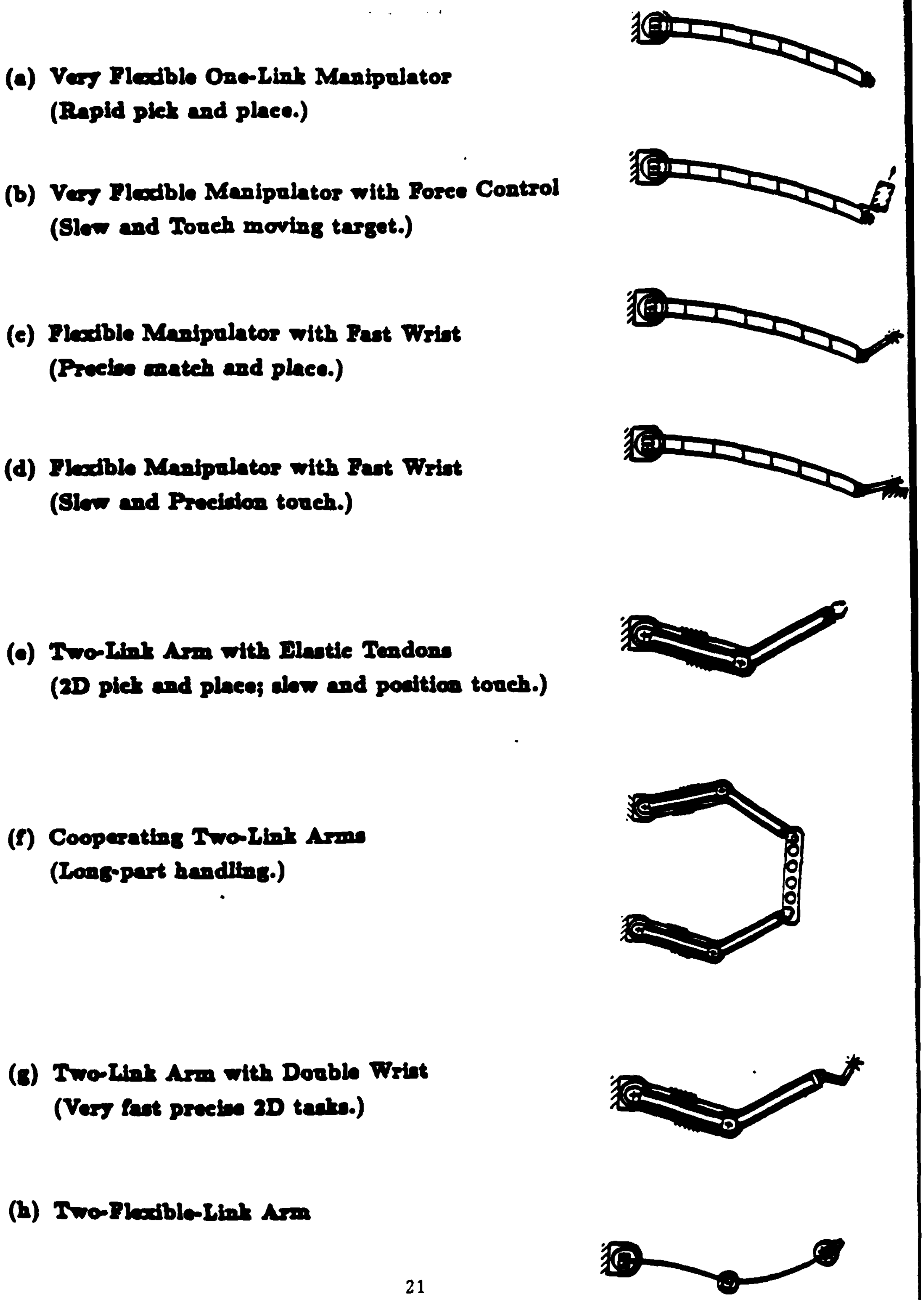
o. Investigate the capabilities and limitations of lumped-compliance robot arms. Specifically, determine the slew speed, end-point settling time and positioning accuracy, proximity motion, and contact-force performance for a range of: flexibility, payload mass and inertia, actuator power, controller bandwidth; and study inherent actuator and sensor non-linearities.

p. Extend the work in tasks o and p to robot arms with distributed flexibility.

q. Investigate the performance to be expected of the complete flexible robot arm with end-effector for various real-life manufacturing, assembly, and inspection tasks.

In Fig. 3-1 the specific projects that address these work-statement items are shown in blue, as are the tasks that are now completed and reported on here, namely:

a. A two-link lumped-compliance experimental robot arm has been built and tested; and both (conventional) joint-controlled tasks and (new) endpoint-controlled tasks have



(a) Very Flexible One-Link Manipulator
(Rapid pick and place.)

(b) Very Flexible Manipulator with Force Control
(Slew and Touch moving target.)

(c) Flexible Manipulator with Fast Wrist
(Precise snatch and place.)

(d) Flexible Manipulator with Fast Wrist
(Slew and Precision touch.)

(e) Two-Link Arm with Elastic Tendons
(2D pick and place; slew and position touch.)

(f) Cooperating Two-Link Arms
(Long-part handling.)

(g) Two-Link Arm with Double Wrist
(Very fast precise 2D tasks.)

(h) Two-Flexible-Link Arm

Fig. 3-2 Sequence of Expermental systems,
(A through E built and tested; F through H under development)

TASK 3a.
TWO LINK MANIPULATOR WITH FLEXIBLE TENDONS

Introduction

The successful achievement of accurate, high-bandwidth end-point control of the Stanford one-link flexible manipulator provided the groundwork for extending the technology to the general multi-link flexible manipulator case. Toward this goal, we constructed a two-link manipulator in the horizontal plane (so-called SCARA configuration) with a vertical drive at the working end (see Fig. 3a.-1).

The end-point control of a two-link arm represents a significant increase in complexity over the one-link flexible arm research pioneered in our lab. The complexities are introduced in three main areas. First, the two-link arm dynamics and kinematics are highly non-linear. Second, the two-link arm system is inherently multi-input, multi-output (MIMO) since the dynamics are highly coupled and in general cannot be decoupled into smaller single-input, multi-output (SIMO) systems. Third, the design and construction of an end-point sensor system that gives a high signal-to-noise ratio and high repeatability over a planar sensing area was required.

There is one area where the two-link arm has been intentionally made simpler than the one-link arm: the system flexibility has been discretely lumped in the form of springs in the drive train instead of continuously distributed in the structure. Thus, the problem has been initially simplified in one respect so that we can more easily handle the new complexities of two links. Later versions of the two-link arm will include continuously flexible structural elements. In fact, the rigid wrist attached to the one-link flexible manipulator represents initial studies of the case of a flexible upper link and a rigid forearm link.

Non-Linear Dynamics and Kinematics

The two-link arm dynamics are non-linear in three different ways. First, the inertia of the complete arm about the shoulder joint is a cosine function of the elbow joint angle. Thus, pulling the arm in will decrease the overall arm inertia, and the system's vibration frequencies will change. Second, there are centripetal accelerations present which are functions of the square of the link angular velocities. For the high speeds at which our system operates, these non-linear acceleration terms are roughly the same order of magnitude as the linear acceleration terms and cannot be ignored. The third major non-linearity is the large variation in inertia due to large changes in payload mass. Other practical non-linearities that have to be compensated or eliminated through careful design include motor and joint stiction, friction, hysteresis, backlash, and gravity effects.

These non-linear dynamics and kinematics have always been present on robotic systems built in the past, but could be accommodated with little adverse impact on performance by operating the system at very low bandwidths.

The unique features of our manipulator, which require meticulous consideration of these non-linearities, are the high operating speeds, the presence of variable flexible modes, the large payload-to-arm-inertia ratio about the shoulder joint (about 1:1 for our arm compared to 1:10 for most arms), and especially the use of end-point sensing.

Multi-Input-Multi-Output (MIMO) Character

The MIMO aspect of the two-link arm (along with the quick-wrist system, Task 3b.) represents a major step for our robotics lab. Most of our previous experiments with the one-link manipulator position and touch control have been SIMO systems which can be treated with classical control analysis and design techniques (such as Root Locus, Bode and Nyquist). The two-link arm is inherently a MIMO system and is difficult to treat with these techniques. State-space methodologies such as Linear Quadratic Loss and Sandy Gradient Search techniques have been successfully used in prior research on MIMO systems.

New End-Point Sensor Technology

To achieve useful end-point control for the two-link and future manipulators, the end-point sensor must be able to sense position in a plane (in three dimensions for the general case), have a bandwidth much faster than the desired closed loop bandwidth, have a high signal-to-noise ratio, have the accuracy and repeatability required for the desired task, and the capability to track moving targets as well. As a surrogate for anticipated vision systems, which have all these characteristics, the one-dimensional end-point sensor used for the single-link-manipulator experiments was entirely redesigned for this task.

This final report contains a complete description of the experimental apparatus modeling and identification of the two-link manipulator, a computer simulation of the manipulator, control system design, analysis, and experimental results for two different controllers and a summary.

Experimental Apparatus

The experimental apparatus (Fig. 3a.-1) consists of four major components: (1) the manipulator structure and drive system; (2) the sensors and associated electronics; (3) the control computer and associated interface electronics; and (4) the software for control design, analysis, and real-time control implementation.

Manipulator Structure and Drive System

Fig. 3a.-1a is a photograph of the two-link rigid arm with flexible tendon drives. The basic structure consists of a support base and two rigid links lying in a plane perpendicular to gravity and connected by revolute joints. The drive motors are attached to the base and transmit torque to the links via a lightly geared tendon drive train. The tendons have springs connected in-line as the source of exaggerated flexibility. At the end of the arm is a decoupled vertical axis and simple gripper that gives the arm complete three dimensional positioning capability.

The support base was constructed with about 1000 kg of reinforced concrete and a 3/4 in. thick aluminum mounting plate for the motors, drive train, and first joint. The concept behind such a massive base was to provide an essentially immovable and rigid support structure which does not interact dynamically with the flexible manipulator and adversely affect the results. The aluminum mounting plate also has provisions for balancing the manipulator or tilting it up to 15 degrees to allow up to a 25% component of gravity to affect the system.

As shown in the schematic of the manipulator in Fig. 3a.-1b, the motor in the right cavity of the base drives the shoulder joint and upper arm link through four springs, while the motor in the left cavity of the base drives the elbow joint and forearm link via an idler pulley centered about the shoulder joint and then through two springs. Each spring has a nominal spring constant of 40 lb/in (7000 N/m). The arm was designed to have minimum weight and inertia about the joints and yet be very stiff structurally by using aluminum tube and shell construction. The maximum vertical sag when fully loaded was designed to be less than 1 mm, and any structural vibration frequencies are well beyond those created with the springs in the tendon drives. Each link is about one-half meter long. The shoulder joint has a range of ±90 degrees and the elbow joint has a range of ±120 degrees. This gives the manipulator a 180 degree arc-shaped operational envelope with a minimum reach of 0.5 meter and a maximum reach of 1.0 meter from the shoulder joint. The vertical axis gives about 15 cm of vertical operational space. The arm was designed to have a payload-to-arm-inertia ratio about the shoulder of about one. The large payload capability, combined with the ability to change the springs in the tendon drives and the inertia at the motor shaft, gives the system a very wide range of plant dynamics for study.

The drive motors are so-called "printed circuit" electric DC motors that have no ferrous material in the armature. Thus the motors have no cogging torque and the ripple torque is minimized. The dominant non-linearity in the motors is brush friction and stiction, and these have been reduced to acceptable levels by feedforward and dither. The motors can provide up to 1580 oz-in. (11.2 N-m) of torque, while the motor torque non-linearities can be held to less than 5 oz-in. (0.04 N-m). The motors and gearing were sized such that the unloaded arm can theoretically move between any two points in its operational envelope within one second. For the one meter arm this gives accelerations on the order of one gravity (10 m/sec^2) at the tip. The shoulder drive train has a 6.06 to 1 gear reduction, and the elbow drive train has a 2.91 to 1 gear reduction. The gearing was sized to provide almost direct drive capability to minimize gearing problems such as backlash and large friction, rather than to provide the optimum impedance match for power transmission.

The vertical-axis drive system was designed to move up to a 1 kg payload through the full 15 cm range of motion in about two seconds. The vertical axis is driven with its own decoupled actuation system consisting of a small DC torque motor with optical encoder position sensor which drives a lead screw which in turn moves the gripper vertically. A pneumatic gripper at the tip of the vertical axis provides grip actuation on the order of 50 msec and is designed to pick-up one type of payload, a disk (of mass $\leq$ 1 kg) with a cylindrical grip hole machined in the center.

The two-link arm hardware has undergone several upgrades and "tuning" to improve performance. The elbow joint had to be stiffened, the other joints realigned and all the

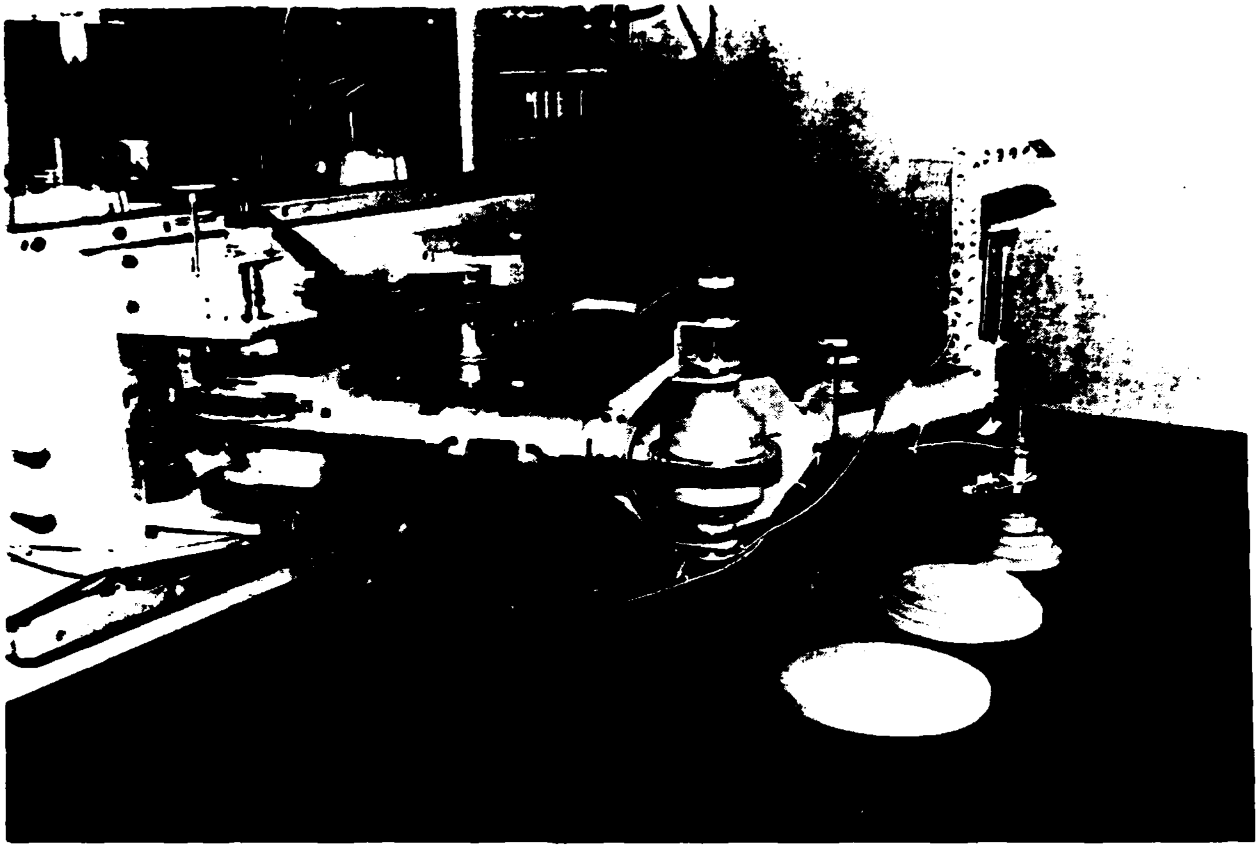


Fig. 3a.-1a Photograph of the Stanford Two-Link Manipulator with Flexible Tendons

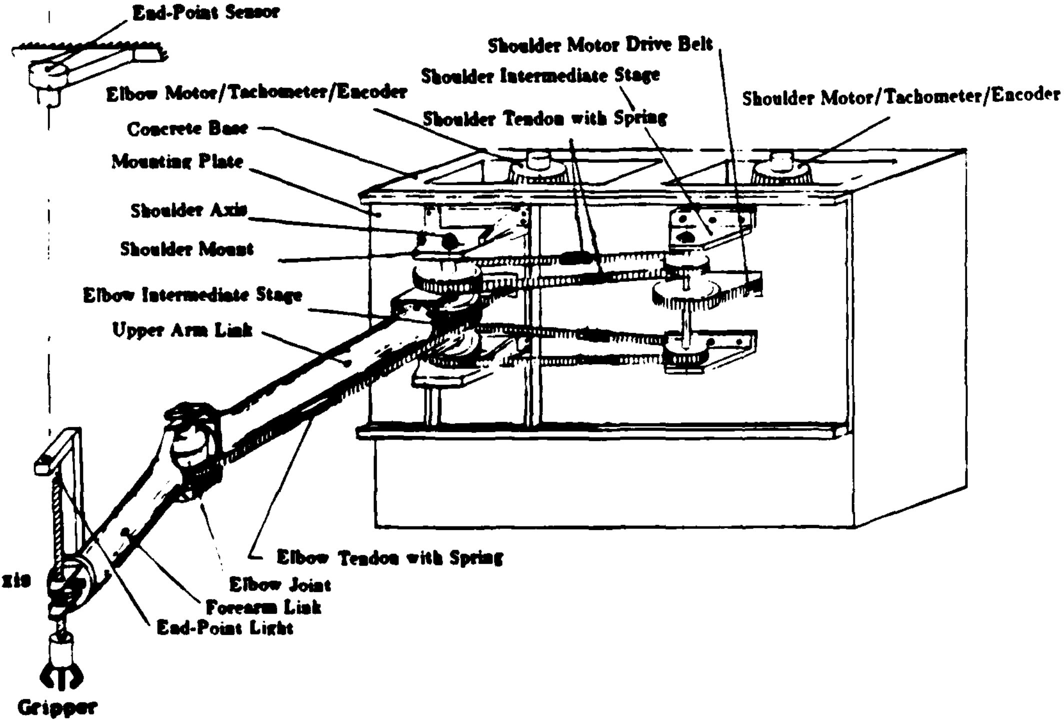


Fig. 3a.-1b Schematic of the Stanford Two-Link Manipulator with Flexible Tendons

bearings replaced with higher grade bearings or self-aligning bearings to reduce stiction in the system. The final major change will be to replace the plastic/steel tendons with all steel tendons to further improve drive train linearity.

Sensors and Associated Electronics

The primary sensor for the two-link arm is the optical end-point sensor (see Fig. 3a.-1). The basic construction of the sensor is a single large silicon wafer photodetector with associated optics. The center of brightness observed by the photodetector is output as a ratio of voltages across two perpendicular axes on the wafer. The new version has a 40 dB improvement in signal-to-noise ratio over the original one-link manipulator end-point sensor. The sensor is tuned to observe rapidly flashing infrared LED's mounted on the tip of the arm without the need of a light isolation hood. The bandwidth of the sensor for tracking one point is about 500 Hz and provides a resolution of 1 part in 1000 over the range of the sensor (i.e., 1 mm for a 1 meter range).

The system is also well instrumented with secondary sensors. Each motor has a tachometer and an 1800 line optical encoder. The shoulder joint has a 4096 line optical encoder, and the elbow joint has a 2540 line encoder. End-point position can be inferred to an accuracy of about 1 mm using these encoders (assuming rigid links and current calibration data), and rate information can be derived from the optical encoders using a special electronics board developed in our lab.

An electronics rack has been built to house the sensor electronics, motor power supplies and amplifiers, and computer input/ouput connections. The sensor electronics housed includes the end-point sensor electronics, five optical encoder position and rate boards (two for the motors, two for the joints and one for the vertical axis), and motor tachometer outputs. The motor controller includes three separate analog inputs to each motor, current limiters, and motor disabling electronics for safe and quick shut down of the system in case of computer failure, intermittent power failure, or imminent harm to equipment or people. The computer I/O connections include twelve 12-bit digital to analog converters, 32 differential 12-bit analog to digital converters, and a digital I/O cable for the encoder electronics.

Control Computer

The control computer is a DEC PDP 11/24 mini-computer with a hardwired floating point board. In addition to the I/O electronics listed above, the computer system has a real-time clock, IEEE 488 bus, two RL02 removable hard disks and asynchronous communication ports for connections to printers and direct lines to a VAX 11/780 mainframe computer cluster. The VAX computers are used for control design, analysis and simulation, while the PDP 11/24 is used to implement the controllers in real-time. The PDP 11/24 is running under RT11 version 5.2 operating system, and the control programs are written in FORTRAN and PDP assembly language. The real-time control programs are run at 50 Hz to 200 Hz sample rates depending upon the complexity of the controller.

Software

Most of the control analysis, design, and simulation software used on this project was either commercially available or previously written here at Stanford. These include the Stanford developed OPTSYS, DISC, and SANDY, which are state-space linear quadratic Gaussian design tools for MIMO systems. OPTSYS designs estimators and regulators for continuous analog control. DISC designs estimators and regulators using discrete, sampled data controllers such as digital computers. SANDY designs low order, robust controllers for continuous analog control. Analysis and simulation tools including the commercially available MatrixX and Control-C have proven very useful in our research. A graphical simulation tool has also been developed here during the course of our research.

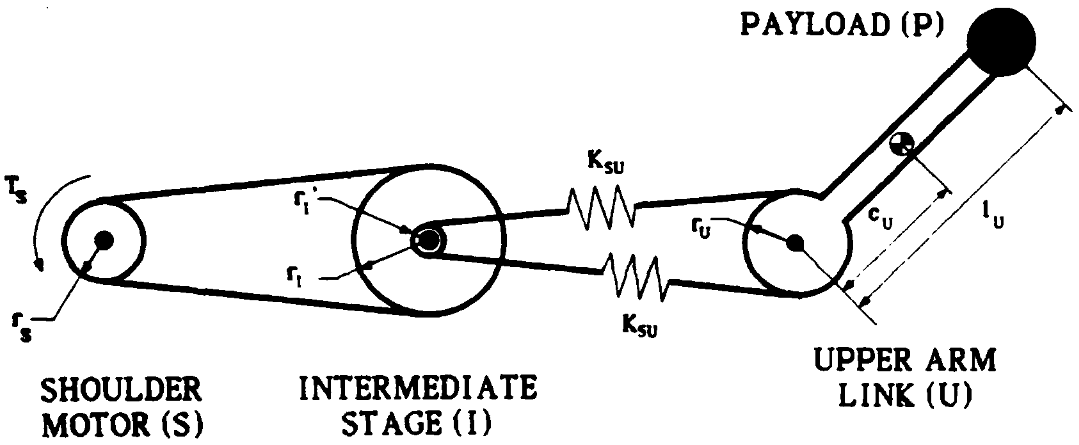
Most of the real-time control and peripheral software was developed in our lab using FORTRAN and PDP assembly language. The software developed for the one-link flexible arm was modified and enhanced for use on the two-link arm. The software includes assembly language drivers for the clock, analog-to-digital converters, digital-to-analog converters, end-point sensor, and optical encoders. Assembly routines were also written to speed up matrix and other floating point calculations used in the control loop. FORTRAN programs include, of course, the control programs, testing and calibration programs for all of the hardware, and a library of useful subroutines that perform the kinematic arm solution, inverse dynamics solution, data storage and retrieval, data transmission to and from the VAX, system identification software, and I/O handlers for the control programs.

Modelling and Identification

A good model of the two-link arm and the identification of the parameters of the model are required for control system design and implementation. As a first step in the modelling and identification of the two-link arm, the system with the elbow locked was modelled and identified. Since the arm with the elbow locked is a single-input linear system, this also helped to isolate and characterize actuator and sensor non-linearities which were masked by the non-linearities and multi-input-multi-output (MIMO) character of the full system. Substantial non-linear motor friction and tendon plasticity were identified and compensated. Next, a complete non-linear model of the plant was developed and then its parameters were identified.

Locked-Elbow Model and Identification

A model of the two-link arm with the elbow locked is shown in Fig. 3a.-2. Since the intermediate gearing stage is kinematically coupled with the motor, only four states are required to describe the system:



Each Motor, Joint and Spring has damping: d_S, d_U, d_{SU}

Each Body has mass and planar inertia: m_S , m_I , m_U , m_P , I_3^S , I_3^I , I_3^U , I_3^P

Fig. 3a.-2 One-Link Arm Model

$$\left\{\begin{matrix} q_S \\ u_S \\ q_U \\ u_U \end{matrix}\right\} = \left\{\begin{matrix} \textit{Shoulder motor position} \\ \textit{Shoulder motor rate} \\ \textit{Upper arm position} \\ \textit{Upper arm rate} \end{matrix}\right\}$$

The equations of motion for this system are given by:

$$\begin{bmatrix} \dot{q}_S \\ \dot{u}_S \\ \dot{q}_U \\ \dot{u}_U \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{\kappa_S n_S^2}{I_S} & -\frac{d_S+\delta_S n_S^2}{I_S} & \frac{\kappa_S n_S}{I_S} & \frac{\delta_S n_S}{I_S} \\ 0 & 0 & 0 & 1 \\ \frac{\kappa_S n_S}{I_U} & \frac{\delta_S n_S}{I_U} & -\frac{\kappa_S}{I_U} & -\frac{d_U+\delta_S}{I_U} \end{bmatrix} \begin{bmatrix} q_S \\ u_S \\ q_U \\ u_U \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_S} \\ 0 \\ 0 \end{bmatrix} T_S \qquad (1)$$

where the intermediate symbols are:

$$\begin{aligned} &\kappa_S = 4K_{SU}r_U^2, \text{ torsional spring constant.} \\ &n_S = n_1 n_2, \text{ total gear ratio.} \\ &n_1 = \frac{r_S}{r_I}, \text{ motor to intermediate stage gear ratio.} \\ &n_2 = \frac{r_U}{r_I^\prime}, \text{ intermediate stage to joint gear ratio.} \\ &I_U = I_3^U + I_3^P + m_U c_U^2 + m_P \ell_U^2, \text{ arm inertia about joint.} \\ &\delta_S = 2d_{SU}r_U^2, \text{ torsional damping.} \end{aligned} \qquad (2)$$

and the physical parameters are:

K_{SU} : Linear spring constant between shoulder motor and upper arm.

$r_S, r_I, r_I^\prime, r_U,$: Radii of shoulder motor pulley, intermediate stage input, intermediate stage output, and upper arm pulley, respectively.

$I_3^S, I_3^I, I_3^U, I_3^P,$: Planar inertias of the shoulder motor, intermediate stage, upper arm, and payload, respectively, as calculated about each body's center of mass.

d_S, d_U, d_{SU} : Viscous friction about the shoulder motor, upper arm joint and between the motor and joint, respectively.

$m_U, m_P,$: Masses of upper arm and payload, respectively.

c_U : Distance from upper arm joint to upper arm center of mass.

ℓ_U : Distance from upper arm joint to payload center of mass.

The system parameters were identified by running a sine sweep test with the input being the voltage command to the shoulder motor which generates torque T_S and the output being the voltage from the tachometer which measures rate U_S. The resulting Bode magnitude and phase plot from the test is shown in Fig. 3a.-3 as dashed lines. From the state-space description of the system (Eq. 1), the transfer function from tachometer output to motor input is given by:

$$\frac{u_S}{T_S/I_S} = \frac{s^2 + 2\zeta_z\omega_z s + \omega_z^2}{(s+d)\,(s^2 + 2\zeta_p\omega_p s + \omega_p^2)} \tag{3}$$

where the frequency domain parameters are:

$$\begin{aligned}
\omega_z &= \sqrt{\kappa_S/I_U} \\
\omega_p &= \sqrt{\kappa_S/I_R} \\
d &= \frac{d_U n_S^2 + d_S}{I_U n_S^2 + I_S} \\
\zeta_z &= \frac{1}{2\omega_z}\left(\frac{\delta_S + d_U}{I_U}\right) \\
\zeta_p &= \frac{1}{2\omega_p}\left(\frac{\delta_S + d_R}{I_R}\right)
\end{aligned} \tag{4}$$

and the "reduced" inertia and damping are:

$$I_R \frac{= I_S I_U}{I_U n_S^2 + I_S}$$

$$d_R \frac{= d_S d_U}{d_U n_S^2 + d_S}$$

A Bode magnitude and phase plot of the transfer function model is shown in Fig. 3a.-3 as solid lines. The numerical values of the physical parameters identified are listed in Table 3a.-1. Note that the low frequency portion of the theoretical Bode plot does not exactly match the experimental Bode plot. This discrepancy is due to non-linear Coulomb friction present in the motors.

Frequency Domain Parameters	Intermediate Parameters	Physical Parameters
ω_z = 1.25 Hz	κ_S = 117.2 N-m/r	K_{SU} = 7000 N/m
ζ_Z = 0.03	n_S = 0.165	r_S = 0.026 m
ω_P = 7.80 Hz	n_1 = 0.406	r_I = 0.065 m
ζ_P = 0.10	n_2 = 0.406	r'_I = 0.026 m
d = 0.10 Hz	I_S = 0.0014 kg-m^2	r_U = 0.065 m
	I_U = 1.90 kg-m^2	d_S = 0.014 N-m/r/s
	δ_S = 0.18 N-m/r/s	d_U = 0.71 N-m/r/s
		d_{SU} = 21.6 N/m/s
		l_U = 0.993 m

Note: The planar inertias, masses, and link center of mass location were not explicitly identified. The damping parameters are somewhat artificial since we are trying to fit linear viscous damping to non-linear Coulomb friction and stiction.

Table 3a.-1

One-Link Arm Identified Parameters

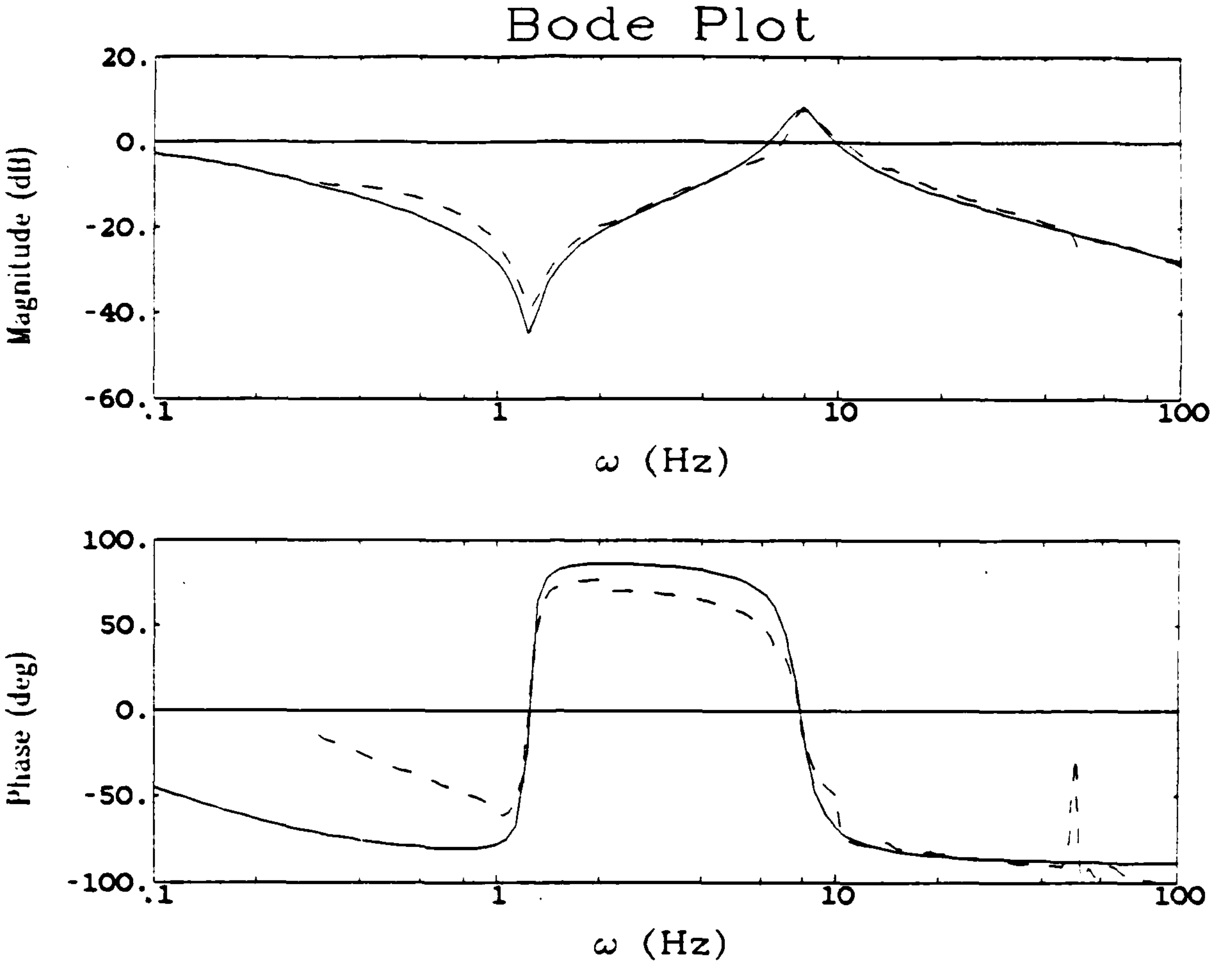


Fig. 3a.-3 **Sine Sweep Identification of Manipulator with the Elbow Locked.**
Analytic Model ▬▬▬
Experimental Data ---------------
Note the non-linear friction effects at low frequencies and the spike due to dither at 50 Hz

Motor Friction and Drive Train Non-Linearities

In the course of identifying the locked-elbow system, the system damping was noted to be much higher than predicted by motor specifications and viscous bearing damping. A simple pendulum test of the motor (the motor acting as the pivot for a hanging rigid pendulum) showed that the resulting sine wave decayed linearly with time instead of exponentially as predicted by assuming viscous damping. The linear decay in amplitude indicates that Coulomb friction predominates. Torque break-away experiments showed that a small amount of stiction torque was also present. The pendulum test results and graph of the resulting friction model are shown in Fig. 3a.-4.

The Coulomb friction was measured to be approximately 11 oz-in. and the stiction about 12 oz-in. The Coulomb friction has been successfully compensated by feeding forward the measured Coulomb friction level as a function of direction of rotation of the motor shaft. This compensation effectively reduces the Coulomb friction levels to 2 - 4 oz-in.

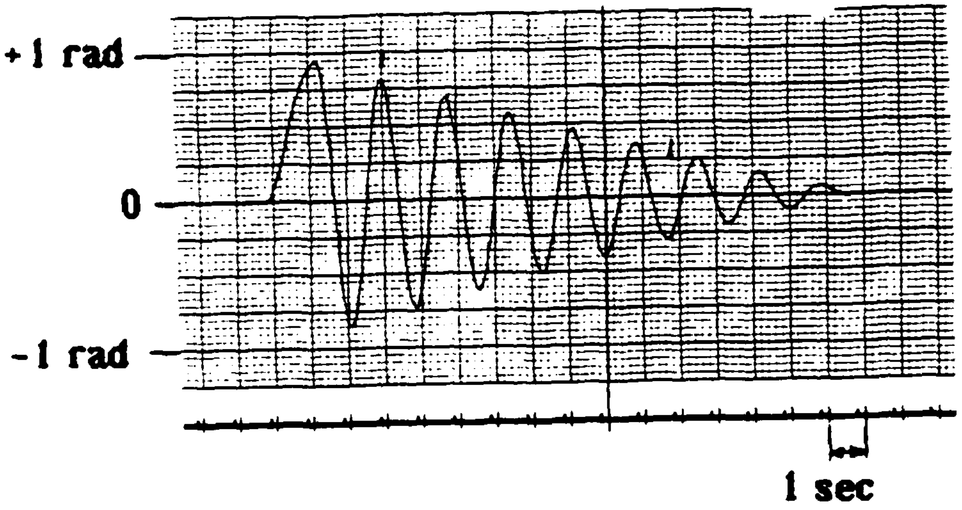
Significant bias torques (on the order of the Coulomb friction torque, i.e., 10 - 20 oz-in. as measured at the motors) are present in the drive train due to plastic deformations of the plastic cable drive as they follow the small gear radii and by other wires and pneumatic tubes that cross the robot joints. These effects are currently compensated by integral control action. However, a new all-steel cable drive has just been built and will be installed soon. This change along with redirection of the wiring should reduce these bias torques significantly.

Two-Link Arm Model and Identification

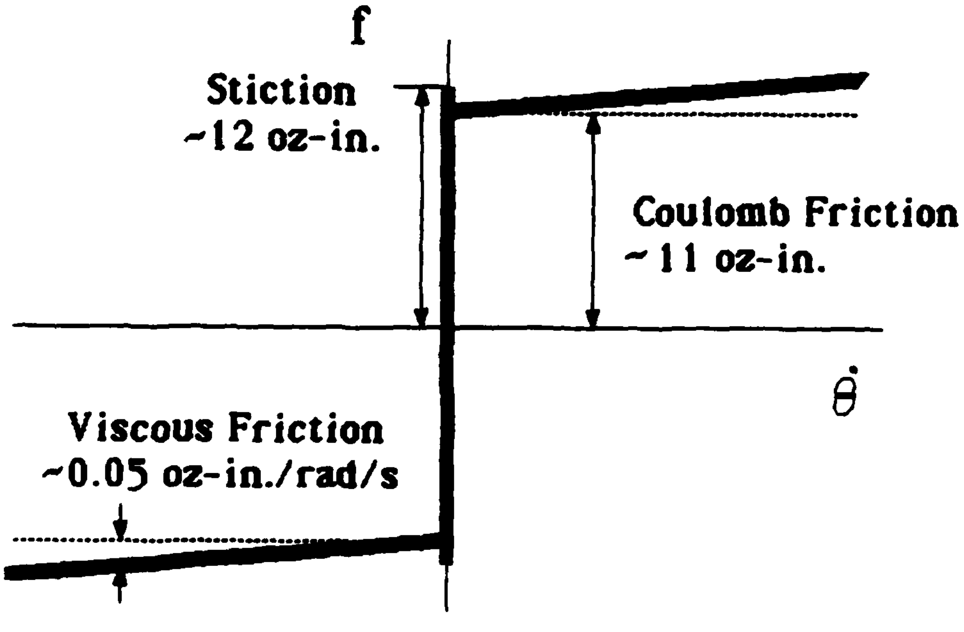
A model of the two-link arm is shown in Fig. 3a.-5. Since the two intermediate gearing stages are kinematically coupled with the motors, only eight states are required to describe the system:

$$\begin{Bmatrix} q_S \\ u_S \\ q_U \\ u_U \\ q_E \\ u_E \\ q_F \\ u_F \end{Bmatrix} = \begin{Bmatrix} \textit{Shoulder motor position} \\ \textit{Shoulder motor rate} \\ \textit{Upper arm position} \\ \textit{Upper arm rate} \\ \textit{Elbow motor position} \\ \textit{Elbow motor rate} \\ \textit{Fore arm position} \\ \textit{Fore arm rate} \end{Bmatrix}$$

The non-linear equations of motion for this system are shown in Fig. 3a.-6, where the intermediate symbols are:



Motor Position vs. Time



Friction Test of Motor

a) Experimental Pendulum Decay

b) Resulting Friction Model

Fig. 3a.-4

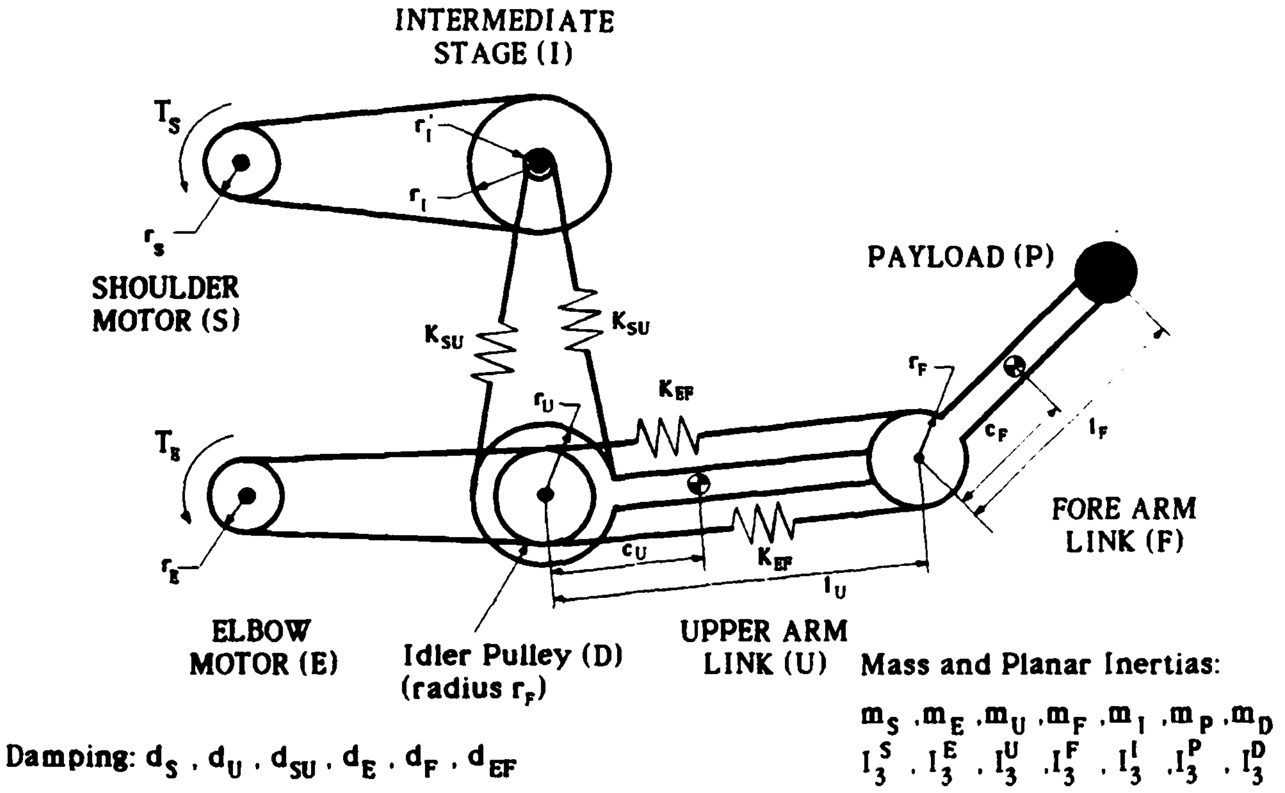

INTERMEDIATE STAGE (I)
T_S
r_I'
r_I
r_S
SHOULDER MOTOR (S)
PAYLOAD (P)
K_{SU}
K_{SU}
r_F
K_{EF}
r_U
T_E
c_F
l_F
FORE ARM LINK (F)
c_U
K_{EF}
r_E
l_U
ELBOW MOTOR (E)
Idler Pulley (D) (radius r_F)
UPPER ARM LINK (U)
Mass and Planar Inertias:
m_S ,m_E ,m_U ,m_F ,m_I ,m_P ,m_D
I_3^S , I_3^E , I_3^U ,I_3^F , I_3^I ,I_3^P , I_3^D
Damping: d_S , d_U , d_{SU} , d_E , d_F , d_{EF}

Fig. 3a.-5 Two Link Arm Model

$$
\begin{aligned}
\kappa_S &= 4K_{SU}r_U^2, \text{ shoulder torsional spring constant}\\
\kappa_E &= 2K_{EF}r_F^2, \text{ elbow torsional spring constant}\\
n_S &= n_1n_2, \text{ total shoulder gear ratio}\\
n_E &= n_3, \text{ elbow gear ratio}\\
n_1 &= r_S/r_I, \text{ shoulder motor to intermediate stage gear ratio}\\
n_2 &= r_U/r_I', \text{ intermediate stage to shoulder joint gear ratio}\\
n_3 &= r_E/r_F, \text{ elbow gear ratio}\\
I_S &= I_3^S + n_1^2 I_3^I, \text{ effective shoulder motor inertia}\\
I_U &= I_3^U + m_U c_U^2 + m_P \ell_U^2 + m_F l_U^2, \text{ effective shoulder joint inertia}\\
I_E &= I_3^E + n_3^2 I_3^D, \text{ effective elbow motor inertia}\\
I_F &= I_3^F + I_3^P + m_F c_F^2 + m_p \ell_F^2, \text{ effective elbow joint inertia}\\
I_{FU} &= m_F c_F \ell_U + m_P \ell_U \ell_F, \text{ inertia coupling term}\\
\delta_S &= 2d_{SU}r_U^2, \text{ torsional shoulder damping}\\
\delta_E &= 2d_{EF}r_F^2, \text{ torsional elbow damping}\\
\Delta &= I_U I_F - I_{FU}^2 \tilde{c}_{FU}^2, \text{ a mass matrix determinant}\\
\tilde{c}_{FU} &= \cos(q_F - q_U), \text{ cosine of relative elbow angle}\\
\tilde{s}_{FU} &= \sin(q_F - q_U), \text{ sine of relative elbow}
\end{aligned}
\tag{6}
$$

and the physical parameters are:

K_{SU}, K_{EF} : Linear spring constant between shoulder motor drive and upper arm; and elbow motor drive and forearm, respectively.

$r_S, r_I, r_I', r_U, r_E, r_F$: Radii of shoulder motor pulley, intermediate stage input, intermediate stage output, upper arm pulley, elbow motor pulley, and forearm pulley, respectively.

$I_3^S, I_3^I, I_3^U, I_3^P, I_3^E, I_3^F, I_3^D$: Planar inertias of the shoulder motor, intermediate stage, upper arm, and payload, elbow motor, forearm, and idler pulley, respectively, as calculated about each body's center of mass.

$d_S, d_U, d_{SU}, d_E, d_F, d_{EF}$: Viscous friction about the shoulder motor, upper arm joint, between the shoulder motor and joint, the elbow motor, the forearm joint, and between the elbow motor and joint, respectively.

m_U, m_P, m_F : Mass of upper arm, payload, and forearm respectively.

c_U : Distance from upper arm joint to upper arm center mass.

ℓ_U : Distance from upper arm joint to elbow joint.

c_F : Distance from forearm joint to forearm center mass.

ℓ_F : Distance from forearm joint to payload center mass.

For purposes of identification, a linearized set of equations for a particular configuration can be used. The equations of motion conveniently decouple for the case of the forearm link 90 degrees relative to the upper arm and small angular velocities. The two resulting decoupled sets of equations of motion are given by state equations very similar to those obtained for the locked elbow case in Eq. 1. Thus, the complete system parameters can be identified by two tests.

The system parameters were identified by running a sine sweep test of each part of the decoupled system. In each case the input was the voltage command to the motor which generates torque T_S (or T_E) and the output was the voltage from the tachometer which measures rate u_S (or u_E). From the state-space description of the decoupled systems , the transfer function from shoulder tachometer to the shoulder motor is given by the same equation as Eq. 3 and the transfer function from the elbow tachometer to the elbow motor is given by a similar equation:

$$\frac{u_E}{T_E/I_E} = \frac{s^2 + 2\varsigma_z\omega_z s + \omega_z^2}{(s+d)\,(s^2 + 2\varsigma_p\omega_p s + \omega_p^2)} \tag{7}$$

where the frequency domain parameters are given by equations similar to Eq. 4 including the "reduced" inertia and damping:

$$
\begin{aligned}
\omega_z &= \sqrt{\kappa_E/I_F} \\
\omega_p &= \sqrt{\kappa_E/I_R} \\
d &= \frac{d_F n_E^2 + d_E}{I_F n_E^2 + I_E} \\
\zeta_z &= \frac{1}{2\omega_z}\left(\frac{\delta_E + d_F}{I_F}\right) \\
\zeta_p &= \frac{1}{2\omega_p}\left(\frac{\delta_E + d_R}{I_R}\right) \\
I_R &= \frac{I_E I_F}{I_F n_E^2 + I_E} \\
d_R &= \frac{d_E d_F}{d_F n_E^2 + d_E}
\end{aligned}
\tag{8}
$$

Numerical values of the measured frequency domain parameters, the intermediate parameters and the physical parameters for the two-link manipulator are listed in Table 3a.-2. Note that the planar inertias, masses and link centers of mass locations were not explicitly identified. Only "lumped" parameters such as I_F could be identified. The damping parameters are somewhat artificial since we are trying to fit linear viscous damping to non-linear Coulomb friction and stiction.

The experimentally determined transfer function and the model transfer function and the model transfer function match very closely except in the very low frequency range. The mis-match is due to motor stiction and Coulomb friction which dominate the motion at the low torque levels commanded by the sine-swep equipment at the lower frequencies. The magnitude of the Coulomb friction in each motor was identified by simple pendulum tests to be about 11 os-in. and is successfully compensated to a residual of about 3 os-in for the controllers described in this paper.

Simulation of Two-Link Arm

Simulations of the linearized and the full non-linear equations of motion of the two-link arm, with graphical output showing the motion of the arm, were developed. The physical parameters of the actual system were used in the simulated theoretical model to verify the accuracy of the model. By comparing the model output and the actual output of the arm, the model was revised to better match the experimental apparatus. Simulations of the revised theoretical model have allowed us to study and to verify physical characteristics of the system, to verify (in the future) closed loop control designs, to avoid dangerous situations, and to extrapolate performance to systems with different physical parameters that cannot be investigated with the experimental apparatus in our lab.

The simulations of the equations of motion were written in Pascal, with the graphics output program written in C, on a Macintosh computer. The non-linear equations of

Frequency Domain Parameters	Intermediate Parameters	Physical Parameters
— Shoulder —	κ_S = 117.2 N-m/r	K_{SU} = 7000 N/m
ω_z = 1.71 Hz	κ_E = 58.6 N-m/r	K_{EF} = 7000 N/m
ζ_Z = 0.03	n_S = 0.165	r_S = 0.026 m
ω_P = 9.10 Hz	n_E = 0.344	r_I = 0.065 m
ζ_P = 0.10	n_1 = 0.406	r'_I = 0.026 m
d = 0.10 Hz	n_2 = 0.406	r_U = 0.065 m
	n_3 = 0.344	r_E = 0.022 m
— Elbow —	I_S = 0.0010 kg-m^2	r_F = 0.065 m
ω_z = 2.61 Hz	I_U = 1.02 kg-m^2	d_S = 0.007 N-m/r/s
ζ_Z = 0.03	I_E = 0.0013 kg-m^2	d_U = 0.402 N-m/r/s
ω_P = 11.9 Hz	I_F = 0.22 kg-m^2	d_{SU} = 30.1 N/r/s
ζ_P = 0.10	δ_S = 0.252 N-m/r/s	d_E = 0.006 N/r/s
d = 0.10 Hz	δ_E = 0.123 N-m/r/s	d_F = 0.091 N/r/s
		d_{EF} = 14.7 N/r/s
		l_U = 0.508 m
		l_F = 0.485 m

Table 3a.-2

Two-Link Arm Identified Parameters

Note: The planar inertias, masses, and link center of mass location were not explicitly identified. The damping parameters are somewhat artificial since we are trying to fit linear viscous damping to non-linear Coulomb friction and stiction.

motion used are described in Fig. 3a.-6. For brevity, the simulation code and graphics code are not included here. Some of the interesting results from the simulations are included in figures 3a.-7 and 3a.-8.

Fig. 3a.-7 shows the graphical output from the simulation (linearized equations of motion) when the initial conditions are such that the natural frequencies, and hence the mode shapes, are excited. Since the system is non-linear, there are different natural frequencies (and mode shapes) for each elbow angle configuration. Also, due to the two elastic modes of the system, there are two natural frequencies, designated high and low, associated with each configuration. As stated earlier, the system decouples at an elbow angle of 90 degrees and is easily seen by the mode shapes. As illustrated in the high frequency regime, the node about which the modal oscillation takes place starts out at the tip and moves in as the elbow angle is increased to 90 degrees, then it moves out again. As expected, the mode shapes for the extended arm look like those of a flexible beam.

Fig. 3a.-8 shows the variation in system poles and transmission zeros as a function of the elbow angle. The rigid body poles are not shown. The poles and transmission zeros are symmetric about an elbow angle of 90 degrees. Also, as shown in the figure, the poles and transmission zeros change significantly due to changes in elbow angle. This is due to the non-linearities of the system, and reveals the difficulty in achieving robustness in the control system design.

Control System Design, Analysis, and Experimental Results

Two different controllers for the two-link arm are described in this section. The first controller described is a simple PID colocated controller which shows the performance of the system without the use of the end-point sensor. It is the comparison case to show the improvements that end-point sensing provides over current robot controllers. The second controller described is a discrete, state-space designed end-point controller which represents our best results to date. Finally, the effects of gravity on the flexible manipulator system are discussed.

The initial experiments performed with the two-link manipulator have concentrated on studying feedback control for small regions of motion where the manipulator dynamics is linear and decoupled, i.e., the relative elbow angle is about 90°. This report describes and compares two controllers designed for this configuration. The first controller was designed using successive-loop-closure of rate and position about the motor only. This represents current methods of manipulator feedback control that use only colocated sensors and actuators and do not compensate for flexibility in the system. The second controller was designed using state-space optimal control methods with the primary sensor being the end-point sensor which was non-colocated with the motors.

Colocated Control

Figure 3a.-9 is a block diagram of a classically designed, successive-loop closure controller for the two-link arm that does not use the end-point sensor. The innermost loop is Coulomb-friction compensated (a constant torque in the direction of rotation of the motor is commanded). This loop successfully reduces the friction by about 70°, but small limit

$$\begin{bmatrix} \dot{q}_S \\ \dot{u}_S \\ \dot{q}_U \\ \dot{u}_U \\ \dot{q}_E \\ \dot{u}_E \\ \dot{q}_F \\ \dot{u}_F \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-\kappa_S n_S^2}{I_S} & \frac{-(d_S+b_S n_S^2)}{I_S} & \frac{\kappa_S n_S}{I_S} & \frac{b_S n_S}{I_S} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{I_F \kappa_S n_S}{\Delta} & \frac{I_F b_S n_S}{\Delta} & \frac{-I_F \kappa_S}{\Delta} & \frac{-I_F(d_U+b_S)}{\Delta} & \frac{-I_{FU}\tilde{c}_{FU}\kappa_E n_E}{\Delta} & \frac{-I_{FU}\tilde{c}_{FU} d_E n_E}{\Delta} & \frac{I_{FU}\tilde{c}_{FU}\kappa_E}{\Delta} & \frac{I_{FU}\tilde{c}_{FU}(d_F+b_E)}{\Delta} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-\kappa_E n_E^2}{I_E} & \frac{-(d_E+b_E n_E^2)}{I_E} & \frac{\kappa_E n_E}{I_E} & \frac{b_E n_E}{I_E} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-I_{FU}\tilde{c}_{FU}\kappa_S n_S}{\Delta} & \frac{-I_{FU}\tilde{c}_{FU} d_S n_S}{\Delta} & \frac{I_{FU}\tilde{c}_{FU}\kappa_S}{\Delta} & \frac{I_{FU}\tilde{c}_{FU}(d_U+b_S)}{\Delta} & \frac{I_U \kappa_E n_E}{\Delta} & \frac{I_U b_E n_E}{\Delta} & \frac{-I_U \kappa_E}{\Delta} & \frac{-I_U(d_F+b_E)}{\Delta} \end{bmatrix} \begin{bmatrix} q_S \\ u_S \\ q_U \\ u_U \\ q_E \\ u_E \\ q_F \\ u_F \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{I_{FU}^2 \tilde{c}_{FU}\tilde{S}_{FU}}{\Delta} & \frac{I_F I_{FU}\tilde{S}_{FU}}{\Delta} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{-I_U I_{FU}\tilde{S}_{FU}}{\Delta} & \frac{-I_{FU}^2 \bar{C}_{FU}\tilde{S}_{FU}}{\Delta} \end{bmatrix} \begin{bmatrix} u_U^2 \\ u_F^2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_S} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_E} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_S \\ T_E \end{bmatrix}$$

Fig. 3a.-6 Nonlinear equations of motion for the two-link manipulator. Explicit formulation is shown.

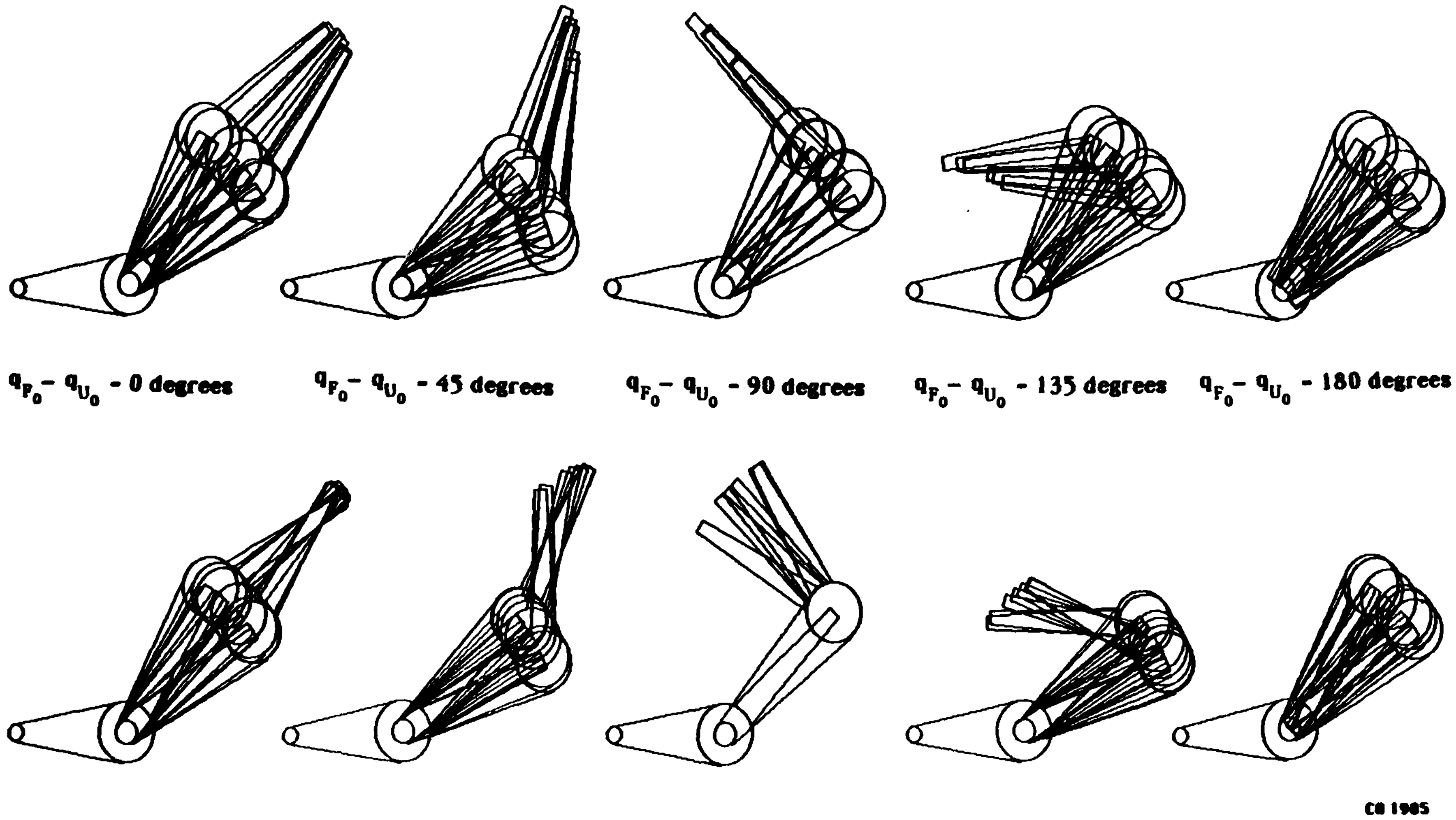

$q_{F_0} - q_{U_0}$ - 0 degrees
$q_{F_0} - q_{U_0}$ - 45 degrees
$q_{F_0} - q_{U_0}$ - 90 degrees
$q_{F_0} - q_{U_0}$ - 135 degrees
$q_{F_0} - q_{U_0}$ - 180 degrees
CB 1985

Fig. 3a.-7

Two-Link Arm Mode Shapes
a) Low Frequency Mode Excited
b) High Frequency Mode Excited

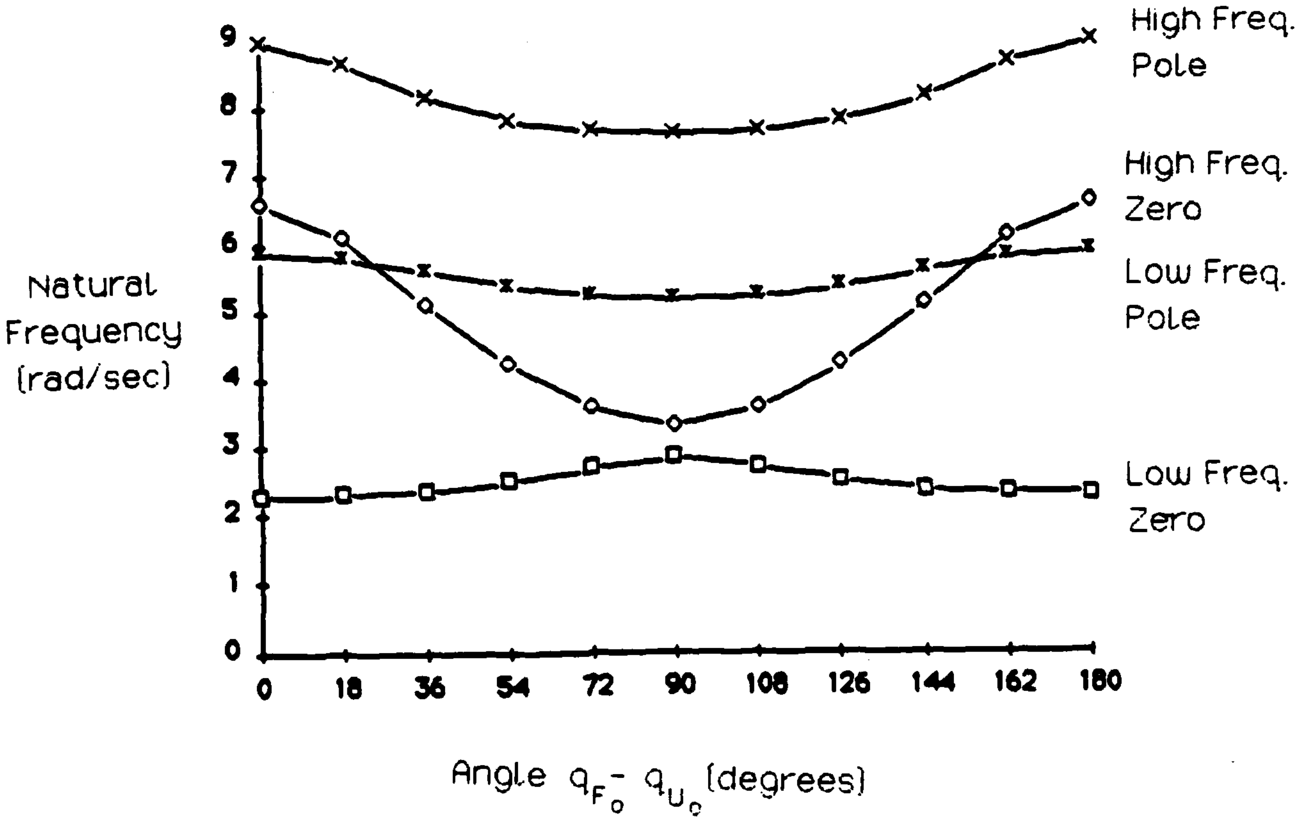


Fig. 3a.-8

Two-Link Arm Imaginary Poles and Zeros as a Functicn of Elbow Angle

Sensors at motors (Colocated transfer function)

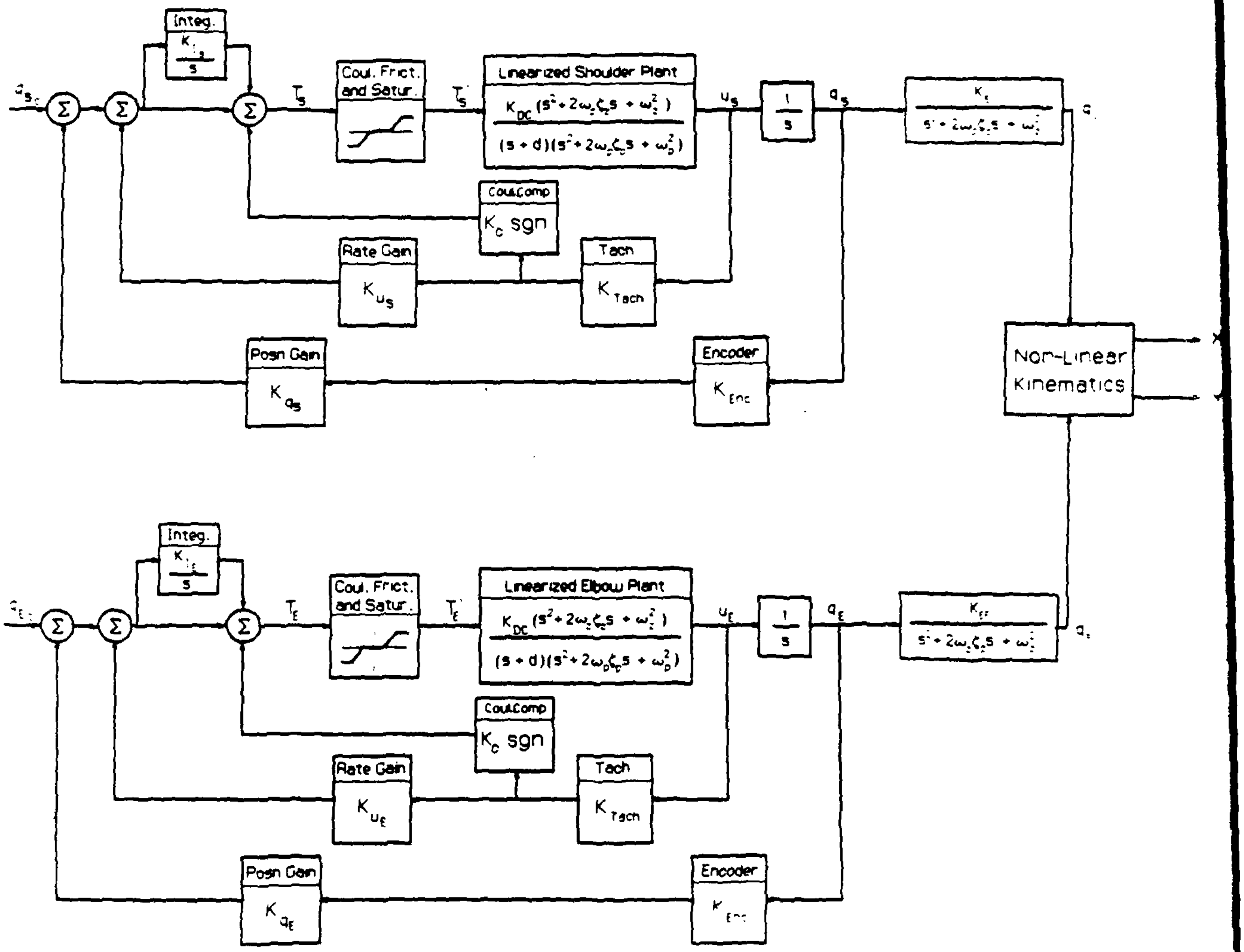


Fig. 3a.-9 Block Diagram for Colocated PID Control

cycles have appeared during quiescent periods when motor rates are near zero. In the next loop, rate is fed back from the tachometers located on the motors to provide damping to the system. In the outermost loop, position is fed back from the optical encoder on the motors. Integral control is applied to the motor position error signal to take out biases in the position of the motor. Note that the end-point position can be in great error even if the motor positions are tightly controlled. This is because any friction, stiction, flexibility, and disturbance torques located further down the drive train are not sensed and compensated. This illustrates one of the primary disadvantages of traditional "dead reckoning" control strategies.

Figure 3a.-10 shows the Evan's root loci design for the rate and position loops for both the shoulder and elbow sub-systems. For the desired critically or near critically damped closed-loop positions poles, the zeros of the plant "capture" these poles near the origin and limit the closed-loop bandwidth to about 50° to 60° of the plant zero frequencies. This agrees well with the traditional "rule-of-thumb" of limiting the closed-loop bandwidth to one-half of the "structural" vibration frequency of the manipulator, i.e., the vibration frequency of the manipulator with the motors locked, which corresponds to the plant complex zeros for the colocated transfer function (Equations 8 and 10). Assuming plant DC gains of 1.0, the rate loop gains are 205 and 225, and the position gains are 510 and 810 for the shoulder and elbow sub-systems respectively. The controller was implemented on the PDP 11/24 digital computer at a sample rate of 200 Hz which is very fast compared to the closed-loop poles and thus very closely approximates continuous control.

Non-colocated End-Point Control

Figure 3a.-11 shows the block diagram for the non-colocated state-space controller for the two-link arm. At the time of this writing, only the linear feedback part of the two-link state-space control had been achieved experimentally. The non-linear feedforward part of the control is still under development. Once again, an innermost loop for Coulomb friction compensation was implemented. A full-state-feedback control law was designed using a Discrete Linear Quadratic Regulator design program called DISC. The eight states are determined by measuring the motor positions, motor rates and joint rates directly and performing a kinematic inversion of the x-y end-point measurement to obtain the joint position states. The steady-state, discrete regulator gains $\underline{\mathbf{C}}$ in $\mathbf{T} = - \underline{\mathbf{C}}\, \underline{\mathbf{X}}$ are obtained by minimizing the discretized expression of the continuous performance J_T.

$$J_T = \int_O^\infty (\underline{x}^T \mathbf{A} \underline{x} + T^2 \mathbf{B}) dt \tag{9}$$

where, for the shoulder sub-system, the state vector and control are:

$$\underline{x} = (q_S \; u_S \; q_U \; u_U)^T$$

$$T = T_S \tag{10}$$

and for the elbow sub-system, the state vector and control are:

$$\underline{x} = (q_E \; u_E \; q_F \; u_F)^T$$

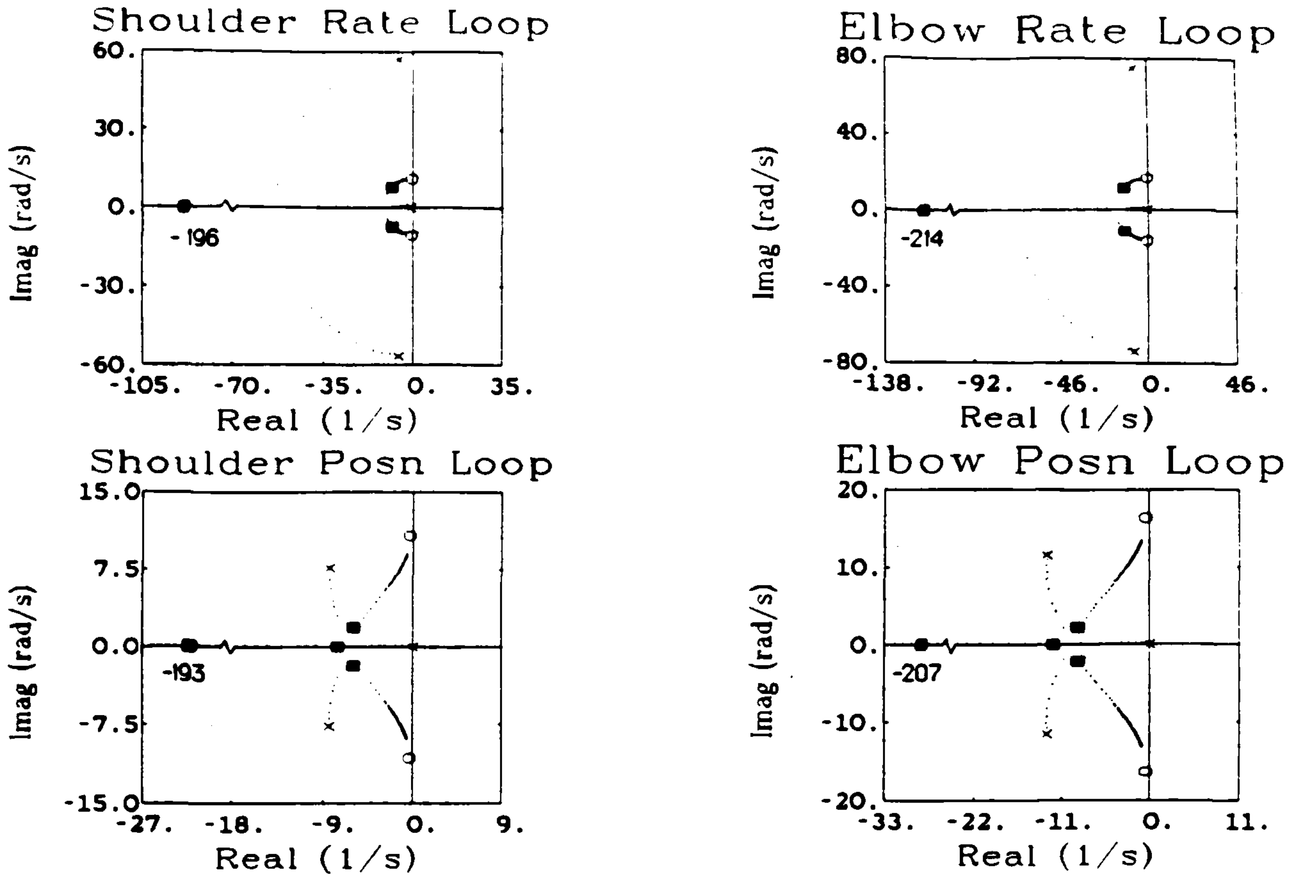


Fig. 3a.-10 Control Design in the s-plane using succesive loop closing: locus of roots versus rat and position gain for the shoulder and elbow sub-systems.

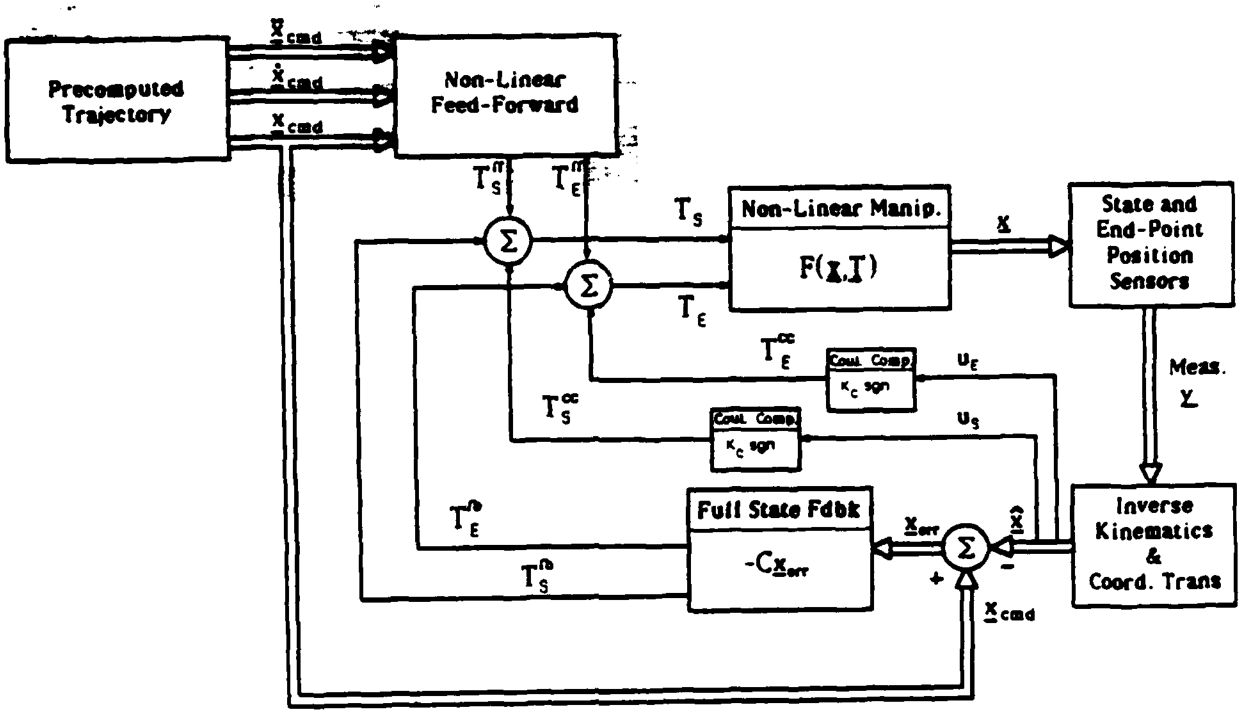


Fig. 3a.-11 Non-colocated Full-state Feedback Control

(Shaded region not currently implemented.)

$$T = T_E \qquad (11)$$

The units were scaled to be: position q's in 0.01 radians, rate u's in rad/sec, and torque T in 0.1 N-m. The state and control weighting matrices were chosen to give the most weighting to the tip position and velocity to insure good performance at the tip:

$$\begin{aligned} \mathbf{A} &= \text{diag}(1\ 5\ 800\ 400), \quad \mathbf{B} = (50) \quad \text{(Shoulder)} \\ \mathbf{A} &= \text{diag}(1\ 5\ 100\ 200), \quad \mathbf{B} = (50) \quad \text{(Elbow)} \end{aligned} \qquad (12)$$

Using these weighting matrics for the shoulder and the elbow sub-systems and choosing a sample rate of 50 Hz gave these shoulder and elbow control matrices:

$$\begin{aligned} \underline{C_S} &= (0.0019\ 0.3411\ 2.4899\ 26.0369) \\ \underline{C_E} &= (-0.0521\ 0.3371\ 1.1723\ 10.7878) \end{aligned} \qquad (13)$$

The resulting shoulder closed-loop poles are located at:

$$\begin{aligned} z &= 0.305 \pm 0.629j,\ 0.796 \pm 0.17; \ \ (z-\text{plane}, 50Hz) \\ s &= -17.9 \pm 55.9j,\ -10.3 \pm 10.6j \ \ (s-\text{plane}) \end{aligned} \qquad (14)$$

The resulting elbow closed-loop poles are located at:

$$\begin{aligned} z &= 0.073 \pm 0.777j,\ 0.812 \pm 0.157j \ \ (z-\text{plane}, 50Hz) \\ s &= -12.4 \pm 73.9j,\ -9.56 \pm 9.54j \ \ (s-\text{plane}) \end{aligned} \qquad (15)$$

Experimental Results

For comparison of the two controllers, an identical step in end-point position of 35 cm was commanded to each controller. Figure 3a.-12 shows the response of both controllers as seen in Cartesian x-y space. The dashed line in each plot represents the colocated classical controller and the solid line represents the non-colocated end-point controller. The top plot shows that the trajectories in x-y space are roughly equivalent in shape. Neither controller has path feedforward so the paths are not quite straight lines. The x and y time response plots indicate that the end-point controller slewed to the vicinity of the terminal position in 0.2 sec with an average slew speed of 1.7 m/sec whereas the colocated controller took 0.9 sec with an average slew speed of only 0.4 m/sec. Both plots of the x and y time histories of the step show critical or near critical damping in the response for the colocated controller and the non-colocated controller has a slight overshoot. The settling times for the non-colocated controller was about twice as fast as for the colocated controller. (Note that the residual oscillations at the terminal position are about 0.6 cm in amplitude, composed, in part, of sensor noise during data acquisition.) The actual positioning accuracy is about 0.5 cm. Increased damping in the control design and further tweaking of Coulomb friction compensation should bring the positioning accuracy to within the desired 0.1 cm.

The plots in Figure 3a.-13 show the exact same step command as seen in joint space. Again, the dashed lines indicate the response of the colocated controller and the solid lines indicate the response of the non-colocated end-point controller. The speed of response is about three to four times faster for the non-colocated end-point controller. Note the completely different time histories of motor position and commanded torque for the two controllers. The end-point controller commands two large reversals in torque during the slew maneuver. The motor first winds up the springs in the direction of travel to transmit torque out to the link. Then the motor reverses the torque command to wind the springs in the direction opposite of travel to slow the link down. Finally, the motor reverses again just as the link is approaching the terminal position to unwind the springs. Residual torque oscillations damp any remaining motion in the links. These results are very similar to those obtained for the simple continuously flexible beam experiments reported by Cannon and Schmitz (except for the non-minimum phase character of the continuously flexible system which caused an initial reversal of tip motion after a small time delay while a transient wave propagates down the flexible beam).

Compensating The Effects of Gravity

There are two primary effects of gravity on our flexible manipulator system. The first effect is large DC torques that are a non-linear function of the position of the manipulator are introduced. This effect has, of course, already been adequately studied for rigid manipulators and can be compensated using several well-known techniques including integral control, feedforward compensation using calibration data tabulated in software or calculated, and physically counter-weighting the system. The second effect on the two-link flexible manipulator and all other flexible manipulators is the change in dynamic response of the system. The vibration frequencies and mode shapes are a non-linear function of position of the manipulator in the gravity field.

We conclude that compensation for gravity effects is about as complex as the compensation required for the flexibility alone. The non-linear feedforward technique described for the second controller can be applied almost directly for gravity compensation since the accuracy of the plant model required for the non-linear dynamics feedforward will suffice for the gravity compensation. Gravity compensation is one of the functions in the non-linear feedforward block in Fig. 3a.-11.

The two-link manipulator was constructed so that gravity effects could be studied. The base plate was designed to be tilted up to 15 degrees to include up to one-fourth of the gravity component. As indicated above, the non-linear feedforward block of the last controller has not yet been implemented experimentally. Thus, the experimental results for gravity compensation will be reported later.

Summary

This summary directly addresses the sections in the Statement of Work that pertain to the research described above, namely, sections n and o. All of the objectives have been completed at least to the level of theoretical considerations and the key experimental results have been achieved that positively demonstrate the usefulness of end-point control techniques for flexible, multi-link manipulators.

Section n. directed to "extend the end-point control laws developed for single-link flexible arm operation to two-link operation in the horizontal plane and investigate extension

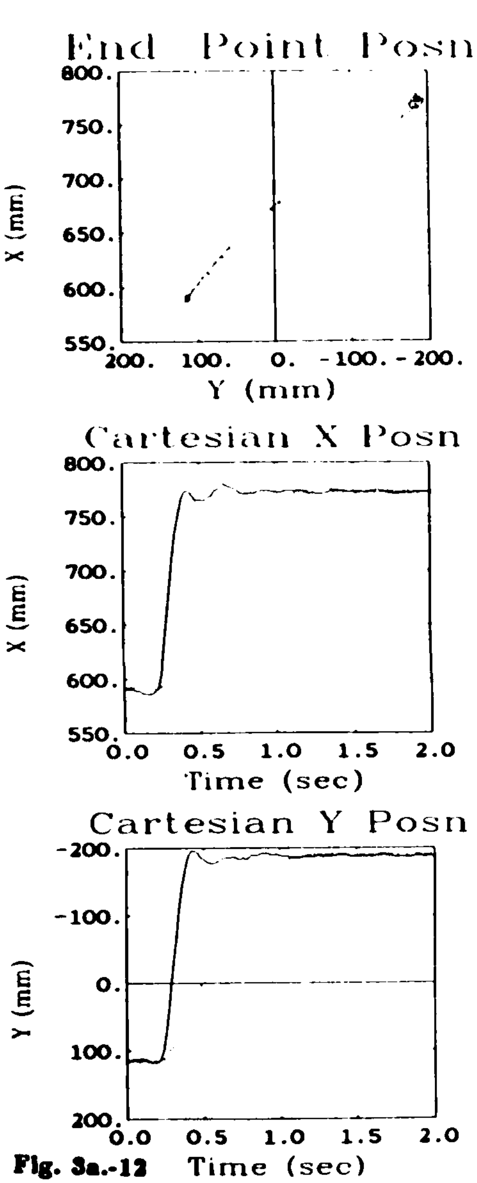


Fig. 3a.-12 Response of the two-link arm to a step command of 35 cm. X-Y trajectory and time history plots. Dotted line for colocated control and solid line for non-colocated control.

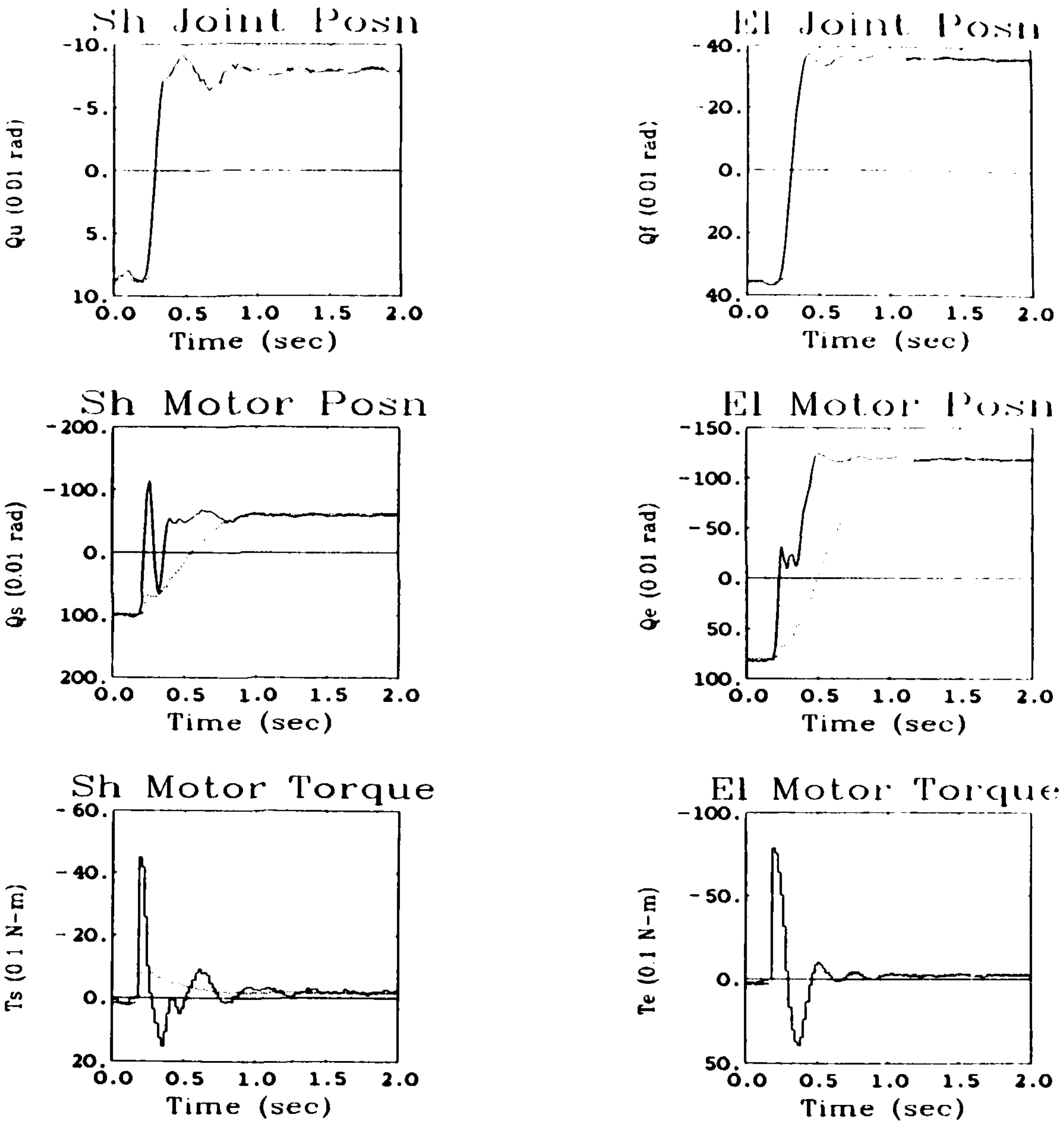


Fig. 3a.-13 Response of the two-link arm to a step command of 35 cm. Time histories of joint position, motor position, and commanded torque for the shoulder and elbow sub-systems. Dotted line for colocated control and solid line for non-colocated control.

to the vertical plane dimensions where gravity affects the system."

Both a classical PID designed and a modern state-space designed end-point position controlled two-link flexible-tendon-drive manipulator have been built and demonstrated. Advanced control techniques beyond those developed for the one-link arm have been theoretically analyzed and are close to experimental demonstration. The effects of gravity were considered theoretically and found to be fairly complicated, but attainable with techniques developed in our lab.

Section o. directed to "investigate the capabilities and limitations of lumped-compliance robot arms. Specifically, determine the slew speed, end point settling time and positioning accuracy, proximity motion, and contact-force performance for a range of flexibility, payload mass and inertia, actuator power, and controller bandwidth; and study inherent actuator and sensor non-linearities."

By "capabilities and limitations of lumped-compliance robot arms", we realize that ALL manipulators have inherent flexibility (perfectly rigid structures can not actually exist) and it is precisely this inherent flexibility that has impaired robot performance by limiting bandwidth and requiring heavy, stiff linkages to be built to obtain reasonable accuracy. With end-point control technology, all of the above mentioned robot performance criteria can be greatly improved. The increase in slew speed and end-point settling time of the end-point controlled robot over the co-located controller tested was about a factor of 3 to 4. These are not theoretical limitations, but were due to practical considerations such as the drive belts slipping off of the gears if too much torque was commanded.

The positioning accuracy can be easily seen to have much better performance with direct sensing of the end-point position than inferring the position using sensors on the actuators. Any flexibility in the drive train will corrupt the final estimate of the end-point position. The use of full-state estimators which model the flexibility can close the gap between colocated sensors and directly sensing the end-point. However, directly sensing the end-point position with a good sensor (at least as accurate as the colocated sensors) represents the asymptotic "best" performance that computer estimators can possibly achieve, and the estimators would need extremely accurate models, and very fast computers.

For the manipulator system we studied, which *did* have stiff links, the joint sensors could be used as a surrogate to the end-point sensor since no significant structural flexibility lies between the joints and the end-point. However, even in this case, the end-point sensor has five distinct advantages over the joint sensors. One, the kinematics of the robot must be extremely well calibrated to accurately infer end-point position. In fact, calibration of joint-axis misalignment, link lengths, and colocated sensor biases represent a major expense on current robots. Even small collisions can knock the calibrations out of whack and time and money must be spent on recalibration. The end- point sensor can be isolated from collisions and calibrated just once. Second, the end-point position inferred from joint sensors is propagated along the entire chain of links and these errors can grow much larger compared to direct sensing. Third, end-point position inferred from joint positions has computational singularities when the joint-space-to-Cartesian-space Jacobian becomes singular. Also the end-point sensor accuracy is uniform over the entire operating range of the robot whereas joint-position-inferred end-point position accuracies are a non-linear function of the joint space and deteriorate rapidly near the Jacobian singularities. Fourth, the computational power required to compensate flexibility is the same using either sensor system, so why not use the superior end-point sensing. Fifth, and most importantly, end-point sensing can handle the most difficult case of structural flexibility as well as drive train flexibility whereas the joint sensors fail and stiffness once again limits the bandwidth.

Contact force performance is inherently improved for lumped-compliance robot arms. Most obvious is the greatly decreased impact forces from both inadvertent collisions and desired, but misaligned contact since the flexible drive train can absorb some of the impact shock. Also, because the robots can be made much lighter, contact forces can be greatly reduced. Experiments on controlled contact and force control have not been performed yet on the two-link arm in our lab. Equipment is currently under fabrication. However, force control experiments on the one-link flexible manipulator (performed under DARPA) indicate that the improved contact performance achieved on the one-link arm can be transferred to the two-link arm without much difficulty.

The modelling and analysis gives a clearer understanding of the manipulator performance under a range of flexibility, payload mass and inertia, actuator power and bandwidth. As indicated by the analysis, the plant zeroes (which effectively limit response) are primarily a function of the drive train spring constant and the inertia outboard of the springs. Thus, decreasing the spring constant or increasing the payload mass and inertia will decrease the robot transmission zero frequencies and the performance.

Actuator power considerations can be addressed by considering the optimal controller for limited power. This is the so-called "bang-bang" control which either turns the motor full forward, full reverse, or off. As indicated in the last section, the state-space controllers we studied came close to the optimal "bang-bang" control. Controller bandwidth for this case increases proportionally with available torque. Thus, it appears that we can theoretically apply almost all available motor torque using our end-point control techniques. However, for our robot, the drive belts slipped long before full torque is applied. This can be more carefully considered when constructing robots for real applications.

Inherent actuator and sensor non-linearities were carefully studied and assessed during our research since these represented many of our limitations. Effective compensation for motor friction is essential for good control. A combination of high loop-gain rejection of the "friction disturbance" and feedforward compensation of the friction gave us good compensation of the motor friction. Other motor non-linearities such as torque ripple and cogging torque should be eliminated at the outset of the design process by careful selection of the motors and power amplifiers. Another technique we are planning to study will bypass most of the actuator non-linearity problems by directly closing a tight analog torque loop between the motor and a force/torque sensor on the joint being driven. Sensor non-linearities must also be carefully assesed, especially for the final position accuracy. We achieved acceptable sensor performance by careful calibration and filtering of known noise sources.

TASK 3b.
VERY FLEXIBLE MANIPULATOR WITH FAST WRIST

Introduction

In current robot control practice, sensor and actuator are always colocated. Joint angles are measured with encoders at all motor shafts. The end effector's position and direction are calculated using coordinate transformation from the set of joint angle measurements. This method assumes all links to be rigid, and the control bandwidth is chosen low enough not to excite the structural flexibility, which results in oversized structure and very low control bandwidth.

Previous research in the Stanford Aeronautics and Astronautics Robotics Laboratory in the control of flexible manipulators includes : (1) End-point position control of a flexible beam by Eric Schmitz [Schmitz 1], and (2) Contact force control of a flexible beam by Jim Maples [Map 1]. They both used end-point sensors that are not colocated with the actuator to get high performance. In order to design high-performance controllers, they identified the linear dynamic model of the system and implemented a state estimator and state-feedback regulator using the LQG modern control techniques. Reduced-order robust, i.e., parameter insensitive, controllers were also designed to control the flexible beams. Both the LQG modern control technique and the reduced-order robust control design require the description of the plant dynamics in terms of a set of linear equations. The performance of the system under closed-loop control depends on the accuracy of the linear model describing the plant dynamics during the design phase of the control system. System performance and controller robustness can be traded off depending on the accuracy of the linear model describing the plant dynamics.

Schmitz has found that there exists a theoretical limitation on the control speed, even in the extreme case where the plant dynamics are linear and known exactly. Some non-minimum phase zeros exist in the transfer function from the actuator input (the hub motor) to the tip position (the end point) of a flexible beam due to the structural flexibility and the separation of the end point sensor from the actuator. Those non-minimum phase transfer zeros limit the maximum achievable closed-loop bandwidth. (This bandwidth limitation is of course directly related to the time for a bending wave to travel the length of the beam.)

A minimanipulator, carried by a main robot arm, can be used to improve the speed and accuracy of the motion of the end effector. Within a relatively smaller working range, a minimanipulator can achieve higher bandwidth and accuracy due to its smaller physical size and local actuation. Process time can be reduced for jobs requiring frequent small motions of the robot manipulator. Large motions can also benefit, since the motion error of the main robot can be corrected easily with the minimanipulator, and higher speed of the main robot can be tolerated.

In isolation, high bandwidth and high accuracy are not difficult to obtain for a minimanipulator due to its smaller size, and there is theoretically no difficulty in controlling it with high bandwidth when it is mounted on a fixed base or carried by a very heavy and rigid robot arm. However, it becomes more difficult to obtain fast control when the minimanipulator is carried by a main manipulator having some structural frequencies lower than the desired control bandwidth of the minimanipulator.

The idea of minimanipulator has been considered on other mechanical systems already. A fast and light secondary mirror has been be designed for the Shuttle Infrared Telescope Facility. The fast motion of the secondary mirror compensates for any pointing error of the telescope whose attitude is controlled under a slow control system. However, the mirror actuator does not excite the structure, and thus the controller design is not difficult.

Even though it is theoretically possible to achieve high performance for a control system under any condition, as long as the dynamics of the plant are linear and known exactly, a practical controller design for such a system may not be feasible due to the nonlinear effects and the modeling uncertainty of the system dynamics. It is important to know how to design and control such a system properly, that is, to get the high bandwidth and accuracy inherent in the minimanipulator and overcome the limitation set by the structural flexibility of the main robot arm.

Purpose of This Research

The objectives of this research are :

- To study and understand the general properties of the dynamic interaction between the fast motion of a minimanipulator and the structural flexibility of the main robot arm.
- To search for the proper way to design the mechanical system and the correct method to design the control system, in order to achieve high performance in the whole system, but still be insensitive to the uncertainty of the plant dynamics.
- To demonstrate several fast maneuvers of the minimanipulator with a one-dimensional experimental system.

Experimental Devices and System Dynamics

Introduction

The mechanical experiment system, a one-dimensional wrist and flexible beam system, is described here. The rigid and short wrist represents a minimanipulator, and the flexible beam represents the main robot arm with structural flexibility. The structural flexibility of the main arm is considered since it has modal frequencies lower than the desired control bandwidth of the minimanipulator.

Analyses of system dynamics are also presented. Systems analyzed include: (1) the symbolic equations of motion of a generic wrist-beam system, (2) a particular wrist-beam model for a subsequent numerical analysis, and (3) a linear model of the mechanical wrist-beam system used for maneuver demonstration.

Description of Experimental Devices

A one-dimensional plane-motion mechanical wrist-and-flexible-beam system was constructed to study the basic dynamic behavior of a mini-macro manipulator system in which structural flexibility of the main arm is not negligible. This system was then used to demonstrate achievement of several kinds of quick maneuvers under feedback control with an end point sensor. A schematic drawing of the mechanical system is shown in Fig. 3b.-1.

The wrist is a 16.5 *cm*. long beam with 2.5 × 1.4 *cm*. rectangular cross section. It is made of 0.07 *cm*. thin aluminum plate. The wrist is made rigid because it is considered that the structural flexibility of the minimanipulator can be neglected due to its small size.

A pneumatic gripper with an inflatable rubber bulb, operated by compressed air at 40 psi, is mounted near the far end of the wrist in order to pick up a specially designed object, the load, for demonstrating maneuvering tasks. The load is a light weight spool shaped object cut out of a piece of aluminum block. A center hole is made for the gripper's rubber bulb to insert in, and the load is grabbed when the bulb is inflated.

A 4.5 watt incandescent lamp is mounted on top of the pneumatic gripper to indicate the position of the end point, which is sensed by an UDT photo sensor.

The other end of the wrist is mounted on the shaft of a DC motor which is capable of delivering 25 oz-in peak torque. An RVDT made by Pickering is mounted on the motor shaft to measure the shaft angle. The motor housing is mounted on the end plate of the large flexible beam, which is similar to the ones used earlier in the same lab for previous flexible beam projects.

The flexible beam, which is 96.8 *cm*. long, consists of two parallel 0.1 *cm*. thick and 6.58 *cm*. high aluminum side plates connected by seven rectangular bridge plates with flex hinges. Those seven bridge plates link the two slender side plates together so that the beam is stiff in torsion and vertical bending, leaving the majority of the structural flexibility in the horizontal direction only. The hub of the flexible beam is mounted on the shaft of a limited-angle brushless DC motor attached to a fixed table. A potentiometer is mounted on the motor shaft to detect the shaft angle. Four pairs of strain gauges are attached to both sides of the right aluminum side plate.

First-order high-pass analog filters are used to derive the pseudo-rate information from the end point position sensor and the hub-motor potentiometer, before sensor output signals are sampled and digitized. The advantage of doing so is that more sensor information becomes available without using additional sensors, when a digital computer is used to implement the controller.

Both motors are installed with their shaft vertical, allowing only horizontal motion of the wrist-beam system. The effect of gravity is eliminated, since the long beam is flexible only in the horizontal plane.

Current amplifiers having current output proportional to input voltage are used to drive the two DC motors, so that motor command torques can be directly specified by the feedback controller. A PDP-11/23 minicomputer, running RT-11 operating system and equipped with twelve-bit A/D and D/A ports, is used to implement the controller digitally.

Dynamics of the Wrist-Beam System

Dynamics of the wrist-beam system are analyzed and presented in this section. The equations of motion in symbolic form are first studied for a generic system consisting

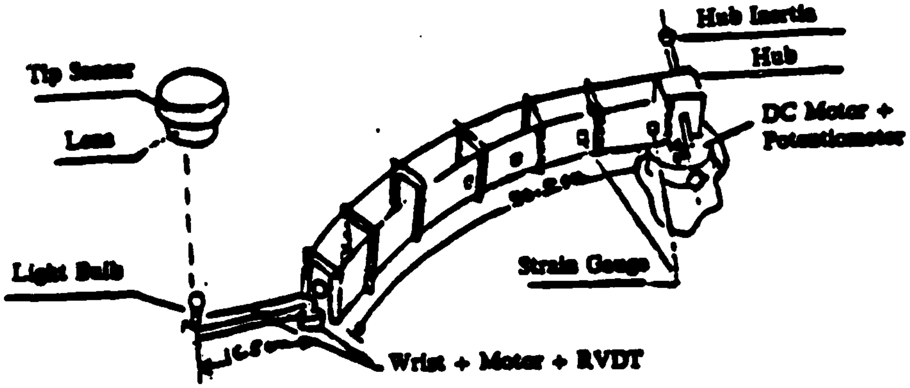


Fig. 3b.-1: The mechanical experiment system.

of a uniform flexible beam and a rigid short link. This analysis reveals a fundamental interaction between a minimanipulator and a main robot arm in the one dimensional case.

The dynamics of an idealized wrist-beam model is also analyzed for the purpose of subsequent numerical analysis. This model is chosen such that the wrist is very heavy and produces large interaction with the flexible beam. The size and mass properties of this system are assumed to be known exactly, which makes it more convenient to do theoretical analysis and investigate the ultimate system performance.

Finally, the dynamics of the mechanical experiment system are analyzed and tested, so that real time feedback controllers can be designed.

Equations of Motion of a Rigid Wrist Carried by a Uniform Flexible Beam

A generic wrist-beam model is established for the study of the basic properties of the dynamic interaction between the fast minimanipulator and the structural flexibility of the main robot arm. Fig. 3b.-2 shows a drawing of this system and the free body diagrams of its components. The minimanipulator is considered to be rigid, and is represented by the short link. The main robot arm is represented by a uniform beam with structural flexibility. One end of the flexible beam is attached to a hub where a torque motor is mounted. Another motor is located at the axis joining the flexible beam and the rigid wrist.

Variables used in the analysis are defined as:

L: Length of the flexible beam.

l: Distance between the wrist-beam joint axis and the end effector.

h: Distance between the wrist-beam joint axis and the mass center of the wrist.

y: The lateral displacement of a point in the system.

θ_f: The absolute rotation of the wrist.

θ_h: The hub angle.

I_1: The moment of inertia of the hub, including the rotor of the hub motor.

m_2: The mass at the wrist-beam joint axis, including the wrist motor stator and housing.

I_2: The moment of inertia of m_2.

θ_2: The absolute rotation of I_2.

m_3: The mass of the rigid wrist, including the end effector and the load.

I_3: The moment of inertia of m_3 at the mass center of the wrist.

u_1: The control torque of the hub motor.

u_2: The control torque of the wrist motor.

T_1: The reaction torque between the hub and the flexible beam.

T_2: The reaction torque between the flexible beam and the inertia I_2.

F_2: The reaction force between the flexible beam and the mass m_2.

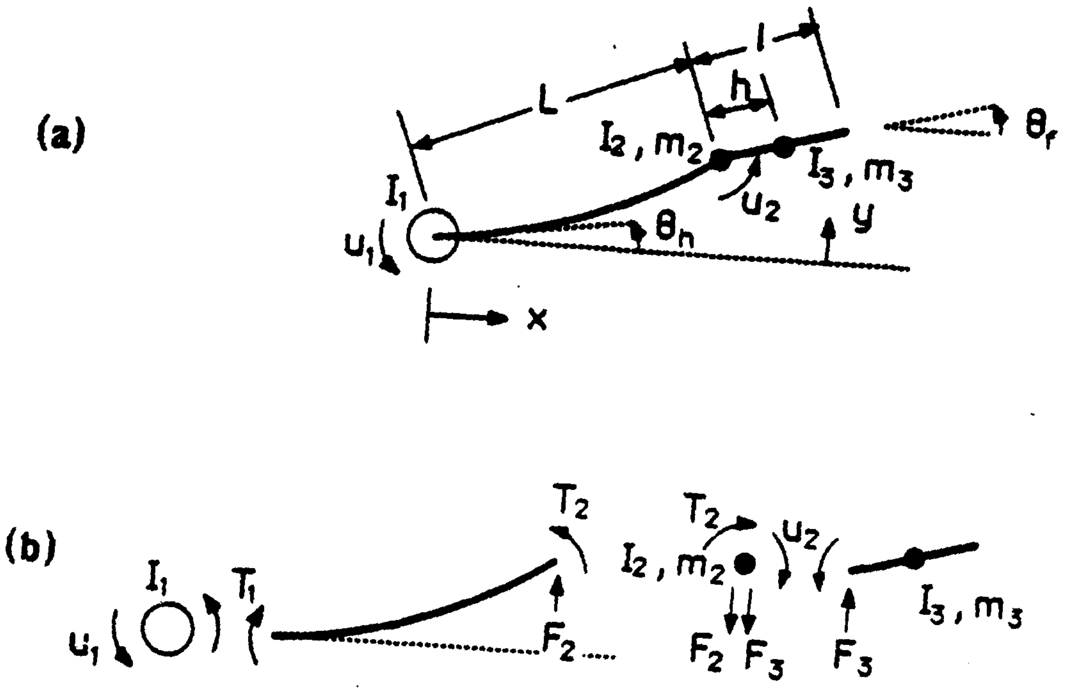


Fig. 3b.-2: The schematic drawing of (a) the generic wrist-beam system, and (b) the free body diagrams.

F_3: The reaction force between the mass m_2 and the rigid wrist.

ρ: Linear mass density of the uniform flexible beam.

E: Young's modulus of the material of the flexible beam.

I: Area MOI of the cross section of the uniform flexible beam.

Considering only small motions and neglecting any friction and damping, the equations of motion of the wrist-beam system can be expressed as:

$$T_1 + u_1 = I_1\ddot{\theta}_h = I_1 \frac{\partial^3 y}{\partial t^2 \partial x}\bigg|_{x=0} \tag{1}$$

for the hub,

$$\rho\frac{\partial^2 y}{\partial t^2} + EI\frac{\partial^4 y}{\partial x^4} = 0 \tag{2}$$

for the flexible beam,

$$T_2 + u_2 = -I_2\ddot{\theta}_2 = -I_2 \frac{\partial^3 y}{\partial t^2 \partial x}\bigg|_{x=L} \tag{3}$$

and

$$F_2 + F_3 = -m_2 \frac{\partial^2 y}{\partial t^2}\bigg|_{x=L} \tag{4}$$

for the wrist joint,

$$u_2 - F_3 h = I_3\ddot{\theta}_f \tag{5}$$

and

$$F_3 = m_3 \frac{d^2}{dt^2}(y(L,t) + h\theta_f) \tag{6}$$

for the rigid wrist, with the boundary conditions for the flexible beam written as:

$$y|_{x=0} = 0 \tag{7}$$

$$EI\frac{\partial^2 y}{\partial x^2}\bigg|_{x=0} = T_1 = I_1\frac{\partial^3 y}{\partial t^2 \partial x}\bigg|_{x=0} - u_1 \tag{8}$$

$$EI\frac{\partial^2 y}{\partial x^2}\bigg|_{x=L} = T_2 = -I_2\frac{\partial^3 y}{\partial t^2 \partial x}\bigg|_{x=L} - u_2 \tag{9}$$

$$EI\frac{\partial^3 y}{\partial x^3}\bigg|_{x=L} = -F_2\,. \tag{10}$$

Those equations of motion and boundary conditions can be rearranged to get

$$\rho\frac{\partial^2 y}{\partial t^2} + EI\frac{\partial^4 y}{\partial x^4} = 0 \tag{11}$$

$$y|_{x=0} = 0 \tag{12}$$

$$EI\frac{\partial^2 y}{\partial x^2}\bigg|_{x=0} = I_1\frac{\partial^3 y}{\partial t^2 \partial x}\bigg|_{x=0} - u_1 \tag{13}$$

$$EI\frac{\partial^2 y}{\partial x^2}\bigg|_{x=L} = -I_2\frac{\partial^3 y}{\partial t^2 \partial x}\bigg|_{x=L} - u_2 \tag{14}$$

$$\frac{EI\frac{\partial^3 y}{\partial x^3}\Big|_{x=L} = (m_2 + \frac{m_3 I_3}{I_3 + m_3 h^2})\frac{\partial^2 y}{\partial t^2}\Big|_{x=L} + m_3 h}{I_3 + m_3 h^2 u_2} \tag{15}$$

and

$$(I_3 + m_3 h^2)\ddot{\theta}_f = u_2 - m_3 h\frac{\partial^2 y}{\partial t^2}\bigg|_{x=L}\,. \tag{16}$$

Equations 1 to 15 are the boundary conditions (BC's) of the partial differential equation (PDE), Equation 11, which describes the dynamic motion of the flexible beam. Equation 16 describes the dynamics of the rigid wrist. The PDE and BC's of the flexible beam's motion are now decoupled from the rigid wrist's dynamics, and they can be solved separately. The load to the free end of the beam due to the motion of the wrist is equivalent to a point load with a mass of $(m_3 I_3)/(I_3 + m_3 h^2)$. The beam is also subjected to a lateral force equivalent to $(m_3 h u_2)/(I_3 + m_3 h^2)$ in addition to the torque u_2 applied at its free end.

The natural modal frequencies and shapes can be found by solving those equations with their input terms neglected. The introduction of the "separation of variables"

$$y(x,t) = \phi(x)T(t) \tag{17}$$

for each natural mode transforms the PDE into

$$\rho\ddot{T}\phi + EIT\phi^{IV} = 0 \, , \tag{18}$$

which can then be separated to form

$$\ddot{T} + \omega^2 T = 0 \tag{19}$$

and

$$\phi^{IV} - \frac{\rho\omega^2}{EI}\phi = 0 \tag{20}$$

for non-negative real ω's. With the definition of

$$\beta^4 = \frac{\rho\omega^2}{EI} \, , \tag{21}$$

the solution of the natural modes are harmonic motions in the time dependent part; and the modal shapes are sinusoidal and hyperbolic functions of the spatial variable, which can be expressed as:

$$\phi(x) = c_1 \sin(\beta x) + c_2 \cos(\beta x) + c_3 \sinh(\beta x) + c_4 \cosh(\beta x) \tag{22}$$

while the following four boundary conditions

$$\phi(0) = 0 \tag{23}$$

$$EI\phi''(0) = -I_1\omega^2\phi'(0) \tag{24}$$

$$EI\phi''(L) = I_2\omega^2\phi'(L) \tag{25}$$

$$EI\phi'''(L) = -(m_2 + \frac{m_3 I_3}{I_3 + m_3 h^2})\omega^2\phi(L) \tag{26}$$

are satisfied. The modal motion of the rigid wrist can then be derived from Equation 16 as:

$$\theta_f = \Theta_f T(t) \tag{27}$$

where its modal shape

$$\Theta_f = -\frac{m_3 h}{I_3 + m_3 h^2}\phi(L) \tag{28}$$

for $\omega > 0$.

With the ith oscillatory mode shape of the flexible beam expressed as $\phi_i(x)$ and the rigid mode shape as $\phi_0(x)$, the general motion of the wrist-beam system can be expressed in terms of n vibration modes and two rigid body modes as:

$$y(x,t) = \sum_{i=1}^{n} \phi_i(x)q_i(t) + \phi_0(x)q_0(t) = \sum_{i=0}^{n} \phi_i(x)q_i(t) \tag{29}$$

and

$$\theta_f(t) = \sum_{i=1}^{n} \Theta_{f_i}q_i(t) + \theta_{f_0}(t) = -\frac{m_3 h}{I_3 + m_3 h^2}\sum_{i=1}^{n} \phi_i(L)q_i(t) + \theta_{f_0}(t) \tag{30}$$

where $q_i(t)$ represents the ith modal variable, $q_0(t)$ represents the rigid body modal variable for the flexible beam, and $\theta_{f_0}(t)$ corresponds to the rigid body mode of the wrist.

In order to express the system dynamics as a linear differential equation in the matrix form

$$M\ddot{q} + Kq = Q \ , \tag{31}$$

the elements of the matrices M, K, and the column vector Q have to be derived from either analyses or tests performed on the system.

The elements M_{ij} of the mass matrix M can be obtained by first deriving the expression of the kinetic energy

$$
\begin{aligned}
T =& \frac{1}{2}\int_0^L \rho\frac{(\partial y^{2}}{\partial t)} dx + \frac{1}{2}I_1(\left.\frac{\partial^2 y}{\partial t\partial x}\right|_{x=0})^2 + \frac{1}{2}I_2(\left.\frac{\partial^2 y}{\partial t\partial x}\right|_{x=L})^2 \\
&+ \frac{1}{2}I_3\dot{\theta}_f^2 + \frac{1}{2}m_2(\left.\frac{\partial y}{\partial t}\right|_{x=L})^2 + \frac{1}{2}m_3(\left.\frac{\partial y}{\partial t}\right|_{x=L} + h\dot{\theta}_f)^2
\end{aligned}
\qquad (32)
$$

which can be expressed in terms of modal variables and modal shapes as

$$
\begin{aligned}
T =& \frac{1}{2}\int_0^L \rho(\sum_{i=0}^{n}\phi_i(x)\dot{q}_i(t))(\sum_{j=0}^{n}\phi_j(x)\dot{q}_j(t))dx \\
&+ \frac{1}{2}I_1(\sum_{i=0}^{n}\phi_i{}'(0)\dot{q}_i(t))(\sum_{j=0}^{n}\phi_j{}'(0)\dot{q}_j(t)) \\
&+ \frac{1}{2}I_2(\sum_{i=0}^{n}\phi_i{}'(L)\dot{q}_i(t))(\sum_{j=0}^{n}\phi_j{}'(L)\dot{q}_j(t)) \\
&+ \frac{1}{2}I_3\dot{\theta}_f^2 + \frac{1}{2}m_2(\sum_{i=0}^{n}\phi_i(L)\dot{q}_i(t))(\sum_{j=0}^{n}\phi_j(L)\dot{q}_j(t)) \\
&+ \frac{1}{2}m_3(\sum_{i=0}^{n}\phi_i(L)\dot{q}_i(t) + h\dot{\theta}_f)(\sum_{j=0}^{n}\phi_j(L)\dot{q}_j(t) + h\dot{\theta}_f)\ .
\end{aligned}
\qquad (33)
$$

The elements M_{ij} is twice the coefficient of the term q_iq_j in T, and can be expressed as:

$$
\begin{aligned}
M_{00} =& \int_0^L \rho(\phi_0(x))^2dx + I_1(\phi_0{}'(0))^2 + I_2(\phi_0{}'(L))^2 \\
&+ (m_2 + m_3)(\phi_0(L))^2
\end{aligned}
\qquad (34)
$$

$$
\begin{aligned}
M_{ii} =& \int_0^L \rho(\phi_i(x))^2dx + I_1(\phi_i{}'(0))^2 + I_2(\phi_i{}'(L))^2 \\
&+ (m_2 + \frac{m_3I_3}{I_3 + m_3h^2})(\phi_i(L))^2
\end{aligned}
\qquad (35)
$$

$$
M_{f0} = I_3 + m_3h^2 \qquad (36)
$$

$$
M_{0,f0} = M_{f0,0} = m_3h\phi_0(L)\ . \qquad (37)
$$

The mass for the rigid body mode of the flexible beam can be expressed as:

$$M_0 = M_{00} \ , \tag{38}$$

and for the ith flexible mode as:

$$M_i = M_{ii} \ . \tag{39}$$

The terms M_{ij} vanish for $i \neq j$ due to the orthogonality of modal shapes for different natural frequencies. M_{f0} is the mass of the rigid body mode associated with the rigid wrist, and the two terms $M_{0,f0}$, $M_{f0,0}$ represents the cross coupling mass between the two rigid body modes.

It is convenient to define the rigid body mode shape of the beam as

$$\phi_0(x) = x \tag{40}$$

so that

$$\phi_0{}'(x) = 1 \tag{41}$$

$$M_{0,f0} = M_{f0,0} = m_3 hL \tag{42}$$

and

$$M_0 = \int_0^L \rho x^2 dx + I_1 + I_2 + (m_2 + m_3)L^2 \ . \tag{43}$$

The matrix K will be diagonal with $K_0 = K_{f0} = 0$ due to the orthogonality of the modal shapes with distinct natural frequencies, and the spring stiffnesses vanish for the two rigid body modes.

The input vector Q of Equation 31 can be derived from the expression of the generalized force

$$\overline{\delta w} = u_1 \delta(\frac{\partial y}{\partial x}\bigg|_{x=0}) + u_2 \delta(\theta_f - \frac{\partial y}{\partial x}\bigg|_{x=L}) \tag{44}$$

which can also be written as:

$$\begin{aligned}\overline{\delta w} =& u_1 \sum_{i=0}^{n} \phi_i'(0)\delta q_i(t) + u_2((\frac{-m_3 h}{I_3 + m_3 h^2}) \sum_{i=1}^{n} \phi_i(L)\delta q_i(t) \\ &+ \delta\theta_{f_0} - \sum_{i=0}^{n} \phi_i'(L)\delta q_i(t)) \ .\end{aligned} \tag{45}$$

The elements Q_i of the input vector Q is defined by

$$\overline{\delta w} = \sum_{i=0}^{n} Q_i \delta q_i(t) + Q_{f0} + \delta\theta_{f_0} \tag{46}$$

and it can be seen from Equation 45 that

$$Q_0 = u_1 \phi_0'(0) - u_2 \phi_0'(L) \tag{47}$$

$$Q_i = u_1 \phi_i'(0) + u_2((\frac{-m_3 h}{I_3 + m_3 h^2})\phi_i(L) - \phi_i'(L)) \tag{48}$$

and

$$Q_{f0} = u_2 \tag{49}$$

With elements of M and Q determined previously, Equation 31 can be expressed as:

$$\begin{bmatrix} M_0 & M_{0,f0} \\ M_{f0,0} & M_{f0} \end{bmatrix} \begin{bmatrix} \ddot{q}_0 \\ \ddot{\theta}_{f_0} \end{bmatrix} = \begin{bmatrix} Q_0 \\ Q_{f0} \end{bmatrix} \tag{50}$$

for the two rigid body modes, and

$$M_i \ddot{q}_i + K_i q_i = Q_i \tag{51}$$

for all flexible modes. Values of K_i can be related to the modal frequency ω_i by

$$\frac{K_i}{M_i} = \omega_i^2 \ . \tag{52}$$

The sensor outputs of the system can now be expressed as:

$$\theta_h = \frac{\partial y}{\partial x}\bigg|_{x=0} = q_0(t) + \sum_{i=1}^{n} \phi_i'(0) q_i(t) \tag{53}$$

for the hub angle,

$$\begin{aligned} y_t(t) &= y(L,t) + l\theta_f(t) \\ &= Lq_0(t) + \frac{I_3 + m_3 h(h-l)}{I_3 + m_3 h^2} \sum_{i=1}^{n} \phi_i(L) q_i(t) + l\theta_{f_0}(t) \end{aligned} \tag{54}$$

for the end effector's position, and

$$\begin{aligned} \theta_w(t) &= \theta_f(t) - \frac{\partial y}{\partial x}\bigg|_{x=L} \\ &= \theta_{f_0}(t) - q_0(t) + \sum_{i=1}^{n}((-\frac{m_3 h}{I_3 + m_3 h^2})\phi_i(L) - \phi_i'(L)) q_i(t) \end{aligned} \tag{55}$$

for the wrist axis angle.

It is important to examine the transfer function from the wrist motor input u_2 to the end effector's position y_t, since the former provides the most direct control to the latter, which is exactly the most important variable to be controlled. Neglecting the hub motor input u_1 and initial conditions, the Laplace transformation applied to the previously derived equations of motion yields

$$\begin{aligned} \frac{Y_t(s)}{U_2(s)} =& \frac{L(m_3(l-h)(h+L) - I_3) + l(M_0 - m_3 L^2)}{s^2(M_0 I_3 + m_3 h^2(M_0 - m_3 L^2))} \\ &+ \frac{I_3 + m_3 h(h-l)}{I_3 + m_3 h^2} \sum_{i=1}^{n} \phi_i(L)(\phi_i(L)(-\frac{m_3 h}{I_3 + m_3 h^2}) - \phi_i'(L)) M_i(s^2 + \omega^2) \; . \end{aligned} \tag{56}$$

The coefficient of the term $\frac{1}{s^2}$ is usually positive (actually it is more desirable to design the mechanical system resulting in a positive term here), and so is the term $\phi_i(L)(\phi_i(L)(-\frac{m_3 h}{I_3 + m_3 h^2}) - \phi_i'(L))$, since the displacement at the end point of a flexible beam $\phi_i(L)$ has the same sign as the slop $\phi_i'(L)$ there for all modes, as can be seen in Fig. 3b.-3.

Equation 56 can be rewritten as:

$$\frac{Y_t(s)}{U_2(s)} = \frac{A_0{}^2}{s^2} - (I_3 + m_3 h(h-l)) \sum_{i=1}^{n} \frac{B_0{}^2}{(s^2 + \omega^2)} \; , \tag{57}$$

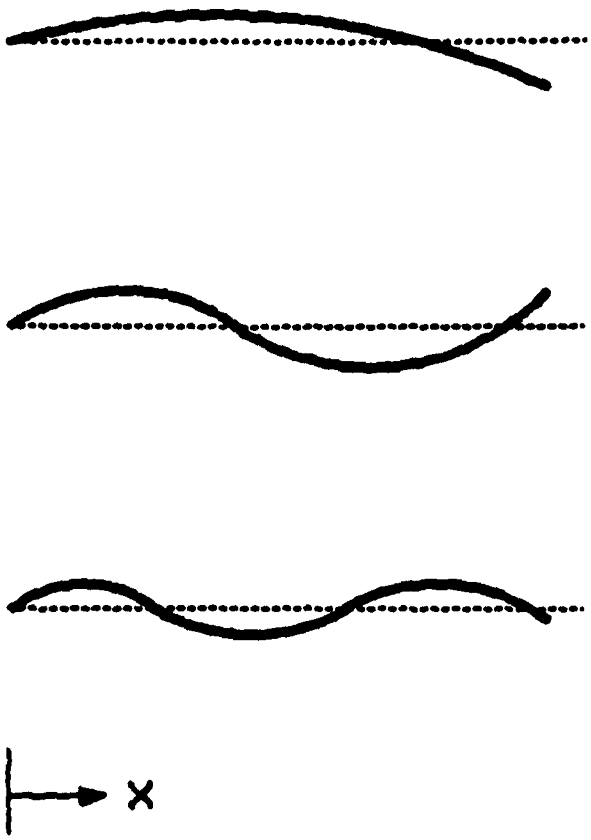


Fig. 3b.-3: The three mode shapes of a continuous flexible beam showing $\phi_i(L)\phi_i'(L) > 0$

and a critical length l^* can be defined such that the term $(I_3 + m_3h(h - l^*))$ vanishes.

Equation 57 has three different characteristics depending on the value of l relative to that of l^*:

1. if $l = l^*$, then

$$(I_3 + m_3h(h - l)) = 0 \tag{58}$$

which means

$$\frac{Y_t(s)}{U_2(s)} = \frac{{A_0}^2}{s^2}\ , \tag{59}$$

the motion of the end effector's position is decoupled from the structural flexibility of the main beam, and the motion is similar to that of a rigid wrist mounted on a fixed base. This condition is the easiest to control, but it requires the relation in Equation 58 to be satisfied exactly.

2. if $l > l^*$, then

$$(I_3 + m_3h(h - l)) < 0 \tag{60}$$

which means

$$\frac{Y_t(s)}{U_2(s)} = \frac{{A_0}^2}{s^2} + \sum_{i=1}^{n} \frac{{C_i}^2}{(s^2 + \omega^2)}\ , \tag{61}$$

and the motion of the end effector's position is coupled with the structural flexibility of the main beam in such a way that the motion is similar to that of a colocated sensor-actuator pair in a flexible structure system. The system can be stabilized easily with high robustness since the transfer function possesses alternating poles and zeros along the imaginary axis of the s-plane

3. if $l < l^*$, then

$$(I_3 + m_3h(h - l)) > 0 \tag{62}$$

which means

$$\frac{Y_t(s)}{U_2(s)} = \frac{{A_0}^2}{s^2} - \sum_{i=1}^{n} \frac{{C_i}^2}{(s^2 + \omega^2)}\ , \tag{63}$$

and the motion of the end effector's position is coupled with the structural flexibility of the main beam in an unfavorable way that there is no longer alternating pole-zero pattern on the imaginary axis in the s-plane. The system becomes difficult to stabilize since accurate knowledge about the system dynamics is required to design the feedback controller, and it is highly sensitive to parameter variations in general.

The physical meaning of l^*, which makes $(I_3 + m_3 h(h - l^*))$ vanish, can be explained with the free body diagram as shown in Fig. 3b.-4.

There is little torque generated by any of the two motors during the oscillatory motion at one of the natural frequencies; but there is a reciprocal reaction force transmitted between the main arm and the rigid wrist in the lateral direction through the wrist motor axis. If the load at the end effector is considered as part of the rigid wrist, then that reaction force is the only external force applied to the rigid wrist, and generates the oscillatory motion at one of the system's natural frequency. For small motions, there is a point on the rigid body, also known as the center of percussion, which remains motionless due to the cancellation of the rotational motion and the linear translation at that particular point. The distance between the motionless point and the wrist motor axis is l^* which satisfies the condition

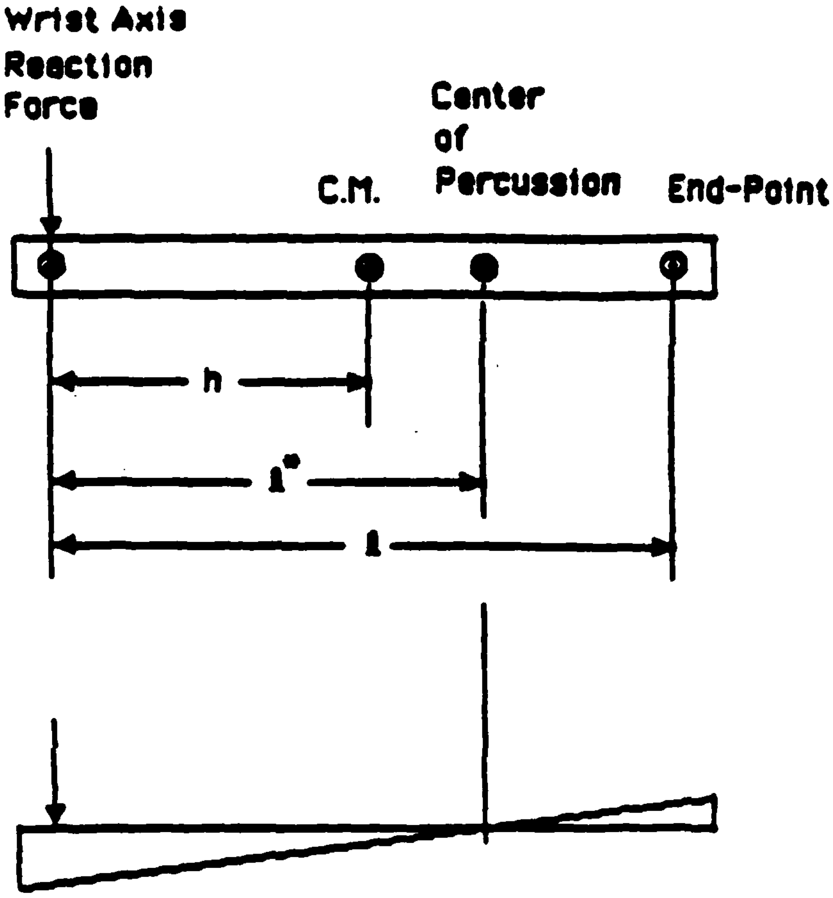
$$I_3 = m_3 h(l^* - h) \ . \tag{64}$$

The location of the center of percussion does not change with the magnitude of that reaction force, so that it is always motionless for any natural frequency the wrist-beam system is oscillating at. It is quite obvious that the natural modes are not observable by the displacement of the end effector if it is located at the center of percussion, i.e. if $l = l^*$.

If the end effector is located further away from the wrist motor axis, i.e. $l > l^*$, then its displacement will have the same sign as that of the wrist axis angle θ_w, and this results in an alternating pole-zero pattern in the transfer function $Y_l(s)/U_2(s)$ due to the fact that the wrist axis angle is really colocated with the actuator. If $l < l^*$, then all flexible modes will have coefficients with different sign from that of the rigid body mode expressed in $Y_l(s)/U_2(s)$.

Appendix E shows that the inequality condition as shown in Equation 58, 60, and 62 will not be changed by the mass of the load carried by the end effector, and only an increase in the moment of inertia of the load by the amount of $(m_3 h(l - h) - I_3)$ or more will change the relations from Equation 58 and 60 to Equation 62. We shall consider only the effect of the mass of the load in this analysis, since the moment of inertia of the load is associated with the rotation of the load which is not to be controlled by the one-dimensional wrist-beam system.

Eight transfer functions can be defined of which the most important five are list below as:



$$I_3 = m_3 h \left(\ell^* - h \right)$$

Fig. 3b.-4: The free body diagram of the rigid wrist.

$$\frac{Y_1(s)}{U_1(s)} = \frac{\Theta_h(s)}{U_1(s)} = \frac{60.554(s^2+2.63^2)(s^2+20.7^2)(s^2+62.8^2)}{s^2(s^2+11.24^2)(s^2+29.6^2)(s^2+66.7^2)} \,, \tag{65}$$

$$\frac{Y_2(s)}{U_2(s)} = \frac{\Theta_w(s)}{U_2(s)} = \frac{1009.2(s^2+9.82^2)(s^2+26.65^2)(s^2+58.04^2)}{s^2(s^2+11.24^2)(s^2+29.6^2)(s^2+66.7^2)} \,, \tag{66}$$

$$\frac{Y_4(s)}{U_1(s)} = \frac{Y_{bt}(s)}{U_1(s)} = \frac{-0.3355(s^2-12.75^2)(s^2-(54.7+j26.91)^2)(s^2-(54.7-j26.91)^2)}{s^2(s^2+11.24^2)(s^2+29.6^2)(s^2+66.7^2)} \,, \tag{67}$$

$$\frac{Y_2(s)}{U_1(s)} = \frac{\Theta_w(s)}{U_1(s)} = \frac{17.1262(s^2-8.88^2)(s^2-(44.82+j19.98)^2)(s^2-(44.82-j19.98)^2)}{s^2(s^2+11.24^2)(s^2+29.6^2)(s^2+66.7^2)} \,, \tag{68}$$

and

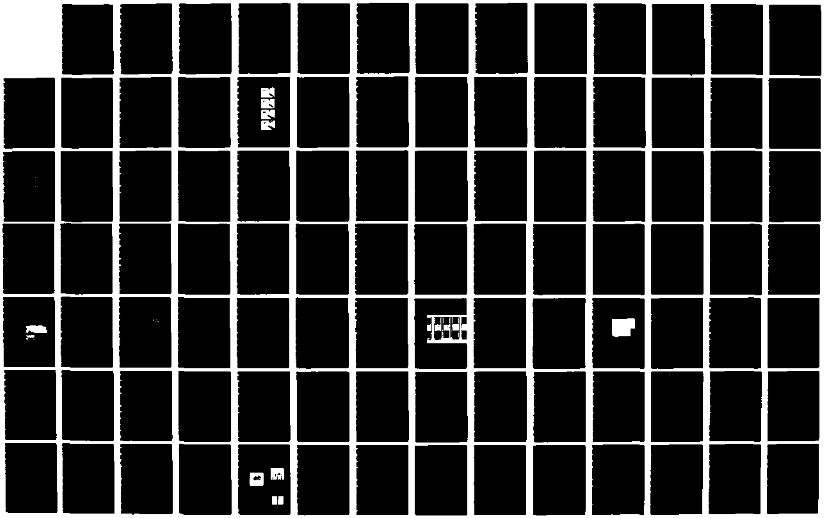
$$\frac{Y_3(s)}{U_2(s)} = \frac{Y_t(s)}{U_2(s)} = \frac{66.5299(s^2+10.82^2)(s^2+28.91^2)(s^2+64.95^2)}{s^2(s^2+11.24^2)(s^2+29.6^2)(s^2+66.7^2)} \,. \tag{69}$$

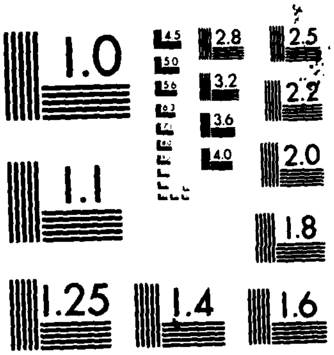
The pole-zero locations on the s-plane of the five transfer functions above are shown in Fig. 3b-5.

It is shown clearly that both $\Theta_h(s)/U_1(s)$ and $Theta_w(s)/U_2(s)$ have alternating.poles and zeros along the imaginary axis on the s-plane, since they both represent the transfer functions between colocated actuator-sensor pairs. Both $Y_{bt}(s)/U_1(s)$ and $\Theta_w(s)/U_1(s)$ have non-minimum phase zeros due to the separation of the sensors from actuators. The flexible beam tip position is better than the wrist axis angle when used to design the controller for the flexible main beam, since the former has higher frequencies for its transfer

AD-A172 266 CENTER FOR AUTOMATION AND MANUFACTURING SCIENCE(U) 2/3
STANFORD UNIV CA R H CANNON ET AL. SEP 86
SPO-13649-01-00 AFWAL-TR-86-4053 F49620-82-C-0092

UNCLASSIFIED F/G 13/9 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

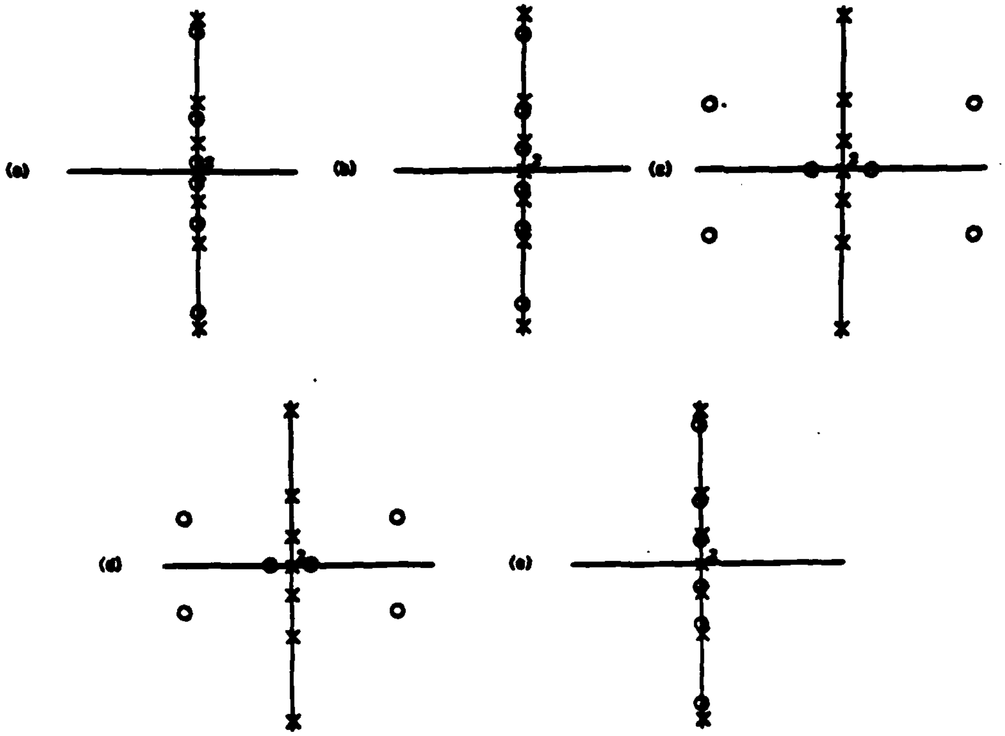


Fig. 3b.-5: Five of the eight transfer functions of the wrist-beam system for numerical analysis: (a)$\frac{Y_1(s)}{U_1(s)} = \frac{\Theta_A(s)}{U_1(s)}$, (b)$\frac{Y_2(s)}{U_2(s)} = \frac{\Theta_B(s)}{U_2(s)}$, (c)$\frac{Y_4(s)}{U_1(s)} = \frac{Y_{4c}(s)}{U_1(s)}$, (d)$\frac{Y_3(s)}{U_1(s)} = \frac{\Theta_B(s)}{U_1(s)}$, and (e)$\frac{Y_3(s)}{U_2(s)} = \frac{Y_1(s)}{U_2(s)}$.

zeros. $Y_t(s)/U_2(s)$ has alternating poles and zero since $l > l^*$ as the end effector is located at the end of the rigid wrist.

Dynamics of the Mechanical Experiment System

The mechanical experiment wrist-beam system was tested and analyzed to yield a set of linear equations of motion to describe its dynamics. One actuator at a time was driven with sinusoidal currents to measure the resonant frequencies and modal shapes. Values of coefficients related to the two rigid body modes were determined based on calculations, due to the difficulty in doing frequency response tests at DC frequency. Dampings were neglected and two natural modes along with the two rigid body modes were included in the system's model.

Current amplifiers are used to drive the two motors, so that the torques developed are proportional to the voltage inputs sent to the amplifiers. Including all actuators, sensors, amplifiers, and the mechanical system as the "plant", the plant's inputs and outputs are expressed in units of volts and it's dynamics are expressed in modal state form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\omega_1{}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_2{}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 \\ 0.5013 & -0.3133 \\ 0 & 0 \\ 1.6166 & 0 \\ 0 & 0 \\ 1.3075 & -0.2015 \\ 0 & 0 \\ -2.986 & 16.518 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{70}$$

and

$$\begin{bmatrix} \theta_h \\ v_{sg1} \\ v_{sg2} \\ v_{sg3} \\ v_{sg4} \\ \theta_w \\ y_t \end{bmatrix} = \begin{bmatrix} 3.301 & 0 & 10.644 & 0 & 8.6093 & 0 & 0 & 0 \\ 0 & 0 & -23.57 & 0 & -135.1 & 0 & 0 & 0 \\ 0 & 0 & 26.86 & 0 & 38.15 & 0 & 0 & 0 \\ 0 & 0 & -33.2 & 0 & 30.62 & 0 & 0 & 0 \\ 0 & 0 & 31.93 & 0 & -66.76 & 0 & 0 & 0 \\ -12.02 & 0 & 0 & 0 & -53.78 & 0 & 12.02 & 0 \\ 33.05 & 0 & 0.5187 & 0 & -0.3002 & 0 & 5.708 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \tag{71}$$

where

$$[\omega_1 = 12.318 rad/sec = 1.96 Hz] \tag{72}$$

and

$$[\omega_2 = 24.662 rad/sec = 3.925 Hz] \tag{73}$$

are the natural frequencies, v_1 and v_2 are the voltage inputs to the current amplifiers driving the hub and wrist motor respectively, and v_{sg1} to v_{sg4} are the four strain gauge outputs.

The wrist motor has little controllability over the first natural mode and the wrist axis angle has little observability for the same mode either, since the wrist motor is mounted on the end plate of the flexible beam and that end plate does not rotate with the side plates of the beam at the first mode. However, the second mode is still disturbed by the wrist motor.

Neglecting the flexible modes, the transfer functions from the wrist motor to the wrist tip position and to the wrist axis angle are:

$$\frac{Y_t(s)}{U_2(s)} \approx \frac{84}{s^2} \tag{74}$$

and

$$\frac{\Theta_w(s)}{U_2(s)} \approx \frac{202.31}{s^2} \,. \tag{75}$$

The equations of motion, Equation 70 and 71, are the best estimates from tests and calculations, but they still do not agree with some of the tests performed due to the following two difficulties in doing the system identification:

1. The system has two rigid body modes and two control inputs (actuators) and the system (wrist and beam) drifts away from the reference position in these two modes during tests for dynamic identification.
2. There is unmodeled nonlinearity of the system caused by the geometric second order effect and by friction at the wrist motor shaft. These nonlinearities introduce components of higher frequency harmonics when the system input is excited by a sinusoidal signal. These nonlinearities also affect the amplitude of the output signals making them not proportional to the input signals.

This poses a challenge to the control design, it must be as less sensitive to the plant variation as possible.

Controller Design Analysis And Evaluation

One of the goals of this research is to demonstrate several fast maneuvers of the minimanipulator with the wrist-beam mechanical system. The control system can be designed with different methods, resulting in different performances to command responses and different sensitivities to variation in system parameters. It is important to examine different controller designs, compare their ultimate performances and their sensitivities, so that the best one can be suggested and implemented for the demonstration.

A numerical analysis based on an ideal and linear wrist-beam model can serve that purpose most properly. A perfect knowledge about the system dynamics can be assumed during the analysis to compare the theoretical ultimate performances of different controller designs. Effects of regulators and state estimators can be separated due to the availability of all the system state information. System sensitivity can also be studied by varying the system parameters selectively.

Equations 65 to 69 are used as the "plant" for the numerical analysis described in this chapter. This model is assumed to have a heavy wrist which results in a large interaction between the fast wrist motion and the structural flexibility of the main beam. The large interaction represents the worst case and makes this analysis more general and covers more possible practical designs in the future.

Different types of regulators are compared first for their performances, since an ideal state estimator should be able to reconstruct all the system state variables (regardless of what states are actually to be fed back). Regulators designed with the modern control technique (LQG), with the pole-placement method, and with a direct end-point feedback method are compared.

Two regulator design methods are selected based on the comparison, and state estimators are designed for those two regulators accordingly. The performances of the estimators, and those of the overall control systems, are then compared. A control system design method is suggested as the result of the analysis.

Closed Loop Response

The dynamics of the whole system can be studied when the equations of motion of the plant, the regulator, and the estimator are combined for analysis. System sensitivity to plant variations can also be analyzed when some parameters are changed in the equations.

The dynamics of the regulator and that of the estimator are no longer separated when the plant varies, or when a reduced order estimator is used instead of a full order one. Step command responses are simulated. The system output errors due to disturbances are also studied.

The performances of the closed loop systems with different controller designs are compared and the best controller design is suggested.

LQG Regulator and Estimator

Figure 3b.-8 shows the system output responses to a step command and to impulsive disturbance inputs. There is not too much difference whether the flexible main beam's tip position, Fig. 3b.-8b, or the wrist axis angle, Fig. 3b.-8c, is used as sensors for the state estimation. However, the error of the system outputs due to disturbances are much larger than the errors of the state estimation as shown in Fig. 3b.-6 and Fig. 3b.-7.

Direct End-Point Feedback for the Wrist Motor Loop

The closed loop system responses will be different than those of the regulator and of the estimator separately, since a reduced plant model is used in the designs. Step command and impulsive disturbance responses are simulated and shown in Fig. 3b.-9.

The step command responses are similar to those of the full state regulator design, and the use of the reduced system dynamics in the controller design is thus justified. The wrist tip position response is slightly better when the flexible beam's tip position is used for the hub motor loop control than when the wrist motor axis angle is used. No estimation error is investigated here since an estimator based on a reduced system model cannot really estimate all state variables of the system. But the system output error due to some disturbances can still be calculated, and they are shown in Fig. 3b.-9c and d.

Comparison of Disturbance Responses

The system output responses to the step commands are similar for both the all-LQG design and for the direct end-point feedback designs, and the use of the state estimator does not degrade too much from the full state feedback designs. The system responses due to the impulsive disturbances have also been investigated, and a comparison between different controller designs reveals some important results.

The main purpose is to control the wrist tip position, and the simulation has shown that the disturbance from the wrist motor has larger effect than that from the hub motor with identical magnitude. The errors of the predicted wrist tip position based on the estimated states are compared and shown in Table 3b.-1 which indicates that the all-LQG design results in a better state estimation since the wrist tip position sensors are not used for state estimation in the direct end-point feedback controller. The errors of the actual wrist tip position are also compared in the same table, which shows a result opposite to that of the position estimation: and of course it is the performance of the actual wrist tip – not of the estimator – that is important. The direct end-point feedback controller results in less wrist tip position error because the wrist motor loop uses the actual wrist tip position and rate for feedback control instead of using the estimated state as in the all-LQG controller design.

The comparisons also show that there is not much difference between the sensor selections.

Variation of Plant Dynamics

Any change or unmodeled part of the plant dynamics will influence the system performance, and even destabilize the closed loop system. Parameters describing the system

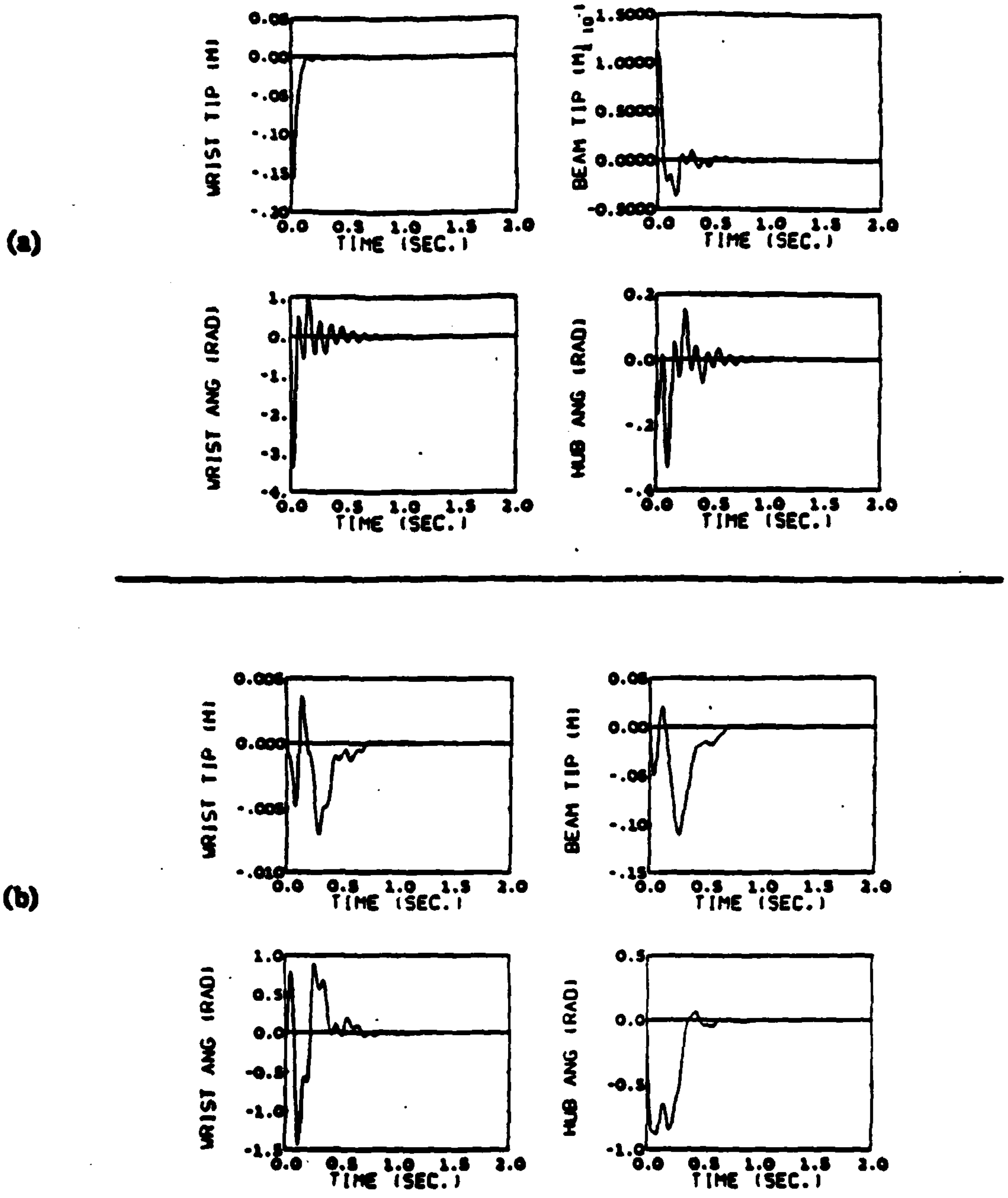


Fig. 3b.-6: The dynamic responses of the LQG estimator to impulsive disturbances from (a) the wrist motor and (b) the hub motor. The main beam's tip position and the other three sensors are used for the estimator feedback.

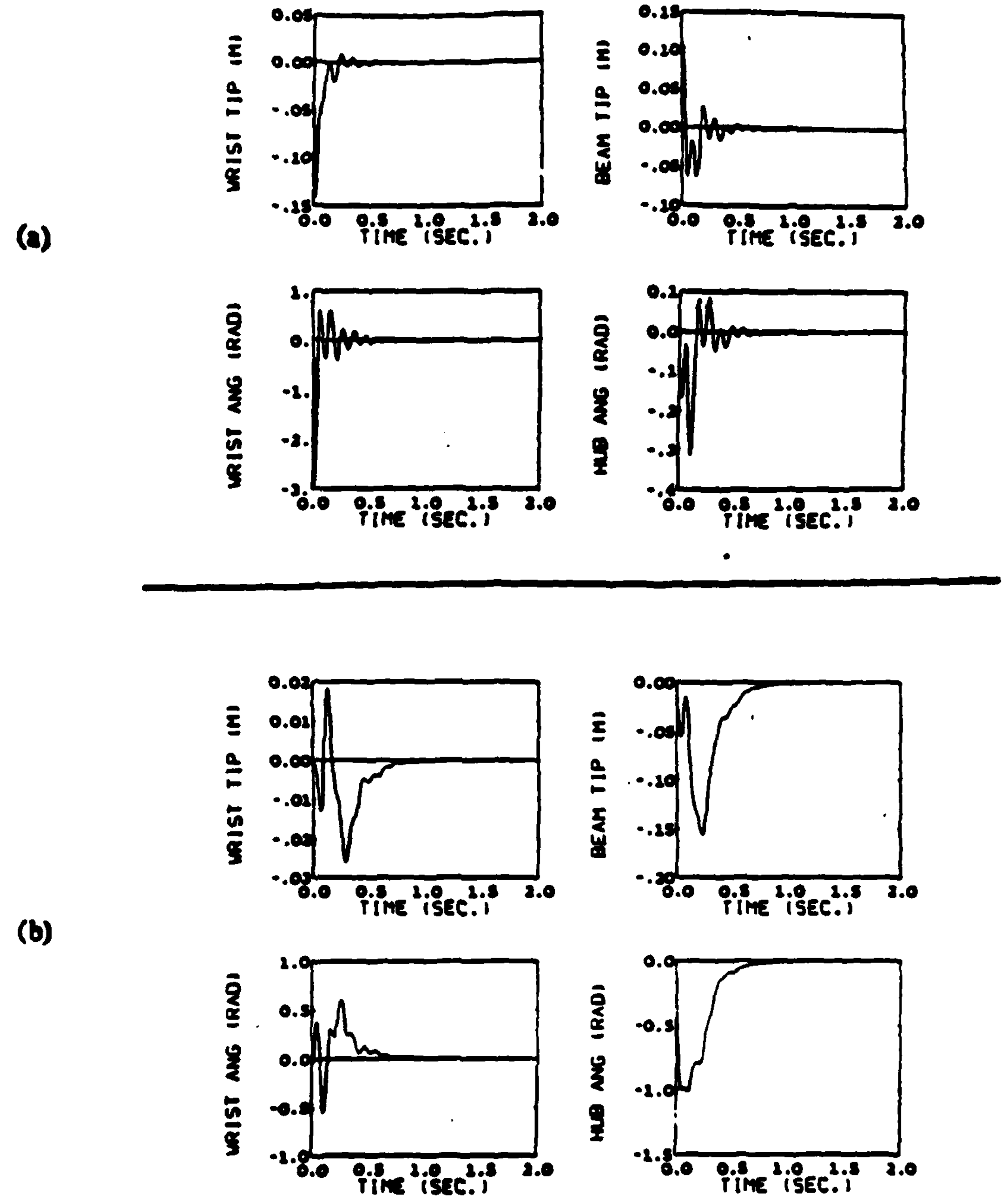


Fig. 3b.-7: The dynamic responses of the LQG estimator to impulsive disturbances from (a) the wrist motor and (b) the hub motor. The wrist axis angular position and the other three sensors are used for the estimator feedback.

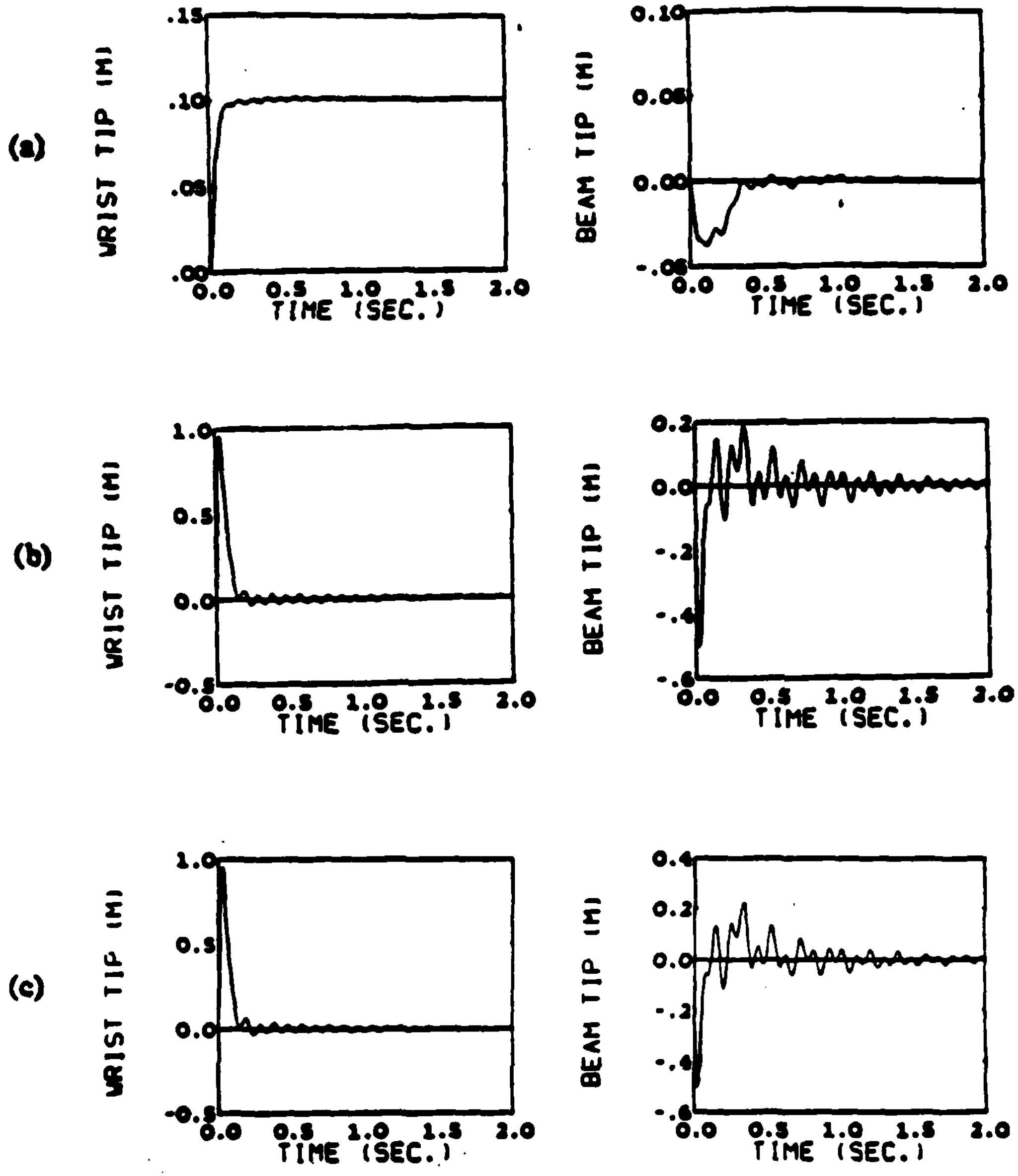


Fig. 3b.-8: The step command and impulsive disturbance response of the system with LQG regulator and estimator: (a) step command in wrist tip position; and impulsive wrist motor disturbances when (b) the flexible beam tip position, or (c) the wrist axis angle is used with other three sensors.

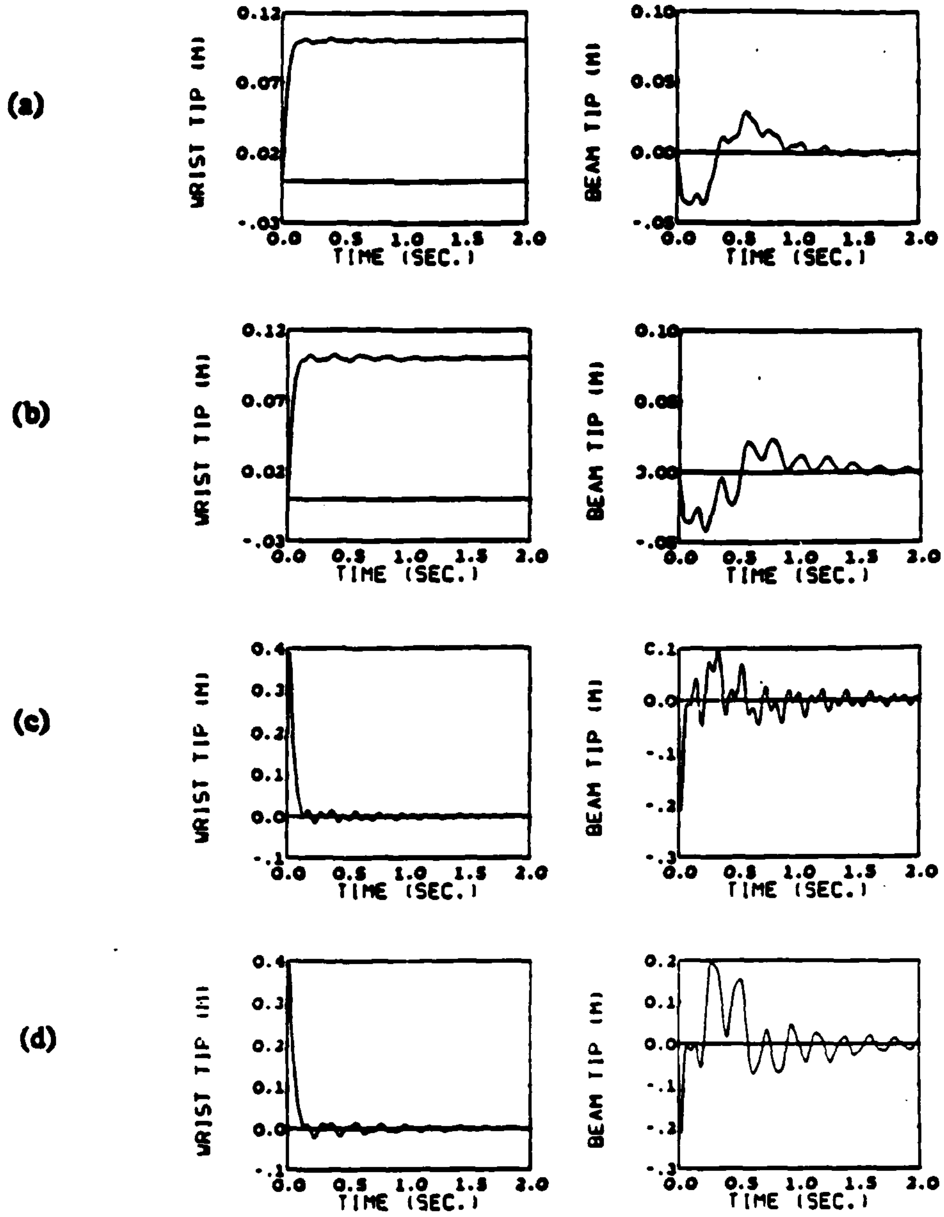


Fig. 3b.-9: **The step command ((a) and (b)), and impulsive disturbance ((c) and (d)) response of the system with reduced LQG regulator and estimator, when the wrist motor loop is closed with direct end-point feedback. ((a) and (c) use flexible beam's tip position and hub rate sensors, (b) and (d) use wrist axis angle and hub rate sensors.)**

Maximum Estimation Error of Wrist Tip Position

CTRL Sen.	All-LQG	Direct EPFB
Flex-Beam Tip Pos.	0.16 m	0.4 m
Wrist Angle	0.15 m	0.4 m

Maximum Response Error of Wrist Tip Position

CTRL Sen.	All-LQG	Direct EPFB
Flex-Beam Tip Pos.	0.95 m	0.4 m
Wrist Angle	0.95 m	0.4 m

Table 3b.-1: **Comparison of errors of the predicted wrist tip position, and of the actual wrist tip motion.**

dynamics can be varied arbitrarily to investigate their influences.

The structure stiffness of the flexible main beam is considered to be the most important parameter in this study, and analysis is performed to investigate the system sensitivity to this parameter. It is found lowering the stiffness by fifteen percent will start destabilizing the system with any of the controller design, and this high sensitivity is believed due to the estimator part of the controller. However, the direct end-point feedback control scheme can use, for the hub motor loop, slower controller which is less sensitive but does not degrade the wrist tip position response. Besides, it is much easier and more accurate to test the system and identiŕy its dynamics after the wrist motor loop is closed, if a high performance hub motor loop controller is required.

Conclusion from the Numerical Analysis

A direct end-point feedback controller for the wrist tip position control is recommanded based on the analysis above. It has the same nominal performance as obtained from other prominent controller designs, and it will not be degraded by parameter uncertainties or by the controller chosen for the flexible main beam. The controller design for the main beam can be based on a suitable compromise between performance and robustness, which allows a good practical design even in the absence of a very accurate model for the system dynamics.

Experimental Control Designs, Results And Demonstrations

The direct end-point feedback scheme is selected for the controller design of the mechanical experiment device, due to the analysis results from Controller Designs Analysis and Evaluation and the fact that the dynamic model described in Dynamics of the Mechanial System may not be very accurate.

The procedures for and the results from the feedback controller designs are described in this chapter. Maneuvers of the mechanical system are also recorded.

The Feedback Controller Designs

Several controller designs are described in this section. Different controllers are required for different operational modes. They also result in different performance, and require accordingly different accuracy in the modeling of the system dynamics.

Not all sensors considered in the earlier numerical analysis are implemented in the mechanical experimental system. A photo sensor is used to measure the wrist's tip position, but its velocity is not measured. A RVDT is installed at the wrist motor axis to measure its angle, but no position sensor is implemented to measure the tip position of the flexible main beam. The hub angle rate information is obtained by high passing the angular position sensed by a potentiometer. Four pairs of strain gauges are mounted along the sides of the flexible main arm. They are used to help identify the numerical values in the model of the system's flexibility, but are not used in the real time feedback control.

Wrist Motor Loop Closed with Wrist Tip Position Measurement

The wrist motor is controlled using feedback from the wrist tip position sensor when the wrist tip is in the vicinity of the target position.

Wrist Motor Loop Controller Design

Neglecting the flexible modes, the transfer function from the wrist motor to the wrist tip position can be approximated as

$$\frac{Y_t(s)}{U_2(s)} \approx \frac{84}{s^2} \tag{76}$$

as shown in Equation 74. It is discovered from many trials that a sample rate as fast as 100 Hz is needed to control this system in order to get desirable response. The transfer function expressed in the sampled time domain becomes

$$\frac{Y_t(z)}{U_2(z)} \approx \frac{0.0042(z+1)}{(z-1)^2} \tag{77}$$

and a lead compensator

$$\frac{U_2(z)}{Y_{te}(z)} = \frac{18(z-0.85)}{(z-0.55)} \tag{78}$$

where y_{te} is the wrist tip position error calculated from

$$y_{te} = y_{tc} - y_t \tag{79}$$

can be used to control the wrist tip's motion. The root locus of the closed loop pole locations versus the feedback gain is shown in Fig. 3b.-10.

The compensation as shown in Equation 78 results in the closed loop poles at $(0.856 \pm j0.274)$ and 0.75 on the z-plane, which are equivalent to $(-10.67 \pm j30.98)$ and -28.7 on the s-plane, or bandwidths at 5.21 Hz and 4.6 Hz respectively. The step response of the wrist tip position is shown in Fig. 3b.-11.

An overshoot is observed since the closed loop system damping is 0.33 only.

Hub Motor Colocated Feedback Control

The flexible main beam can be controlled by the hub motor with colocated sensor feedback when the dynamics of the flexible main beam are not known very well. Both the lead compensation

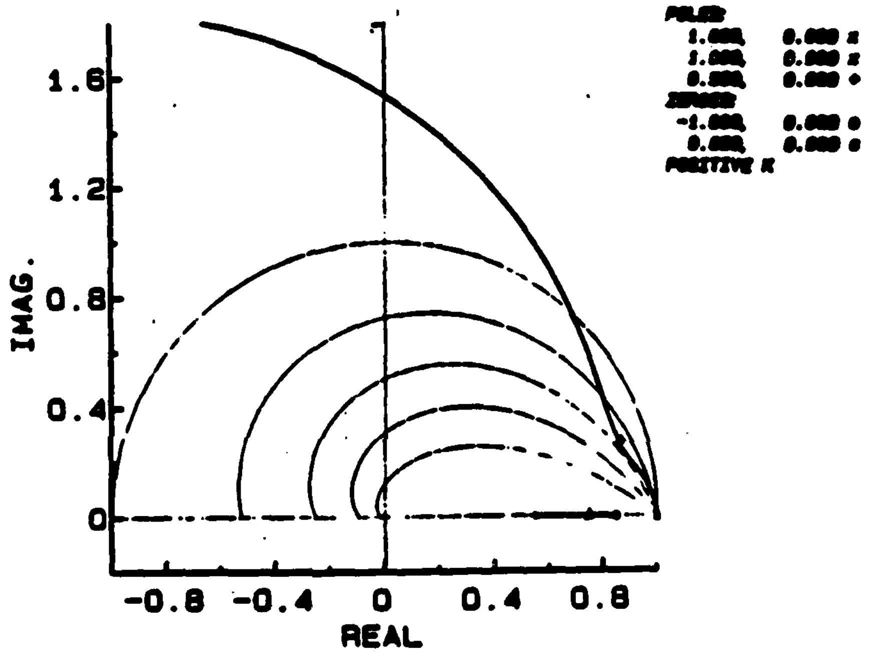


Fig. 3b.-10: **The root locus of the closed loop pole location versus the loop gain, for the wrist motor control loop using the wrist tip position sensor with lead compensator.**

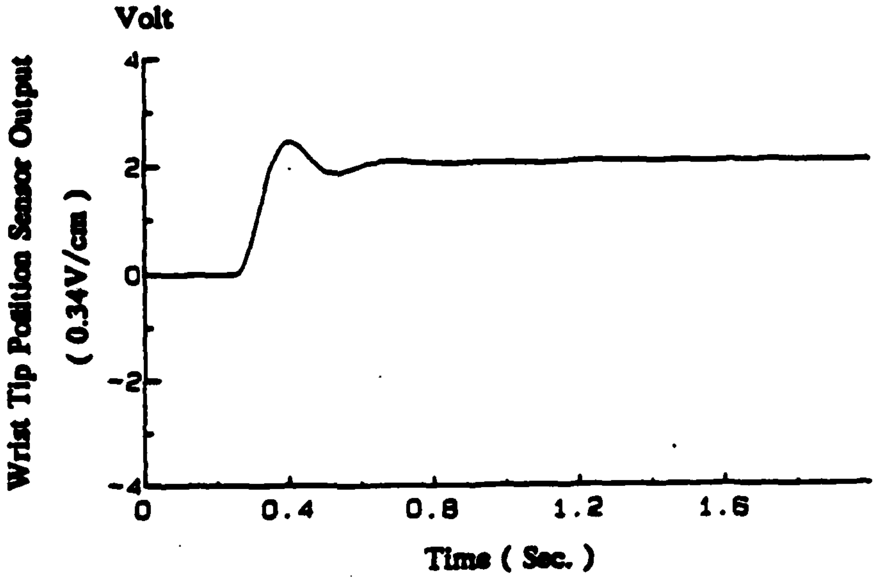


Fig. 3b.-11: The recorded step response of the wrist-beam system under the control of the wrist motor with a lead compensator using the wrist tip position sensor. No control is applied to the hub motor.

$$\frac{U_1(z)}{\Theta_{he}(z)} = \frac{35(z-0.96)}{(z+0.25)} \tag{80}$$

running at 100 Hz where θ_{he} is the hub angle error calculated from

$$\theta_{he} = \theta_{hc} - \theta_h \; , \tag{81}$$

and the position-plus-rate feedback

$$\begin{bmatrix} K_p & K_v \end{bmatrix} = [\, 1.26 \quad 1.26 \,] \tag{82}$$

are obtained from several trials. These result in a slow but robust controller for the flexible main beam. The step response is shown in Figs. 3b.-12 and 13.

The flexible main beam is given a command to follow the wrist tip and keep the wrist-beam system aligned at all times. Fig. 3b.-12a shows the wrist tip position being identical to that shown in Fig. 3b.-11; and this test demonstrates that the wrist tip response is independent of the motion of the flexible main beam. Fig. 3b.-12b shows the hub motion of the flexible main beam, when the hub motor is controlled with a lead compensator. The lead compensator results in an underdamped system, and the hub overshoots in the step motion. Fig. 3b.-12c shows the hub motion when the hub motor is controlled with the position-plus-rate feedback. The overshoot in the hub angle is eliminated due to the hub rate feedback, even though the rate information is derived from the hub position signal. The second structural mode is excited by the wrist motor during the step command, and Fig. 3b.-12b and 12c show the appearance of that mode in the motion of the main beam.

Fast Controller Design for the Flexible Main Beam

A fast controller for the flexible main beam can be designed based on the dynamic model identified in the last section. An LQG design algorithm is used to design the regulator and estimator.

The regulator is designed in the continuous time domain with the penalty weightings

$$R = 6.051 \tag{83}$$

for the hub motor control input, and

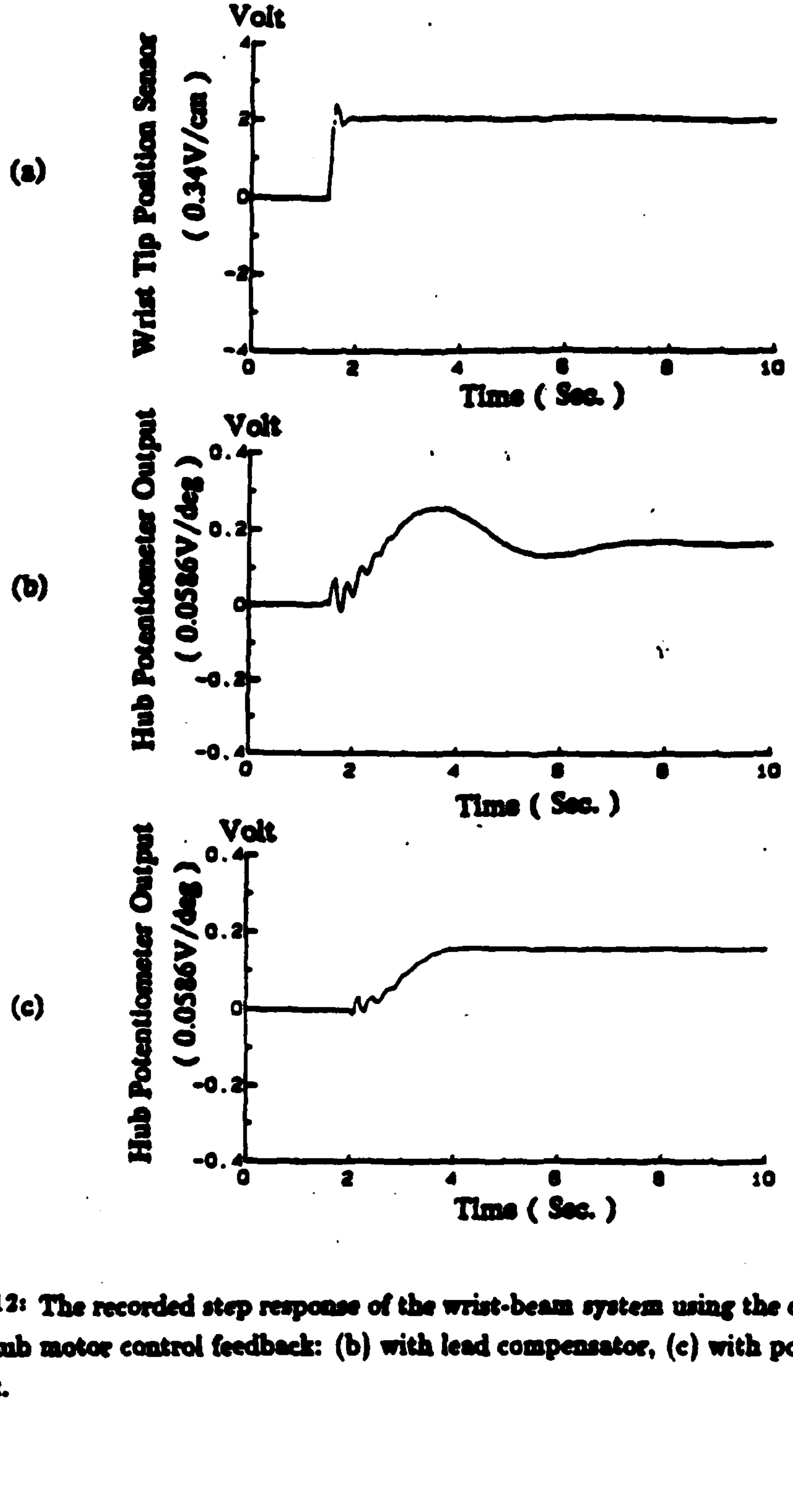


Fig. 3b.-12: The recorded step response of the wrist-beam system using the colocated sensor for the hub motor control feedback: (b) with lead compensator, (c) with position-plus-rate feedback.

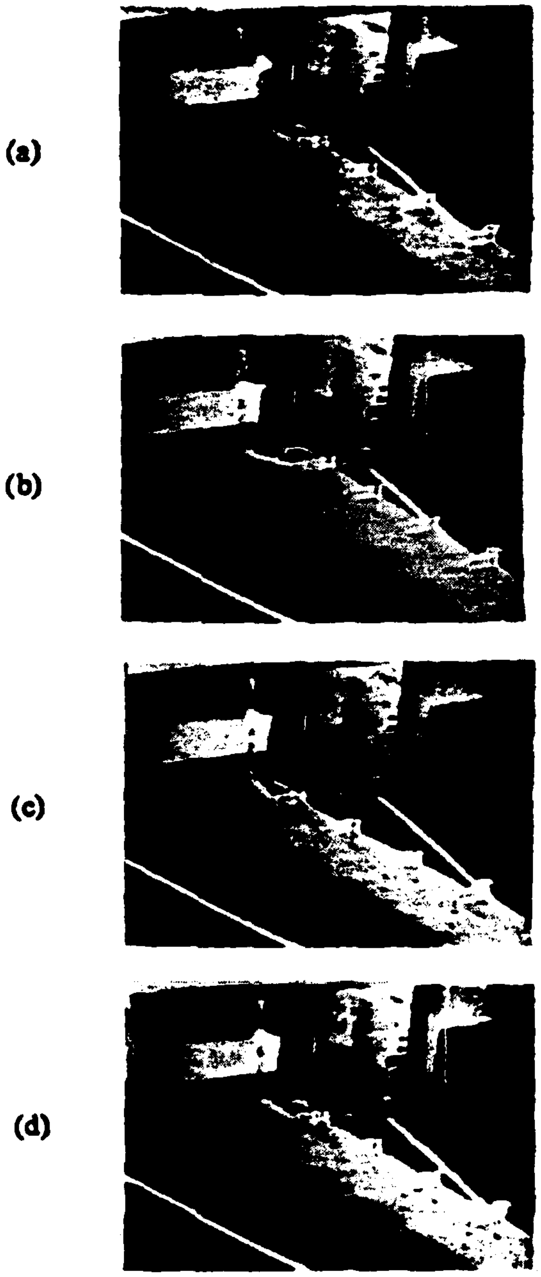


Fig. 3b.-13: Photograph sequence of the step response of the wrist-beam system using the wrist tip position sensor and a lead compensator for the wrist motor feedback control.

$$Q = \begin{bmatrix} 128480 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1284.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 500 \end{bmatrix} \tag{84}$$

for the six state variables. The wrist motor axis angle has a very small observability for the first flexible mode, which will result in a regulator leaving that mode uncontrolled if only the wrist motor axis is penalized in the regulator design. It turns out that penalizing errors of all the modes produce a better regulator in this particular case.

The regulator feedback gain obtained is

$$K = [145.72 \quad 41.253 \quad 110.82 \quad 7.5273 \quad 222.74 \quad 3.6613] \tag{85}$$

which results in the closed loop poles at

$$\begin{gathered} -4.677 \pm j23.25 \\ -9.889 \pm j9.705 \\ -5.465 \pm j6.083 \end{gathered}$$

and a corresponding bandwidth of 1.3 Hz, which should be compared with the beam's first cantilever natural frequency of 0.55 Hz! The continuous regulator feedback gain K can be used directly in a digital implementation without causing too much error as long as the sample rate is not too slow.

The estimator is designed in the discrete time domain since the system dynamics and the sensor feedback gains have to be transformed into the discrete-time format when they are implemented into a digital estimator. The computation time required for the state estimation limits the sample rate to 25 Hz, so that the the implementation of the feedback control for the whole system is running at multiple clock rates - 25 Hz for the estimator and 100 Hz for the wrist loop.

The wrist motor axis angle and the hub angular rate are used as the sensor outputs for the estimator design. The hub rate information is actually derived by sending the hub angular position measurement through a high-pass analog filter. The system dynamics of the plant is therefore augmented to include that filter in order to reduce the estimation error.

The system dynamics can be expressed in the discrete time domain as

$$x_{n+1} = \Phi x_n + \Gamma u_n \tag{86}$$

and

$$y_n = Hx_n \tag{87}$$

where

$$\Phi = \begin{bmatrix} 1 & 0.04 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .8787 & .0384 & 0 & 0 & 0 \\ 0 & 0 & -5.938 & .8787 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .559 & .0339 & 0 \\ 0 & 0 & 0 & 0 & -20.26 & .559 & 0 \\ -190.6 & -5.263 & -571.1 & -16.433 & -367.2 & -12.07 & 0.081 \end{bmatrix},$$

$$H = \begin{bmatrix} -82.52 & 0 & -4.722 & 0 & -55.307 & 0 & 0 \\ -207.4 & 0 & -665.2 & 0 & -528.1 & 0 & -1 \end{bmatrix}, \tag{88}$$

and

$$\Gamma = \begin{bmatrix} 4.29 \times 10^{-4} \\ 2.14 \times 10^{-2} \\ 1.35 \times 10^{-3} \\ 6.60 \times 10^{-2} \\ 1.01 \times 10^{-3} \\ 4.63 \times 10^{-2} \\ -7.61 \times 10^{-1} \end{bmatrix}. \tag{89}$$

Choosing the noise covariance values

$$Q_E = 1.6525 \times 10^5 \tag{90}$$

for the process noise coming from the hub motor, and

$$R_E = \begin{bmatrix} 1.325 \times 10^5 & 0 \\ 0 & 5.4483 \times 10^6 \end{bmatrix} \tag{91}$$

for the two sensor outputs, the estimator feedback gain matrix

$$L = \begin{bmatrix} -3.516 \times 10^{-3} & -5.5 \times 10^{-6} \\ -1.778 \times 10^{-2} & -3.093 \times 10^{-3} \\ -7.18 \times 10^{-4} & -5.52 \times 10^{-4} \\ 5.804 \times 10^{-3} & -2.788 \times 10^{-2} \\ -8.744 \times 10^{-4} & -3.42 \times 10^{-4} \\ -5.698 \times 10^{-3} & -1.741 \times 10^{-2} \\ 1.579 & 1.516 \times 10^{-1} \end{bmatrix} \tag{92}$$

is obtained with corresponding closed loop poles

$$\begin{array}{l} 0.5528 \pm j0.7131 \\ 0.692 \pm j0.3417 \\ 0.826 \pm j0.2035 \\ 0.0911 \end{array} \tag{93}$$

on the z-plane, or equivalent pole locations

$$\begin{array}{l} -2.572 \pm j22.78 \\ -6.476 \pm j11.46 \\ -4.039 \pm j6.038 \\ -59.90 \end{array} \tag{94}$$

on the s-plane. This estimator has a bandwidth of 1.16 Hz.

The motion of the flexible main beam becomes faster when the LQG optimum controller is used. A step response is recorded in Fig. 3b.-14. The hub moves back and forth in order to provide the optimum motion control, as revealed by Schmitz. The speed of the motion of the flexible main beam is increased by two to four times, as compared to those shown in Fig. 3b-12. Fig. 3b.-14 also shows an improvement of the wrist tip's motion.

Plant Dynamics Switching

The system dynamics, with the wrist loop closed, changes as the wrist motor loop controller switches when the tip position approaches or leaves the target position. The estimated state variables must be initialized to reduce the transient effect.

It is assumed that the true state and outputs of the system do not change abruptly when the wrist control mode switches. The four pairs of strain gauges provide mode shape information about the system which is shown in the output matrices H. They can be used to calculate the transition matrix for the state variable initializations.

Conclusions and Suggestions for Further Research

The research reported here is a theoretical and experimental investigation of a robotic concept, conceived by the author, where a quick-acting wrist is carried by a large, very flexible manipulator arm. Making the arm very flexible enable us to study the structural flexibility easily, and investigate controllers with bandwidth higher than the structural frequency. Once we have learned how to control a robotic system faster than its natural modes, we will be able to use robots that are less rigid in structure, and therefore lighter, to achieve higher accuracy and faster motion with less power consumption at a lower cost.

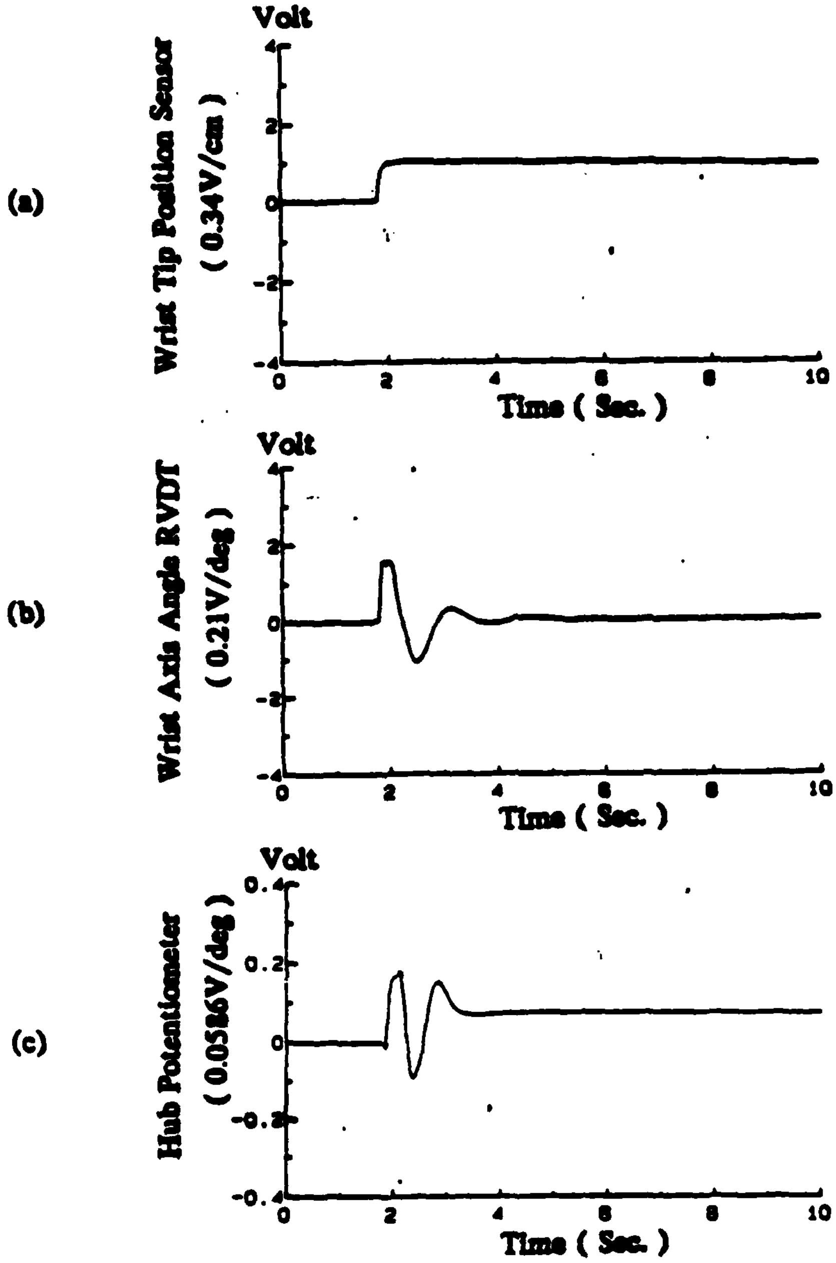


Fig. 3b.-14: The step response of the wrist-beam system with position-plus-rate feedback from the wrist tip to the wrist motor.

Conclusions

The following conclusions can be drawn from the research described in this report:

1. Both the theoretical analysis and maneuver demonstrations using a one-dimensional mechanical system show the feasibility to control a mini-manipulator at a bandwidth several times higher than the structural flexibility of the main manipulator, with the use of end-point sensors.

2. It has been discovered and demonstrated that the location of the end effector relative to that of the center-of-percussion of the minimanipulator is important. A proper mechanical design can avoid non-minimum phase zeros, so that a robust, high-bandwidth controller can be implemented.

3. A comparison of different controller design methods in the numerical analysis reveals that :

   a. Pole-placement control design results in the best response only if the system is perfectly linear and an exact knowledge about the system dynamics is available. Otherwise, its performance is no better than the optimal control (LQG) design.

   b. Optimal control (LQG) design produces good response, and is recommended if truly good knowledge about the system dynamics is available. However, its performance and stability are quite sensitive to plant variations or modeling errors.

   c. Direct end-point feedback control for the minimanipulator has tip-position responses similar to those of the LQG design. It is much more desirable, however, because (a) it is much less sensitive to incorrect knowledge about plant parameters, (b) its response does not degrade when a low performance controller is chosen for the main manipulator, and (c) it is much easier to test the system dynamics with the inner loop closed if a high performance controller for the main manipulator is desired. This method is especially recommended when uncertainty exists about the plant's dynamics. The plant dynamics changes with the inner loop control mode, and the controller for the main manipulator has to be modified accordingly.

4. The importance of using end-point feedback is demonstrated. It will be impossible to achieve high performance under modeling uncertainty, without the proper usage of at least one end-point sensor.

TASK 3c.
STRATEGIES FOR TASK COMMAND AND CONTROL OF TWO-LINK MANIPULATORS

Two separate, rather extensive contributions were made to this area during the year by Bruce Gardner and Avi Weinreb. Both were sponsored by this AFOSR Contract, and both culminated this year in Ph.D. theses which are available as SUDDAR reports, [Refs. 6 and 7].

Gardner's early results were reported in [Ref. 2, pp.41-46]. An update summary is reported here:

Strategies for Moving Target Capture

An analytic design study is conducted to demonstrate circumstances under which the inclusion of feedforward compensation in a target-tracking control scheme can be expected to offer significant performance gain (e.g., enough to justify cost implementation). In particular, a target-tracking controller design problem for a mechanical arm is developed to assess quantitatively the capacity of feedforward to provide a quicker, more accurate tracking response over wide ranges of uncertainty or variability in the dynamic parameters of *both* plant and target.

The Stanford Aeronautics and Astronautics Department Robotics Lab two-link, two-actuator mechanical arm, inherently a system with variable kinematic and dynamic parameters, provides an appropriate framework for this study. Using recent developments in the theory of quadratic synthesis of robust, low-order "optimal" controllers, control logic is developed — both with and without feedforward — that enables the arm end-point to track a physical target characterized in part by periodic motion of variable or uncertain frequency and phase.

It is shown that, using relatively noise-free measurements of target position coordinates only, feedforward compensation can be expected to provide substantial reductions in tracking errors for given constraints on control effort, particularly when the range of variation in target frequency is large. As noise levels in the position measurements increase, the relative improvement in tracking accuracy (for a given level of control effort) offered by feedforward decreases. However, if target rate coordinates are also measured and used in the feedforward control scheme, the improvement is shown to be considerable even for fairly high noise levels in target measurements.

Experimental verification of the predicted results contained herein is scheduled for the near future.

Strategies for Optimal Control with Multiple Bounded Inputs

Several aspects of optimal control of systems with multiple bounded inputs are considered.

A connection between the structure of multi-input systems and the existence of minimum time solutions with singular controls is identified. It is shown that systems which are decoupled or one-way coupled may have (depending on the boundary conditions specified) some singular, non-unique control components as part of the minimum-time solutions.

A new algorithm, which extends the basic gradient algorithm to optimal control problems with bounded controls is presented. When necessary, the resulting controls asymptotically use all the available control range, without violating the control bounds. The algorithm uses an adjustable control weight to enforce the control bounds.

Using the new algorithm, the minimum-time control problem of a two-link robot arm is solved, using an exact rigid body model and bounded controls. The problem is solved both for the given end-points case and for the new given-distance/unknown end-points case. The solutions provide new insight into the dynamic characteristics of the robot arm, pertaining both to path planning and design specifications.

TASK 3d.
ADAPTIVE CONTROL

Rationale

The achievement of high-performance control for complex, flexible mechanisms such as robot arms depends on the control computer having good quantitative understanding, i.e., a good mathematical dynamic model for the system to be controlled. In a robot there is the central difficulty that various important parameters influencing system dynamics will change during the course of system operation. An obvious example is a robot arm picking up a payload: the weight of the payload can significantly alter the dynamics of the arm. To achieve the best possible performance from the control system for such a mechanism, it is necessary to identify changes in significant system parameters while the system is in operation. The identified changes can then be used to update the controller design, again while the system is in operation, in order to optimize performance. This is the essence of adaptive control. We have undertaken the development and testing of such control systems for physical systems available in the manipulator research laboratory.

Description of First Experimental Mechanical System

As a first step, we have developed algorithms for adaptive control of a physical system previously built for the purpose of investigating problems in the control of flexible structures with non-colocated sensors and actuators. It is an experimental system designed to test most severely any adaptive control scheme. This system consists of four steel disks connected to a central torsion spring (see Fig. 3d.- 1). RVDT sensors measure the angular displacement of each disk, and a brushless DC torque motor mounted on the second disk from the top provides the control input. The polar moment of inertia of the top disk can be abruptly changed while the system is being controlled. This is accomplished by making the disk in pieces, as a series of concentric rings. Lifting off rings in succession causes significant and profound changes to occur abruptly in the dynamics of the system, providing a severe test for any adaptive controller.

Moreover, experience has shown that lightly-damped mechanical systems pose a difficult problem even for offline techniques — let alone online ones — when small measurement noise exists. These problems include excessive computation, parameter bias (a euphemism) and convergence to local minima.

Three Approaches

We have carried forward three different approaches to achieving experimental adaptive control of the challenging mechanical laboratory system that we built for this purpose.

The adaptive control system developed first, historically—a natural frequency-detection based approach by Wen Wie Chiang using a using a frequency-locked loop—is reported last in this section, as Approach 3. This work is now complete, and has been reported in the literature [Chi T]. Two more recent projects (Approaches 1 and 2) are based on making

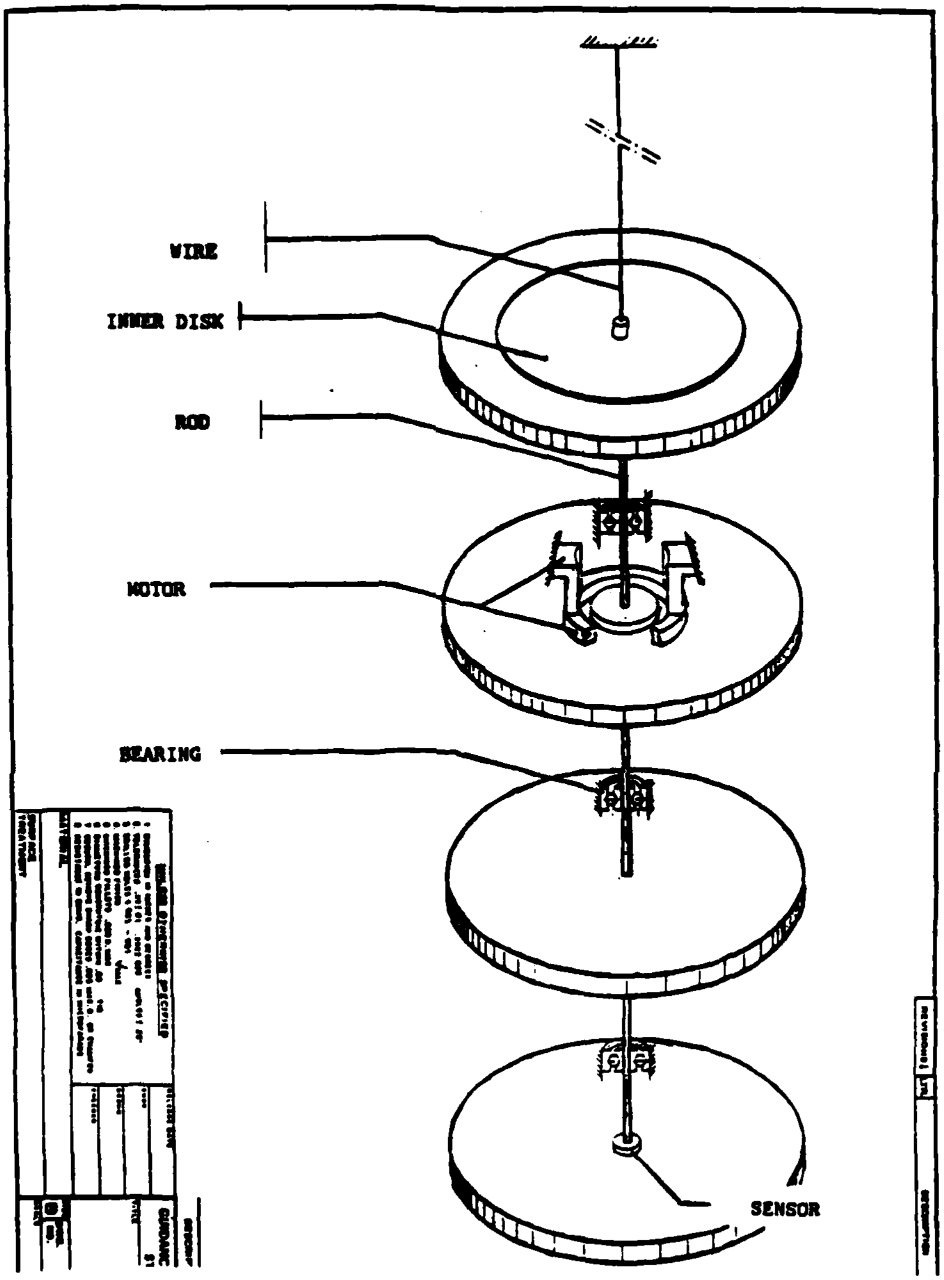


Fig. 3d.-1 Laboratory Four-Disk System

The system consists of four steel disks connected by a long steel rod. There is a brushless DC motor on the third disk, and an angular sensor on the lowest disk. The system has three vibration modes and a rigid body mode.

a polynomial best-fit to the system's transfer function in real time. These are reported first. They both are still under way, and provide a head start into new research under a new AFOSR-supported program at Stanford.

In Approaches 1 and 2, the basic algorithm form we have studied is the Self-Tuning-Regulator, which consists of an on-line identification scheme for measuring system parameters and a pole-placement algorithm for modifying the controller based on the identified parameters. We have taken two approaches to the identification problem. In the one approach, (Approach 2) the transfer function of the system is regarded as (almost) completely unknown, a priori. A recursive least-squares calculation is used to fit a system transfer function to data measured during controller operation. The other approach (Approach 1) regards the system transfer function as being completely known except for the inertias of the two aforementioned discs, which are regarded as being free to take on a continuum of values. The latter technique may have advantages in robustness and speed of convergence, since it involves less uncertainty about the physical system. However, the identification algorithm involves an algebraically non-linear gradient search, which is more computationally intensive than a linear least-squares fit.

In all three approaches the primary goal has been to demonstrate an adaptive control system that works on a mechanical, highly challenging experimental system, namely the four disk system of Fig. 3d.-1 especially constructed by Rosenthal [Ros T]. This system was chosen for two reasons, first because it is highly linear and very well modeled analytically, and second because it is designed to enable abrupt large changes in a critical system parameter to be made readily.

Rosenthal performed experiments on this system to demonstrate the performance of robust controllers designed using parameter optimization techniques. His results showed that the best possible robust (but non-adaptive) contollers that could stabilize the plant for all available values of the adjustable parameter had, in general, poor performance, in that their closed-loop damping was just as low as for the uncontrolled system (about 0.1% in our experimental apparatus). *The objective of applying adaptive control techniques to this system is to demonstrate performance much superior to that achievable with the best robust controller, and more broadly to gain generic experience in applying adaptive controllers to mechanical systems with structural flexibility.*

Design Criteria

There are two major demands made on an adaptive controller by the four disk system. The first is the requirement that the controller be able to adapt to a step change in the inertia of one of the system disks. An adaptive controller that is able only to adapt to gradually varying system parameters will not be satisfactory. The second major demand is closely related to the first. The locations of a pole and zero on the imaginary axis of the system's s-plane diagram 'flip' when the inertia of the top disc is changed. Any control law giving good performance is almost sure to be destabilised by this phenomenon, with a relatively short time constant. Hence the adaptive controller must be able to respond *quickly* to parameter changes.

The self-tuning regulator type of adaptive controller was chosen for implementation, as it was hoped that it would be able to meet these demands on its performance. Such a controller performs two distinct tasks. First, it implements an algorithm to identify the plant transfer function. Second, it calculates a control law to locate the closed-loop poles of the system in pre-specified locations using the identified plant transfer function.

APPROACH 1: Reduced-Order Identification Algorithm

The controller implemented here differs from those described in the literature. Most notably, a technique for using the highly structured nature of the plant uncertainty to speed up the identification process was devised.

The purpose of the identification algorithm is to calculate a transfer function to fit the measured values of plant input and output. The pole placement algorithm (that is, the control law calculation) requires that the *open loop* plant transfer function be calculated. This is what the following algorithm does.

As noted, it is critical that the identification algorithm be speedy for the four disk system. However, the four disk system is of eighth order, so that the transfer function contains sixteen unknowns, the eight coefficients of the numerator and denominator polynomials. In the absence of noise then, at least sixteen pieces of data are required after a step change in a system parameter has occurred before a new transfer function can be identified, *if the system is treated as wholly unknown.* In the case of the four disk system, however, only one parameter is subject to change. This points up the possibility of making use of knowledge of the plant to be controlled to speed identification. The problem is to make use of this knowledge in a way that can be implemented in real time.

Description

The basis of the algorithm is one of the standard techniques of linear, discrete-time identification, which is based on the following fact: The coefficients of the transfer function polynomials, $b(z)$ in the numerator and $a(z)$ in the denominator, can be used directly to form a difference equation satisfied by the samples of the plant input and output. That is, if the plant transfer function is modeled as

$$H(z) = \frac{b_1 z^{n-1} + \cdots + b_n}{z^n + a_1 z^{n-1} + \cdots + a_n} \tag{1}$$

then we can immediately write:

$$y(k) = -a_1 y(k-1) - \cdots - a_n y(k-n) + b_1 u(k-1) + \cdots + b_n u(k-n). \tag{2}$$

as the difference equation corresponding to this transfer function, where $y(k)$ and $u(k)$ are the plant output and input at the kth time step. For identification purposes, the model difference equation is used to predict the value of $y(k)$ given past values of plant outputs and inputs. The presence of disturbances and unmodeled dynamics causes the predicted value to differ from the actual measured value, and an error is formed which indicates how well a given guess of the plant transfer function matches the true system behavior. The error is defined by:

$$e(k) = y(k) - \{-a_1y(k-1) - \cdots - a_ny(k-n) + b_1u(k-1) + \cdots + b_nu(k-n)\} \qquad (3)$$

This is known as the equation or prediction error. In the standard linear least squares algorithm, the transfer function coefficients that minimize a scalar performance index $J = \sum_{k=1}^{N} e(k)^2$ are found using a recursive calculation.

The present algorithm is slightly more complicated. It uses the same error criterion, but considers each transfer function coefficient to be a function of one or two unknowns, say $a_n = f_n(x, y)$ and $b_n = g_n(x, y)$ where x and y are the inertias of two of the four disks. The function f_n (or g_n) is implicitly defined by the equations of motion of the system, and is very complicated. Having done this, the performance index J becomes a function of only two variables, x and y, rather than of all the a_n's and b_n's. Therefore ideally only two pieces of information are needed to find the best fit transfer function, since there are only two degrees of freedom in J. (Note that when the values of x and y that minimize J are determined, the transfer function coefficients are determined from the equations like $a_n = f_n(x, y)$.)

There are two obvious drawbacks to this approach. First, as mentioned, the functions f_n, g_n are generally very complicated. This difficulty is overcome by *approximating* them with polynomials. This can be accomplished numerically by a simple procedure to be described. Second, the problem of minimizing the performance index J no longer reduces to the solution of a set of linear equations. This too must be done numerically, using a gradient search algorithm which will also be described in detail. For a small number of independent variables the minimization can be accomplished in real time.

The final issue which must be addressed is the means of handling step changes in parameters, and the closely-related problem of switching off the identification algorithm when plant input is too small to dominate the effects of noise.

First something needs to be said about recursive versus batch identification. As already mentioned, the algorithms in the literature are usually recursive. Each new value of $e(k)$, along with all or most of the previously-measured values, is used to improve the estimate of the transfer function coefficients. This obviously helps alleviate the effects of noise in the measurements; but it is difficult to make such an algorithm respond quickly to step parameter changes. In contrast, it is hoped that reducing the number of unknowns to identify will provide enough robustness to noise so that a calculation using a small batch of data can be made. That is, a performance index is defined for each time step which sums N immediately-preceding values of $e(k)^2$ where N is *fixed*. For each time step a new (albeit overlapping) "batch" of data is used to form J, and a new pair of values (x, y) is calculated.

When a step parameter change occurs, for n time steps (where n is the order of the plant) it is no longer valid to write Equation 2. This is because Equation 2 is derived under an assumption of time invariance for at least n time steps. This limits the speed with which any identification algorithm based on a high order difference equation can adapt to a step parameter change. However, because of the batch nature of this identification algorithm, adaptation can occur $n+N$ steps after the abrupt change, where N is chosen to alleviate noise problems and is at least two (for two unknowns).

Clearly it is desirable for the identification algorithm to detect the time step at which the abrupt change occurs, since this essentially invalidates all the previous data, as well as the succeeding n data points. The identification calculation must be turned off until Equation 2 is again valid. A good method for doing this has not yet been determined.

The identification algorithm should also be turned off when the signal-to-noise ratio in the data drops too low.

Derivation of Algorithm

Detailed equations for the application of this identification algorithm to the four disk system are now given.

Calculation of f_n's

The calculation of the f_n's starts with the system equations in state space form. For each disc, let θ_i be the angular displacement, and ω_i be the corresponding angular velocity. Let k be the spring stiffness constant. Let J be the nominal inertia of each disk. Numbering discs from the top, the first and third have variable inertias, define x and y by the relations $J_1 = J/x$ and $J_3 = J/y$. Let $\omega_o^2 = k/J$. Let T be the motor torque, and let $u = T/J$. The system equations for any value of x and y are:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\omega}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_2 \\ \dot{\theta}_3 \\ \dot{\omega}_3 \\ \dot{\theta}_4 \\ \dot{\omega}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\omega_o^2 x & 0 & \omega_o^2 x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \omega_o^2 & 0 & -2\omega_o^2 & 0 & \omega_o^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \omega_o^2 y & 0 & -2\omega_o^2 y & 0 & \omega_o^2 y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \omega_o^2 & 0 & -\omega_o^2 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

These equations can also be written as $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u$. The next step is to calculate a set of discrete equivalent system equations in the form $\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \Gamma u(k)$. Since this is done numerically, values must be assigned to all constants, including x and y, and a sampling period must be chosen. A set of values for x and y was chosen as follows, $x \in \{.2(1+j)\}$ for $j = 1, \ldots, 24$ and $y \in \{.2(1+k)\}$ for $k = 1, \ldots, 12$. A typical sampling period is .1 seconds. The value of ω_o^2 is about $185\,\mathrm{sec}^{-2}$. To improve numerical accuracy, the equations are scaled by measuring time in units of tenths of seconds. As an example calculation, for $x = y = 1.0$, the following are obtained:

$$\Gamma = \begin{bmatrix} 0.0587 \\ 0.2154 \\ 0.3826 \\ 0.5683 \\ 0.0555 \\ 0.1974 \\ 0.0033 \\ 0.0188 \end{bmatrix} \qquad \mathbf{\Phi} = \begin{bmatrix} 0.3702 & 0.7649 & 0.5378 & 0.2154 & 0.0866 & 0.0188 & 0.0053 & 0.0008 \\ -0.9124 & 0.3702 & 0.5860 & 0.5378 & 0.2965 & 0.0866 & 0.0300 & 0.0053 \\ 0.5378 & 0.2154 & -0.0810 & 0.5683 & 0.4565 & 0.1974 & 0.0866 & 0.0188 \\ 0.5860 & 0.5378 & -1.2020 & -0.0810 & 0.3195 & 0.4565 & 0.2965 & 0.0866 \\ 0.0866 & 0.0188 & 0.4565 & 0.1974 & -0.0810 & 0.5683 & 0.5378 & 0.2154 \\ 0.2965 & 0.0866 & 0.3195 & 0.4565 & -1.2020 & -0.0810 & 0.5860 & 0.5378 \\ 0.0053 & 0.0008 & 0.0866 & 0.0188 & 0.5378 & 0.2154 & 0.3702 & 0.7649 \\ 0.0300 & 0.0053 & 0.2965 & 0.0866 & 0.5860 & 0.5378 & -0.9124 & 0.3702 \end{bmatrix}$$

Altogether $24 \times 12 = 288$ discrete equivalents were calculated, one for each pair of values (x_i, y_i). Next the discrete state space equations were converted to transfer functions, using the equation $H(z) = \mathbf{h}(z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{\Gamma}$. The measurement matrix is

$$\mathbf{h} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0].$$

As an example, for the same values of x and y as above,

$$H(z) = \frac{0.0033z^7 + 0.1285z^6 + 0.4659z^5 + 0.3675z^4 + 0.3675z^3 + 0.4659z^2 + 0.1285z + 0.0033}{z^8 - 1.1570z^7 + 0.8868z^6 - 1.4126z^5 + 1.3656z^4 - 1.4126z^3 + 0.8868z^2 - 1.1570z + 1.0}.$$

It is a useful simplification to note that $b_1 = b_8$, $b_2 = b_7$, etc. Now the polynomial approximations which give the a_n's and b_n's as functions of x and y are calculated using standard linear least-squares regression and the normal equations. The order of the fit (which is the same for all the coefficients) was chosen by trial and error, by simply increasing the order until the errors were acceptable. I now give the details of the calculation for an example, b_3. Define a vector $\mathbf{Y}$ of function values

$$\mathbf{Y} = [b_3(x_1, y_1), b_3(x_2, y_1), \ldots, b_3(x_{24}, y_1), b_3(x_1, y_2), \ldots, b_3(x_{24}, y_{12})]^T.$$

Next define a vector $\mathbf{\Theta}$ of the coefficients of the *least square fit polynomial* g_3:

$$\mathbf{\Theta} = [c_1, c_2, \ldots, c_{15}]^T.$$

This is for a polynomial which is fourth order in both x and y as follows:

$$g_3(x, y) = c_1 + c_2x + c_3y + c_4x^2 + c_5y^2 + c_6xy + c_7x^3 + c_8y^3 + c_9x^2y \\ + c_{10}xy^2 + c_{11}x^4 + c_{12}y^4 + c_{13}x^3y + c_{14}x^2y^2 + c_{15}xy^3$$

Now define a matrix $\mathbf{\Psi}$ of values of x_i, y_i, x_i^2, etc. so the the product $\mathbf{\Psi\Theta}$ will give a vector of values of the fit to $b_3(x, y)$

$$\mathbf{\Psi} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & \cdots & x_1y_1^3 \\ 1 & x_2 & y_1 & x_2^2 & \cdots & x_2y_1^3 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{24} & y_{12} & x_{24}^2 & \cdots & x_{24}y_{12}^3 \end{bmatrix}.$$

These quantities satisfy an equation of the form $\mathbf{Y} = \mathbf{\Psi\Theta} + \epsilon(\mathbf{\Theta})$ where ϵ is a vector of errors. Setting $\mathbf{\Theta} = (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T\mathbf{Y}$ minimises $J(\mathbf{\Theta}) = \epsilon^T\epsilon$. In the case of b_3, the vector $\mathbf{\Theta}$ is

$$\mathbf{\Theta} = [-0.0096, 0.0141, 0.4412, -0.0058, -0.1703, 0.2336, 0.0009, 0.0225, \\ -0.0267, -0.0378, 0.0001, -0.0013, 0.0008, 0.0029, 0.0016]$$

The same procedure gives a polynomial approximation to each of the transfer function coefficients.

Gradient Search Routine

In this section the equations used to find the values of x and y which best fit the measured plant inputs and outputs are derived. As already explained, this is done by minimizing numerically a performance index based on the equation error. Put the equations of the earlier section in matrix form. First define a vector $\mathbf{Y}$ of plant outputs:

$$\mathbf{Y} = [y(k), y(k-1), \ldots, y(k-N+1)]^T.$$

Here N is the number of time steps in the batch. Next define a row vector of plant outputs and inputs $\psi(n)$:

$$\psi(n) = [y(n-1), y(n-2), \ldots, y(n-N), u(n-1), u(n-2), \ldots, u(n-N)].$$

Here n refers to the nth time step. Now define a matrix $\mathbf{\Psi}$ whose nth row is $\psi(n)$:

$$\mathbf{\Psi} = \begin{bmatrix} \psi(k) \\ \psi(k-1) \\ \vdots \\ \psi(k-N+1) \end{bmatrix}.$$

I now define a matrix $\mathbf{P}$ and a vector $\mathbf{V}$ such that the product $\mathbf{PV}$ (where $\mathbf{V} = \mathbf{V}(x,y)$) is a vector of transfer function coefficients for some pair of values (x,y). This uses the definitions of the f_n's given earlier. Let

$$\mathbf{V} = \left[1, x, y, x^2, y^2, xy, \ldots, xy^3\right].$$

Let

$$\mathbf{P} = \begin{bmatrix} -c_1^1 & -c_2^1 & -c_3^1 & \ldots & -c_{15}^1 \\ -c_1^2 & -c_2^2 & -c_3^2 & \ldots & -c_{15}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_1^8 & -c_2^8 & -c_3^8 & \ldots & -c_{15}^8 \\ c_1^9 & c_2^9 & c_3^9 & \ldots & c_{15}^9 \\ c_1^{10} & c_2^{10} & c_3^{10} & \ldots & c_{15}^{10} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1^{16} & c_2^{16} & c_3^{16} & \ldots & c_{15}^{16} \end{bmatrix}.$$

Here the c_j^i are the j coefficients of the polynomial approximation to $a_i = f_i(x,y)$ or $b_i = g_i(x,y)$. It is easily verified that:

$$\mathbf{PV} = [-a_1, -a_2, \ldots, -a_8, b_1, b_2, \ldots, b_8]^T.$$

Comparison with Equation 3 shows that $\mathbf{E} = \mathbf{Y} - \mathbf{\Psi PV}$ where $\mathbf{E}$ is a vector of equation errors:

$$\mathbf{E} = [e(k), e(k-1), \ldots, e(k-N+1)]^T.$$

Finally, the performance index to be minimized is simply $J = 1/2(\mathbf{E}^T\mathbf{E})$. Unfortunately, J is not quadratic in x and y, so that minimizing J is a nonlinear problem. A numerical algorithm is used to find the pair of values (x, y) which minimize J. This algorithm can be easily switched between a second order Newton-Raphson method and a first order gradient search. The algorithm requires that the first and second partial derivatives of J with respect to x and y be available. Because polynomial approximations to the a_n's and b_n's were chosen, these derivatives are easy to calculate analytically. In fact, all the dependence on x's and y's is in the $\mathbf{V}$ vector. Therefore:

$$J = \frac{1}{2}(\mathbf{Y} - \mathbf{\Psi P V})^T(\mathbf{Y} - \mathbf{\Psi P V})$$

$$\frac{\partial J}{\partial x} = (\mathbf{Y} - \mathbf{\Psi P V})^T(-\mathbf{\Psi P V}_x)$$

$$\frac{\partial J}{\partial y} = (\mathbf{Y} - \mathbf{\Psi P V})^T(-\mathbf{\Psi P V}_y)$$

$$\frac{\partial^2 J}{\partial x^2} = (\mathbf{Y} - \mathbf{\Psi P V})^T(-\mathbf{\Psi P V}_{xx}) + (\mathbf{\Psi P V}_x)^T(\mathbf{\Psi P V}_x)$$

$$\frac{\partial^2 J}{\partial y^2} = (\mathbf{Y} - \mathbf{\Psi P V})^T(-\mathbf{\Psi P V}_{yy}) + (\mathbf{\Psi P V}_y)^T(\mathbf{\Psi P V}_y)$$

$$\frac{\partial^2 J}{\partial x \partial y} = (\mathbf{Y} - \mathbf{\Psi P V})^T(-\mathbf{\Psi P V}_{xy}) + (\mathbf{\Psi P V}_x)^T(\mathbf{\Psi P V}_y)$$

Here $\mathbf{V}_x$ is used for $\partial \mathbf{V}/\partial x$, etc. Calculating these derivatives of $\mathbf{V}$ is trivial, for example,

$$\mathbf{V}_x = \left[0, 1, 0, 2x, 0, y, 3x^2, 0, 2xy, y^2, 4x^3, 0, 3x^2y, 2xy^2, y^3\right]^T.$$

The algorithm uses the Hessian matrix $\mathbf{H}(x, y)$ defined as:

$$\mathbf{H}(x, y) = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix}.$$

Given a guess of x and y, the formula giving the new values which hopefully improve the performance index J is:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \alpha \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix}^{-1} \begin{bmatrix} J_x \\ J_y \end{bmatrix}.$$

In the control program, the size of α is adjusted until an improvement in J occurs. Also, $\mathbf{H}(x, y)$ is tested for positive-definiteness, if the test is not satisfied an increment is added to the diagonal of $\mathbf{H}$ until it is just barely positive definite. This effectively changes the algorithm from second order Newton-Raphson to a first order gradient search. Given sufficient computer speed, a good approximation to the values of x and y which minimize J can be made during each time step. Finally, these identified values of x and y are plugged back into the $\mathbf{V}$ vector and the product $\mathbf{PV}$ is formed. This product gives the identified transfer function coefficients in the form of a column vector ready for input to the control law algorithm.

Control Law Calculation

The control law algorithm basically computes the transfer function of a compensator that will locate the closed loop system poles in predefined locations. It is an unfortunate fact that such an approach requires the use of a compensator of order at least $n-1$, where n is the order of the plant. However, the logic and implementation of the algorithm are simple, which is what makes it an attractive method.

As earlier, denote the discrete plant transfer function (which is calculated by the identification algorithm already described by $H(z) = b(z)/a(z)$. The compensator transfer function is assigned the form:

$$u(z) = \frac{m(z)}{l(z)}r(z) - \frac{p(z)}{l(z)}y(z). \tag{4}$$

Here $r(z)$ is the command input, and m, l, and p are polynomials in z. Note that the compensator has two inputs, the plant output $y(z)$ and the command input, and one output, the control signal $u(z)$. As noted in the discussion of Equation 1, once such a transfer function is known, it can be converted immediately into a difference equation for $u(k)$. This is the approach taken in the control program.

Given a compensator transfer function of the form (4), some simple calculations show that the overall transfer function from r to y of the closed loop system has a characteristic polynomial $a^*(z)$ given by:

$$a^*(z) = l(z)a(z) + p(z)b(z). \tag{5}$$

This equation is sufficient to determine the polynomials $l(z)$ and $p(z)$ from the identified plant polynomials $a(z)$ and $b(z)$ and the desired closed loop characteristic polynomial $a^*(z)$. There is a small amount of latitude in choosing the desired order of a^*, p and l, corresponding to a choice of a predictor, current, or Gopinath compensator. The choice of a current compensator makes a^* 16th order, l 8th order, and p 7th order (for the 8th order four disc system). Specifically,

$$\begin{aligned} a^*(z) &= 1 + a_1^* z^{-1} + \cdots + a_{16}^* z^{-16} \\ l(z) &= 1 + l_1 z^{-1} + \cdots + l_8 z^{-8} \quad . \\ p(z) &= p_0 + p_1 z^{-1} + \cdots + p_7 z^{-7} \end{aligned}$$

The control program reformulates Equation 5 as a 17th order system of linear algebraic equations in the coefficients of l and p which is then solved by Gaussian elimination. This is done by equating like powers of z and results in the following system:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 & b_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & \cdots & 0 & b_2 & b_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_7 & a_6 & a_5 & \cdots & 0 & b_7 & b_6 & b_5 & \cdots & 0 \\ a_8 & a_7 & a_6 & \cdots & 1 & b_8 & b_7 & b_6 & \cdots & b_1 \\ 0 & a_8 & a_7 & \cdots & a_1 & 0 & b_8 & b_7 & \cdots & b_2 \\ 0 & 0 & a_8 & \cdots & a_2 & 0 & 0 & b_8 & \cdots & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_7 & 0 & 0 & 0 & \cdots & b_7 \\ 0 & 0 & 0 & \cdots & a_8 & 0 & 0 & 0 & \cdots & b_8 \end{bmatrix} \begin{bmatrix} 1 \\ l_1 \\ l_2 \\ \vdots \\ l_7 \\ l_8 \\ p_0 \\ p_1 \\ \vdots \\ p_6 \\ p_7 \end{bmatrix} = \begin{bmatrix} 1 \\ a_1^* \\ a_2^* \\ \vdots \\ a_7^* \\ a_8^* \\ a_9^* \\ a_{10}^* \\ \vdots \\ a_{15}^* \\ a_{16}^* \end{bmatrix}.$$

In the control program, the solution vector to this equation is substituted directly into a difference equation formed from Equation 4 which results in the control signal. The polynomial $m(z)$ is calculated in advance (except for a D.C. gain) and affects only the overall zeros of the closed loop system. m is chosen so that it cancels eight of the poles of the closed loop system, and a D.C. gain for m is calculated based on the results of the identification such that the D.C. gain of the closed loop system is unity.

Experimental Results

To date only the identification algorithm has been tested experimentally, due to difficulties encountered with its performance. Many experimental tests of the identification algorithm have been made using fixed-gain controllers to operate the plant. To simplify testing, only one parameter, the inertia of the top disk, has been estimated. Several sampling rates have been utilized, and the effect of sampling rate on identification performance has been studied.

Figure 3d.-2 displays the data from a typical experiment in which a 10 Hz sampling rate was used. Both rings were in place on the top disk during this experiment. Figure 3d.-2 is a plot of the plant output (angular position of the bottom disk in radians) versus time in seconds. Figure 3d.-2 shows the command input which generated this output, a 2.7 Hz sine wave. Figure 3d.-2 is a plot of the estimate of the inertia of the top disk divided into the nominal value of inertia with both rings in place. As can be seen, the estimate is quite good for this combination of sampling rate and command input, showing that the identification algorithm is working properly. The 'batch' size is 10 samples. Figure 3d.-2 shows the input signal to the torque motor.

Figure 3d.-3 displays the data from a similar experiment in which a 20 Hz sampling rate was used. Again both rings were in place, the command input is a 2.7 Hz sine wave, and the batch size is 10 samples. The closed loop poles of the fixed gain controller used for this experiment correspond to those used in the experiment of Figure 2 under the mapping $z' = z^{.5}$ where z' is the pole location for 20 Hz sampling rate. It is notable that the parameter estimate is very poor compared to the 10 Hz sample rate case.

After a long investigation, the source of difficulty was determined. Figure 3d.-4 is a plot of a section of the plant output sequence for two successive runs of the experiment. As can be seen, repeatability of the experimentally measured plant output, while extremely good, is not perfect. The differences plotted are never more than two bits of resolution of the A/D converter, representing about 4% of the output swing for this particular experiment. Also plotted are the *predicted* values of the plant output for the ninth sample calculated from the first eight experimental output values for each of the two runs, using difference equation coefficient values corresponding to the actual value of the top disk inertia. Clearly the predicted values are quite different, despite the almost identical values of the plant output on which the predictions are based. (The effect of the past inputs on the prediction can be calculated separately and are negligible in the present instance.) Thus it is seen that the predictor Equation 2 has the pathological property (for the case of 20 Hz sampling rate) that tiny perturbations in the values of the past plant outputs $y(k)$ cause large variations in the value of the prediction. It is this property which results in the very poor performance of the identification algorithm illustrated in Figure 3d.-3.

A clue as to the source of the problem is seen in Table 3d.-1, which lists the coefficients of the predictor equation for the case of nominal (both rings in place) top disk inertia and both sampling rates. Both sampling rates result in coefficients which alternate in sign, but those of the equation corresponding to 20 Hz sampling are almost 100 times larger.

Some additional insight can be gained by examining the frequency response of the operator $a(z)$. Figure 3d.-5 shows a plot of the magnitude of $a(e^{j\omega T})$ versus frequency for both values of sampling rate. Largely because of the increased Nyquist frequency for the 20 Hz sampling rate, the maximum gain of the operator $a(z)$ is increased by a factor of about 15 compared to the maximum gain for 10 Hz sampling. (This is not a suprising fact, if the operator $1/a(z)$ had been plotted instead, the bode plot would have shown the usual high frequency roll-off of a 4-pole transfer function. The higher Nyquist frequency merely allows for more roll-off before aliasing begins.) To see the effect that such a large high frequency gain might have on the identification process, consider an approximate frequency domain expression for the identification performance index J:*

$$J \approx \frac{1}{2} \sum_{k=1}^{N} |a(e^{j\omega_k T}, \theta)|^2 |u(\omega_k)|^2 |\hat{G}(e^{j\omega_k T}) - G(e^{j\omega_k T}, \theta)|^2.$$

Here N is the number of samples, $\hat{G}$ is the ratio of the periodogram of the output sequence $y(\omega_k)$ to the periodogram of the input sequence $u(\omega_k)$, (termed by Ljung the 'empirical transfer function estimate') and $G(z, \theta)$ is the estimated transfer function depending on the parameter vector θ, $G(z, \theta) = b(z, \theta)/a(z, \theta)$. The error in the approximation decays like $1/\sqrt{N}$ and also depends on the rate of decay of the impulse response of the plant, which is slow for the lightly damped system under consideration. Nevertheless, at least a qualitative understanding of the problems encountered with the identification algorithm can be gained from this formula. The very large high frequency gain of the analytically calculated $a(\omega)$ would tend to give heavy emphasis to fitting G to $\hat{G}$ in the highest frequency regions, where $\hat{G}$ is dominated by noise. Consequently, the algorithm does not estimate $a(\omega)$ properly, in fact it tends to calculate an $a(\omega)$ which does not roll off as fast at high frequencies as the analytically predicted a.

* *Ljung, Lennart, System Identification, Theory for the User, 1984, Unpublished Manuscript.*

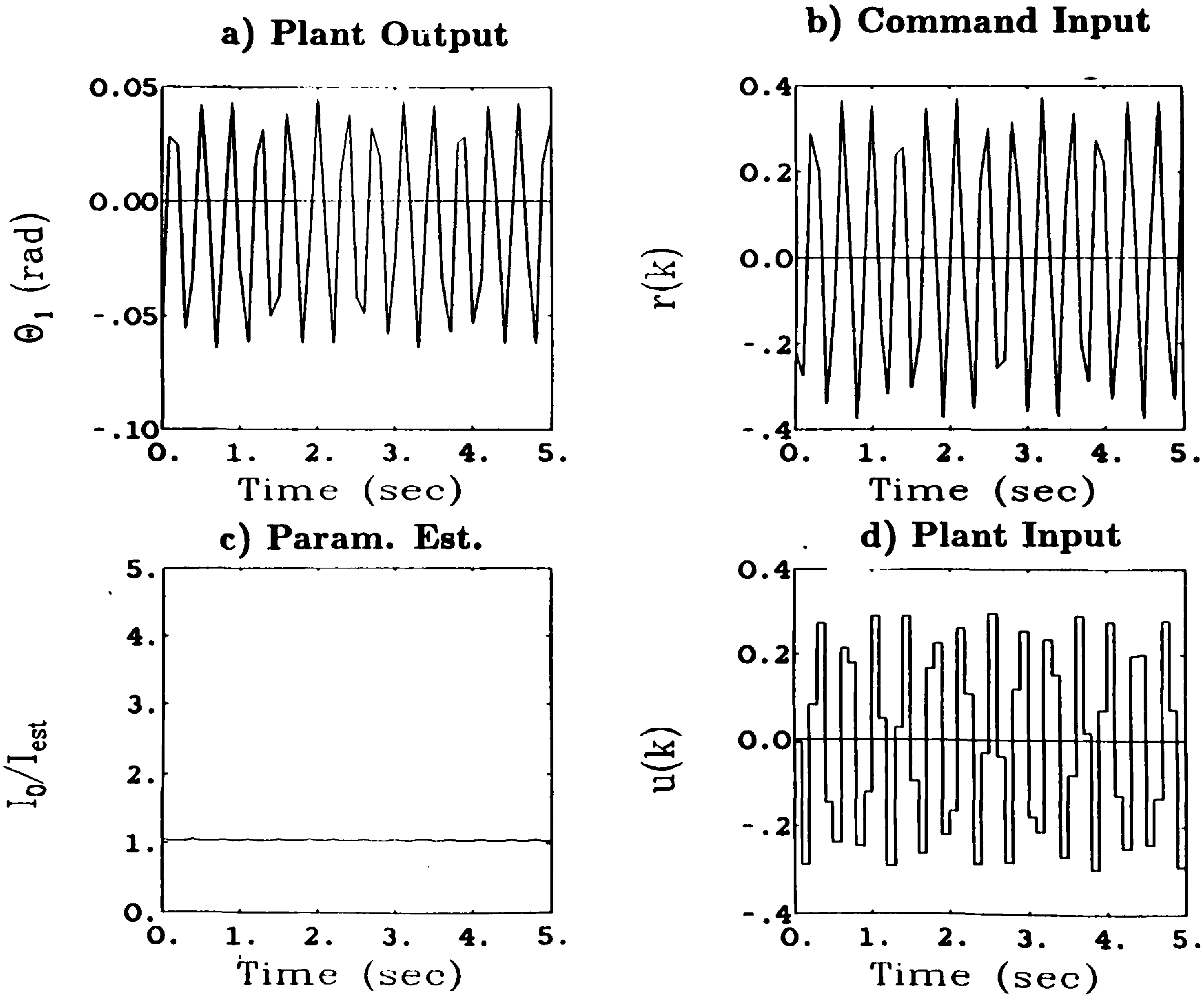

a) Plant Output
Θ_1 (rad)
0.05
0.00
-.05
-.10
0.
1.
2.
3.
4.
5.
Time (sec)
b) Command Input
r(k)
0.4
0.2
0.0
-.2
-.4
0.
1.
2.
3.
4.
5.
Time (sec)
c) Param. Est.
I_0/I_{est}
5.
4.
3.
2.
1.
0.
0.
1.
2.
3.
4.
5.
Time (sec)
d) Plant Input
u(k)
0.4
0.2
0.0
-.2
-.4
0.
1.
2.
3.
4.
5.
Time (sec)

Fig. 3d.-2

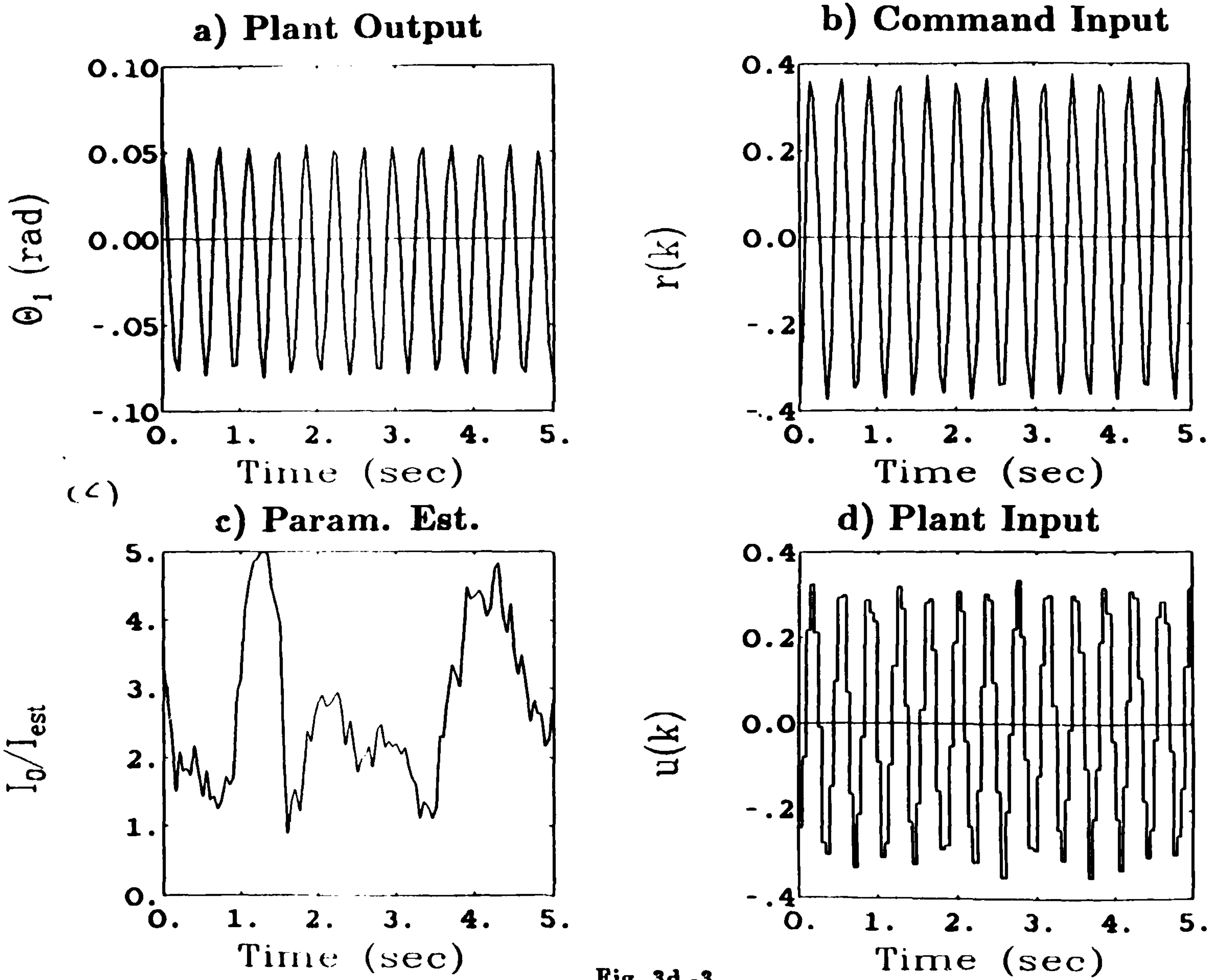
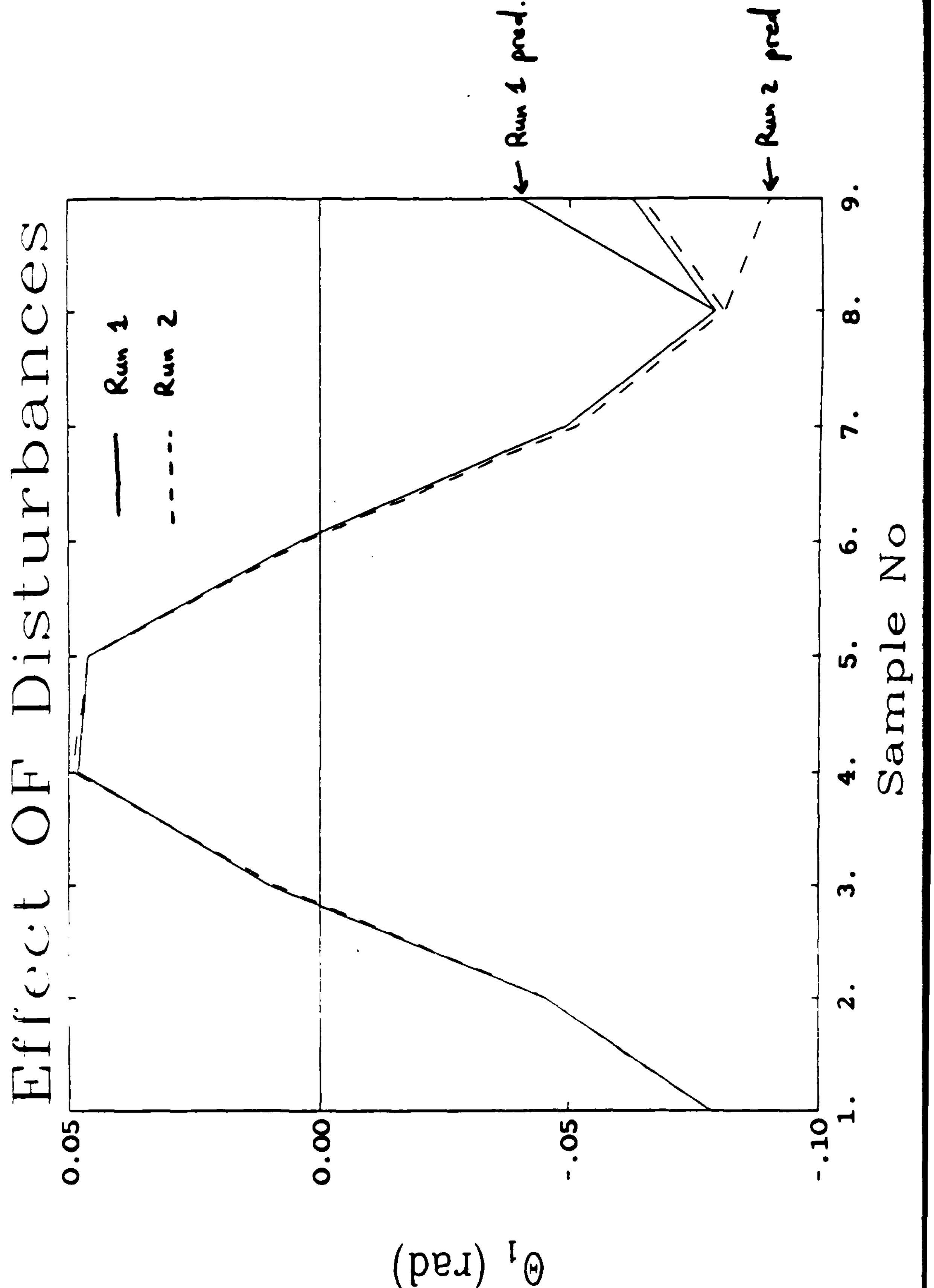

a) Plant Output
Θ_1 (rad)
0.10
0.05
0.00
-.05
-.10
0.
1.
2.
3.
4.
5.
Time (sec)
b) Command Input
r(k)
0.4
0.2
0.0
-.2
-.4
0.
1.
2.
3.
4.
5.
Time (sec)
c) Param. Est.
I_0/I_{est}
5.
4.
3.
2.
1.
0.
0.
1.
2.
3.
4.
5.
Time (sec)
d) Plant Input
u(k)
0.4
0.2
0.0
-.2
-.4
0.
1.
2.
3.
4.
5.
Time (sec)

Fig. 3d.-3


Effect OF Disturbances
Run 1
Run 2
Run 1 pred.
Run 2 pred
0.05
0.00
-.05
-.10
Θ_1 (rad)
1.
2.
3.
4.
5.
6.
7.
8.
9.
Sample No

20 Hz Denominator Coefficients

a0	a1	a2	a3	a4	a5	a6	a7	a8
1.0000	-5.4953	14.7515	-25.2328	29.9535	-25.2328	14.7515	-5.4953	1.0000

10 Hz Denominator Coefficients

a0	a1	a2	a3	a4	a5	a6	a7	a8
1.0000	-0.6951	0.1899	-0.8008	0.6337	-0.8008	0.1899	-0.6951	1.0000

Table 3d.-1 Transfer Function Coefficients

These considerations also suggest a solution to the problem, which is to filter the prediction error sequence (which is equivalent to filtering the input output data $\{y(k) \quad u(k)\}$) through a low-pass filter. (This approach was also suggested by Sidman. See Approach 2 below.) The integrand in the above expression is then multiplied by the magnitude squared of the filter transfer function, say $|L(e^{j\omega})|^2$. The effects of high frequency noise in $\hat{G}$ can thus be mitigated to an extent.

Figure 3d.-6 illustrates the effect of a fourth order butterworth filter on the performance of the identification algorithm for the same data as in Figure 3d.-3. A dramatic improvement is seen.

The effects discussed are not peculiar to the identification algorithm described, which uses a reduced-order parameterization of the trial transfer functions. They are instead related to the choice of performance index minimised by the identification algorithm. A different choice of performance index might give better performance.

APPROACH 2: Full-Order Representation

What is needed is a technique or combination of techniques that reliably lead to unbiased, accurate plant parameter estimates in a relatively small number of sample intervals. This methodology should be insensitive to bias forces in the actuator and sensor offsets as well as measurement noise generated in the A/D conversion process itself. Lastly, this method should not require excessive computation and should be insensitive to unmodeled dynamics if it is to find practical application. Admittedly, this is an ambitious list, but one that is not impossible to satisfy with some minor compromise. The essence of this compromise is to make limited use of known information about the plant to be controlled.

One set of information required to guarantee accurate, fast and reliable parameter estimation of lightly-damped mechanical plants is the nature of disturbances in the control system. Because the identification method pursued here is targeted to feedback control systems, the assumed disturbances take the form of broadband measurement noise, dc and low-frequency bias forces, position-sensor offset and high-frequency unmodeled plant dynamics.

In essence, the problem is evolved here into making use of limited information about the plant without knowing the frequencies of resonances and antiresonances and how they might interact with each other for a given physical plant. In so doing, it has been possible to reduce the number of parameters required to identify the plant by approximately a factor of four, even though the plant dynamics are fully represented. The accuracy of the identification has also improved dramatically, even for relatively short sets of plant input/output data that constitute training sequences.

An ARMAX (Auto-regressive Moving Average with Exogenous input) model of the plant is assumed. In the case of the Four-Disk System (Fig. 3d.-1), the ARMA portion of the model is described with sixteen parameters, which also represent the eight coefficients of the characteristic equation and the eight coefficients of the numerator polynomial of the plant's discrete-time transfer function.

The process for identifying this type of plant is really a combination of techniques

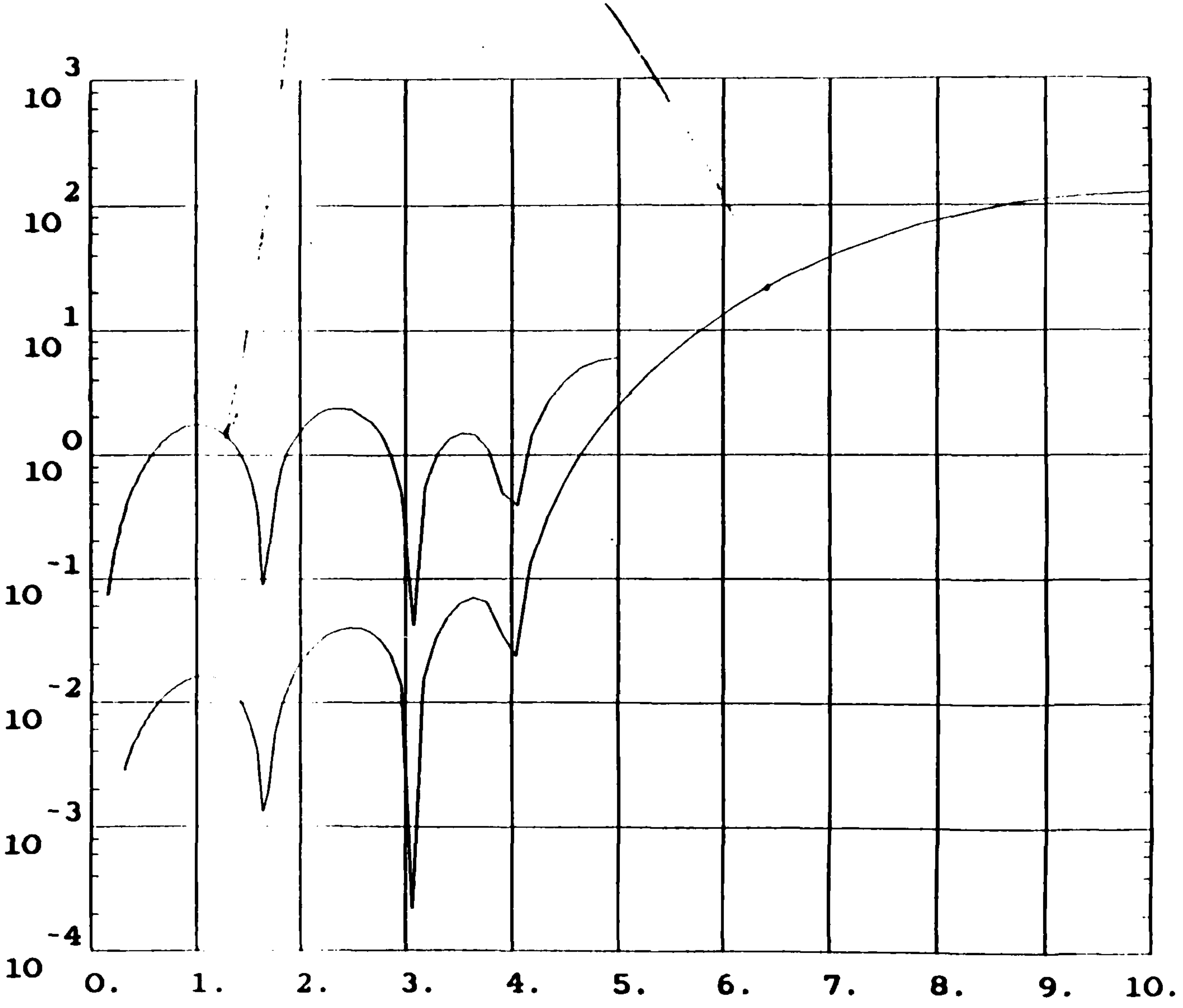


Fig. 3.1.5

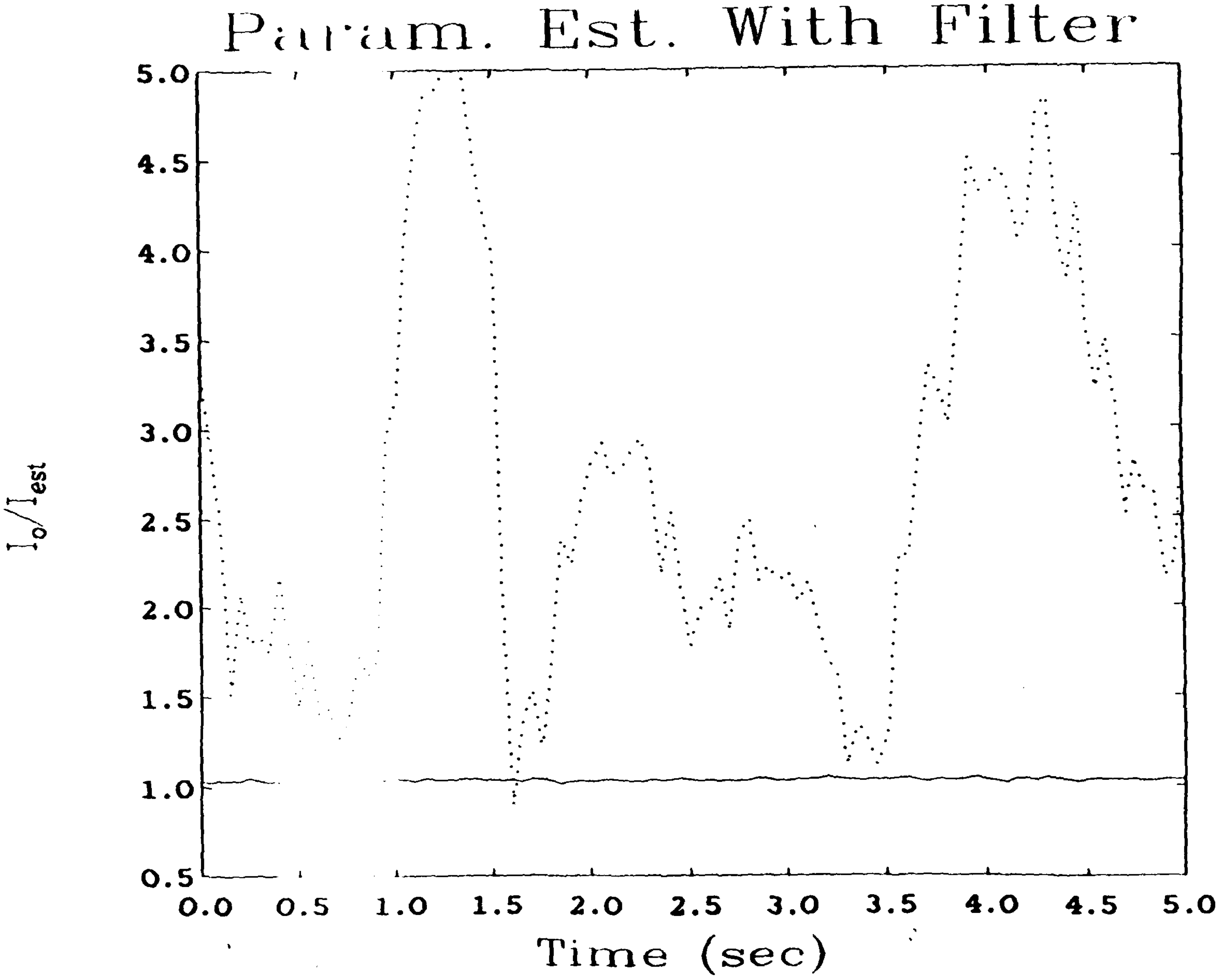

Param. Est. With Filter
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
I_o/I_{est}
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Time (sec)

Fig. 3d.-6

each of which independently can be shown to improve the quality of identification, but which together act in a complementary manner.

The final algorithm has been successfully demonstrated in real-time on the Mechanical Four-Disk System. However, learning what either doesn't work well or isn't practical is probably as important as what appears now to be an effective means of on-line identification for adaptive control.

Summary of Major Results: FWRLS Identification:

Identifying all sixteen of the Four-Disk System's ARMA parameters in the presence of minute amounts of measurement noise results in poor identification of the plant zeroes that lie along the negative real axis in the Z plane. Since the true locations of these zeroes, which come in from infinity in the continuous domain, are quite insensitive to plant variation caused by important types of plant variation such as change in inertia, they can be assumed known *a priori* and removed from the identification problem. In fact these zeroes play a minor role in the time response of the plant other than delay. Likewise, the poles corresponding to the rigid body modes are invariant to such types of plant variation and should be removed from the identification problem.

The FWRLS method developed by Clary makes use of the known plant poles and zeroes and reduces the number of parameters to be identified substantially. Pursuing fewer parameters result in faster convergence and greatly reduced computational loads. The Four-Disk System, which is an eighth-order system, could now be reduced to a nine-parameter identification problem using this method. Since the computational burden associated with identification is a geometric function of the number of parameters to be estimated, a factor of two reduction in the number of parameters by this method resulted in about a factor of four reduction in computation time.

The Problem with Broadband Measurement Noise:

Measurement noise arising from A/D quantization and sensors will 'bias' the parameter estimates substantially in batch or recursive least squares (LS) identification.

This happens because even white measurement noise becomes colored equation error in the least-squares problem formulation. "Colored" is a euphemism for many problems where the number of poles of the unknown, modeled portion of the plant is high and their damping is low. Since LS minimizes the sum of the squared equation error without regard to its whiteness, the result is the identification of false high-frequency dynamics which fit the high-frequency measured plant output. Thus, the very feature of LS which leads to a single, unique minimum solution of a quadratic performance index causes parameter bias.

An effective way to whiten the equation-error and get accurate identification from least-squares is to prefilter the plant input/output data by the reciprocal of the plant characteristic equation. Since this equation is presumed unknown perhaps except for the rigid body poles, this method can only partially be implemented practically.

Other well-known methods such a RML2, RML1 and Steiglitz-McBride essentially can be viewed as recursively updating a filter based on the current estimate of the characteristic equation coming out of the identification. These methods have the potential of solving the problem but often converge slowly and to undesirable local minima in actual simulations. Thus, they tend to provide grossly biased results unless they are started in the vicinity of the true solution. Further, they are quite computationally intensive relative to LS.

Instrumental variable identification may be used to deal with plant measurement noise, but requires a reference signal that is uncorrelated with the noise. This poses a difficult problem for closed-loop identification, where the plant input is in part derived from the measured plant output. Methods that attempt to overcome this problem end up either adding test signals into the loop and/or identifying the closed-loop system, which is of much higher order than the plant itself.

"Modified FRLS Identification"

The method of prefiltering used in FWRLS does not solve this problem even in part because the equation-error remains colored by the entire plant characteristic equation. However, an improved FLS scheme that filters only the plant input data by the known plant numerator polynomial divided by the stabilized (radially projected) known plant characteristic polynomial results in equation error colored only by the unknown portion of the plant characteristic equation. This technique, which shall be called 'Modified FRLS', helps appreciably. Essentially, when applied to the Four-Disk System, it whitens the equation-error by eliminating the effect of the two rigid-body poles of the plant.

Artificial Perturbation or "Buzz"

Simply adding white-noise buzz to the closed-loop system to improve the unfiltered LS identification does not necessarily help much by itself. The necessary levels of white noise or buzz necessary to overcome the identification difficulties experienced with measurement noise in lightly-damped plants may be so large as to make this method alone ineffective as was seen in the Four-Disk System simulations.

However, in combination with some of the other methods described here it may be both useful and necessary to buzz the plant at low levels at some or all times in order to keep the identification sensitive to step changes in plant parameters during quiescent periods. The goal is to maximize the effectiveness of the perturbation signal to facilitate fast identification while limiting its effect on system performance. Thus, buzz ideally should be limited in frequency to those portions of the spectrum where modeled dynamics that need to be identified exist.

Pre-Filtering of Plant Input/Output Data:

In most situations the upper frequency limit of modeled dynamics is known. By low-pass filtering the plant input/output data with a high-order digital low-pass filter with a band-edge just above this limit, several very beneficial results were obtained. The signal/noise ratio of the data to the identification is improved by reducing the effect of measurement noise on equation-error without significantly affecting the signal energy associated with the modeled dynamics of the plant. Since LS identification will tend to identify dynamics only in frequency bands where input/output signal energy exists, the problem of estimating phantom high-frequency modes from noisy data is greatly minimized with this method. Unmodeled high-frequency dynamics will also be far less likely to bias the identification since the filter masks them out. Because this filtering is not in the closed-loop, it does not directly affect system performance other than through the adaptation process.

Typically, a low-frequency limit of modeled but unknown dynamics is also known. By adding a digital high-pass filter with zero dc gain on the plant input/output data to the identification, the effects of position sensor offset and actuator dc bias forces bay be eliminated as contributors to identification parameter estimate bias. This technique compliments Modified FLS since in that method the plant's dc dynamics, the rigid-body poles, are removed from the identification. Obviously, it is not necessary to pass dc or very low-frequency components of the input/output data to the identification since there is nothing to identify there.

The combination of these two filter types is a flat-topped bandpass filter that windows the frequency band in which the unknown modeled dynamics are known to reside. This technique in combination with the Modified FLS method seems to do an excellent job of identification with measurement noise and in relatively short time periods.

"Adidas" — Assumed-Damping Identification for Adaptive Servos

One feature of the Four-Disk System and many servo systems that drive flexible structures is its very light damping. It is possible to make use of this information in order to simultaneously improve the quality of identification and reduce the number of identification parameters almost in half again (from nine to five). The number five corresponds to the number of degrees of freedom in plant parameter space: the frequencies of three resonances and one antiresonance and the gain of the overall transfer function.

Incidentally, estimating this overall gain factor is crucial in many systems due to its variability; the Four-Disk System has proved not to be an exception.

Essentially this technique assumes that all the plant poles and zeroes are at the same, known radius from the Z plane origin. For lightly-damped systems the 'known' radius is just slightly less than one. The reducticn in the number of ARMA parameters comes from the resulting symmetry in the coefficients of the plant numerator and denominator polynomials. For example, the first and last coefficients of the characteristic equation are the same by a factor of radius to the nth power, where n is the order of the polynomial. The second and next to last coefficients are the same by a factor of radius to the (n-2)th power and so on.

This method also compliments Modified FLS since only lightly-damped complex poles and zeroes are left in the identification problem for flexible mechanical plants. Results indicate faster convergence, more accurate estimation of resonant frequencies and as expected very good damping or pole-zero radii accuracy. Lastly, computation time should be down by an order of magnitude versus the original sixteen parameter LS identification.

Overparameterization:

Overparameterization is another means of whitening the equation error and accurately estimating the actual poles of the system. Simulation runs for the Four-Disk System show that at least doubling the number of parameters of the LS identification yields numerator and denominator polynomials whose roots include values that are quite close to the true complex poles and zeroes of the system even without pre-filtering in the presence of measurement noise. In the case of zero measurement noise, the excess estimated poles exactly cancelled the excess estimated zeroes.

However, this was not the case with measurement noise present, and there appears no easy on-line method to extract the true plant poles and zeroes from the overparameterized model at this time.

These effects have also been demonstrated first with the Electronic Four Disk System, an equivalent electronic version of the Four-Disk Mechanical System, using a Genrad GR-2515 structural analyzer. The pole-zero cancellation effect identification does not appear to be any more straightforward even with a twenty-fourth order identification model, one thousand sample points and low-pass pre-filtering.

In the coming year we shall proceed with experiments on the Four-Disk Mechanical System of Fig. 3d.-1.

APPROACH 3: Real Time Frequency Identification

This Approach was pursued by Wen Wie Chiang for future use with his Flexible Arm with Quick Wrist, and was demonstrated by him on a Two-Disk Mechanical System in the second year of his contract.

It is shown clearly that the vibration associated with the third structural mode, at a frequency higher than the control bandwidth, shows up in the wrist's tip position in both the mechanical system maneuvers, and in the theoretical numerical analysis. What is discussed here is whether this mode can be suppressed, even though it is beyond the control bandwidth.

In the mechanical experimental wrist-beam system, the third structural mode cannot be modeled correctly and easily since it is coupled with the vertical vibration of the structure which cannot be described with linear equations. This mode oscillating at 10 Hz cannot be controlled by the hub motor loop running at 25 Hz, nor can its effect on the wrist tip's motion be suppressed by the wrist motor control loop with a bandwidth of 4.6 Hz.

Numerical analysis shows that the effect of the third mode still cannot be eliminated even if we include its model in the controller designs, unless a very heavy penalty is imposed upon it in the LQG design, or an exact pole-zero cancellation is chosen in the pole placement design, both of which require a very accurate model of the system dynamics.

Another option is to use an adaptive control scheme that identifies the system dynamics and updates the controller parameters in real time. A feedforward filter will be sufficient since the third mode is at a frequency higher than the control bandwidth, and a feedforward filter can avoid exciting it by notching out the proper frequency component out of the motion commands. The stability of the system with the existing feedback controller will not be changed by a feedforward filter, nor will it be influenced by the dynamics of the adaptation algorithm.

It is not necessary to identify the whole dynamics of the system for implementing the adaptive feedforward filter, only the frequency of the third structural mode is required; and it can be done fast with a new frequency identification scheme – that is the subject of this section. The same frequency identification scheme may also be used to identify the first structural frequency of the main beam to tune an adaptive feedback controller for the hub motor loop, in order to achieve a high performance in the presence of modeling uncertainty.

The transfer function from the hub motor torque input to the tip position of a flexible beam has positive residues for the rigid body and all even numbered flexible modes, and negative residues for all odd numbered modes. It is the odd numbered flexible modes that cause phase reversal of the feedback control signal and destabilise the closed loop system.

Only the frequency of the first flexible mode is important since the ultimate performance of a controller is still limited by this mode even when an exact dynamic model of the system is available to design the controller.

A closed loop adaptive controller is much more difficult to design, since the dynamics of the feedback controller interacts with that of the system identification algorithm, and the overall system stability must be ensured while achieving a good performance in reasonable time.

Neither the feedforward adaptive filter nor the feedback adaptive controller has been designed and implemented for the wrist-beam system, but the design of the frequency identification algorithm has been completed and tested successfully in a closed-loop adaptive feedback controller for another mechanical system with low damping, flexible structure, and non-colocated actuator-sensor pair.

The Frequency Identification Scheme

The frequency identification scheme originates from the popular Phase-Locked-Loop (PLL), and the theory of its operation is described briefly in this section. The character of this loop is discussed along with the reason for its inadequacy for frequency identification in adaptive controls.

It is described later in this section how the PLL is modified to serve the need of the adaptive controls.

The Phase-Locked-Loop

PLLs have been used widely in electronic engineering applications. They are used primarily for synchronizing clocks with external signals. A PLL has the ability to identify the phase and frequency of a signal contaminated by a relatively large amount of noise at other frequencies. The loop works by tracking the phase of an external signal, which results in locking onto the correct frequency.

Fig. 3d.-7 shows the block diagram of an analog PLL. A voltage controlled oscillator (VCO) generates a sinusoidal signal with the frequency determined by its input voltage ω. The sinusoidal output, which is the output of the PLL also, is multiplied by the external (input) signal, and the product is low-passed to retain the component which is a function of the phase error between the input and output signals. The signal due to the phase difference is used to correct the frequency of the VCO until a steady state is reached.

Their performance depends on how the low-pass filter is implemented, and how the phase error signal is processed to adjust the VCO frequency. Linear analysis can be performed for small frequency and/or phase errors; but the dynamic characteristics of such systems are difficult to analyze due to their nonlinearity, and test results are not easy to characterize either. PLLs usually have different searching ranges and holding ranges, and the former are usually smaller than the latter.

PLLs have some properties that make them unsuitable for frequency identification in adaptive controls:

1. Performance is hard to predict and describe due to the nonlinearity as mentioned earlier. Therefore, it is hard to use them in an adaptive controller design.
2. Relatively small searching range makes the performance of the adaptive controller unacceptable.
3. They have higher sensitivity to phase error than to frequency error, which is unsuitable for adaptive control, since the frequency identification will be disturbed by the random phase of the sensor signal every time a new position command or an external disturbance is applied to the system, even after the frequency has been identified already.
4. Responses are sensitive to the amplitude of the input signal, and a high gain amplification is required to saturate the input channel to reduce this sensitivity. However, such a saturation changes the frequency component of the input signal, and makes it harder for multiple frequency identification.

The Frequency-Locked-Loop

The author has made modification to a PLL to eliminate its sensitivity to phase in the input signal and make the input/output relation much more linear in a larger searching and holding range, so that it works better for frequency identification, while retaining the other virtues of PLL's. The final product, called a Frequency-Locked-Loop (FLL), is shown schematically in Fig. 3d.-8a. It shows that the FLL uses one more multiplier, three more low pass filters, and a trigonometric function than the PLL does, so that it can obtain a signal which is proportional to the frequency error between the input and output signals of the loop. Its input/output relation can be seen from the functional block diagram shown in Fig. 3d.-8b, where ω_i is the frequency of the input signal and ω_o is the output signal — the frequency detected by the FLL. Also shown in the same block diagram are ω_s, the starting oscillation frequency; $\Delta\omega$, the correction on the output; and ω_{er}, the error of the output of the FLL.

The character of the block $G(s)$ can be chosen arbitrarily by the designer as long as

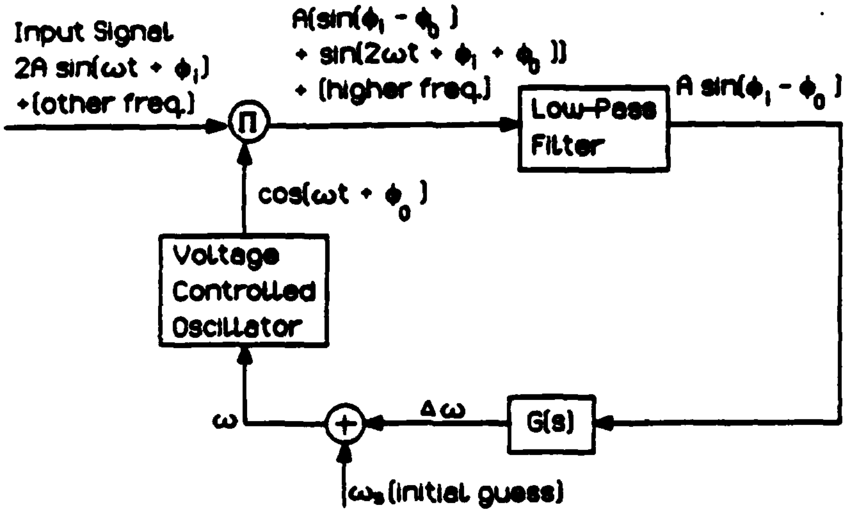
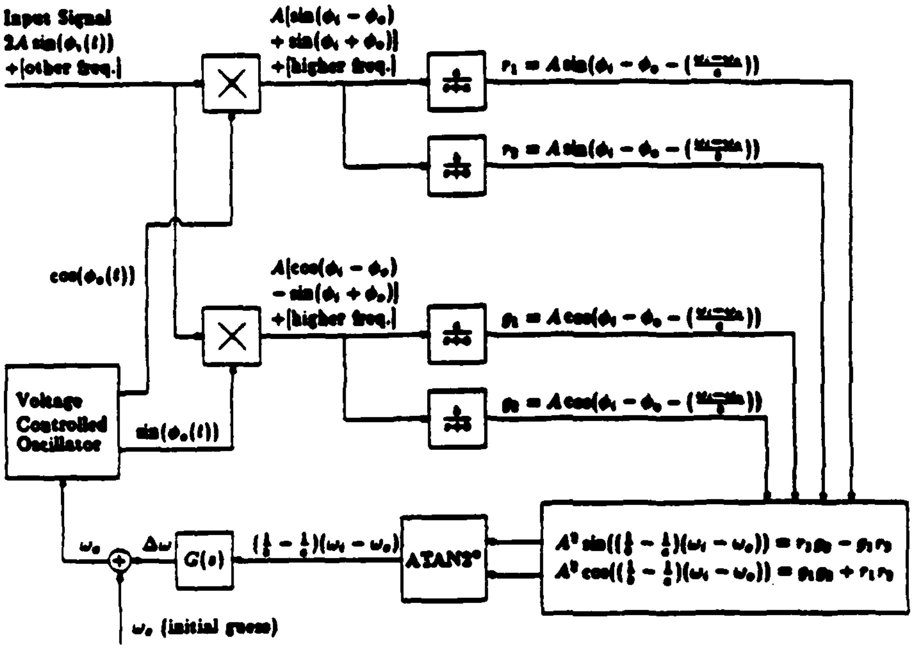

Input Signal
$2A \sin(\omega t + \phi_i)$
+(other freq.)
Π
$A(\sin(\phi_i - \phi_0)$
$+ \sin(2\omega t + \phi_i + \phi_0))$
+ (higher freq.)
Low-Pass Filter
$A \sin(\phi_i - \phi_0)$
$\cos(\omega t + \phi_0)$
Voltage Controlled Oscillator
ω
$+$
$\Delta\omega$
G(s)
ω_s (initial guess)

Fig. 3d.-7 The block diagram of the basic Phase-Locked-Loop.


Input Signal
$2A\sin(\phi_i(t))$
+[other freq.]
$A[\sin(\phi_i - \phi_o)$
$+\sin(\phi_i + \phi_o)]$
+[higher freq.]
$\frac{a}{s+a}$
$r_1 = A\sin(\phi_i - \phi_o - (\frac{\omega_i - \omega_o}{a}))$
$\frac{b}{s+b}$
$r_2 = A\sin(\phi_i - \phi_o - (\frac{\omega_i - \omega_o}{b}))$
$\cos(\phi_o(t))$
$A[\cos(\phi_i - \phi_o)$
$-\sin(\phi_i + \phi_o)]$
+[higher freq.]
$\frac{a}{s+a}$
$g_1 = A\cos(\phi_i - \phi_o - (\frac{\omega_i - \omega_o}{a}))$
$\frac{b}{s+b}$
$g_2 = A\cos(\phi_i - \phi_o - (\frac{\omega_i - \omega_o}{b}))$
Voltage Controlled Oscillator
$\sin(\phi_o(t))$
ω_o
$\Delta\omega$
$G(s)$
$(\frac{1}{b} - \frac{1}{a})(\omega_i - \omega_o)$
ATAN2*
$A^2\sin((\frac{1}{b} - \frac{1}{a})(\omega_i - \omega_o)) = r_1 g_2 - g_1 r_2$
$A^2\cos((\frac{1}{b} - \frac{1}{a})(\omega_i - \omega_o)) = g_1 g_2 + r_1 r_2$
ω_o (initial guess)

Choose $a > b$

*ATAN2 is a FORTRAN arctangent function which keeps tracking the correct quadrant of the angle.

Fig. 3d.-8a The block diagram of the Frequency-Locked-Loop.

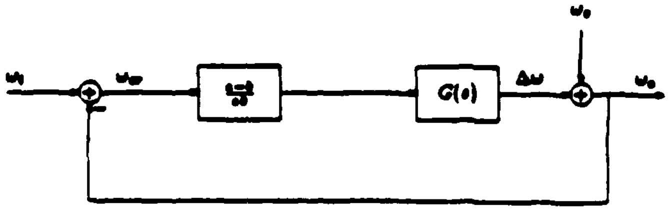


Fig. 3d.-8b The functional block diagram of the Frequency-Locked-Loop.

it can update the output frequency of the FLL according to its error ω_{er}. The most easy way is to use an integrator G/s and the FLL will have a pole at $-K$ where

$$\frac{K = G(a-b)}{ab\ .}$$

Parameters a and b should be determined with the following restriction

$$\omega_n > a > b > /\omega_i - \omega_o/$$

where ω_n is the nominal frequency the FLL is working at, and $/\omega_i - \omega_o/$ represents the working range in frequency.

With the parameters chosen as above, the block diagram in Fig. 3d.-8b can be expressed with the transfer function

$$\frac{\Omega_o(s)}{\Omega_i(s) = \frac{K}{s+K}\ .}$$

For example, a set of values are chosen for the parameters in a design as

$$a = 6.0 \quad , \quad b = 4.0 \quad , \quad G = 20.0 \quad \Longrightarrow \quad K = 1.67\ ,$$

where the nominal working frequency is

$$\omega_n = 13.3 rad/sec$$

and the desired linear searching range is

$$\frac{/\omega_i - \omega_o/ = \omega_n}{4 = 3.3 rad/sec\ .}$$

Fig. 3d.-10 shows the tested result of the FLL's output when the frequency of the input signal is changed suddenly. The response for a small input change (the first change in Fig. 3d.-10a) is similar to the step response of a first-order filter with a pole at $-K$, as shown in Fig. 3d.-10b. The response for a large input change which is much larger than the designed linear searching range experienced some nonlinearity at the beginning because the FLL's internal structural is not linear; however, the FLL still tracked the input signal and provided the correct output in a reasonable time.

The almost linear behavior of the FLL over a large working range makes it very suitable for frequency identification in real-time adaptive controls.

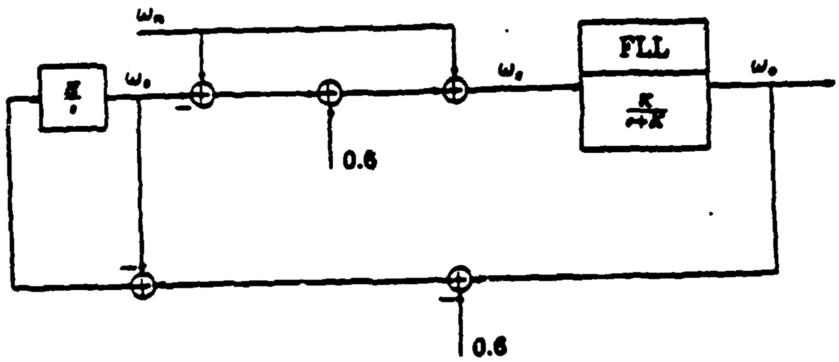


Fig. 3d.-9 The closed-loop dynamics of the parameter variation, identification, and correction of the entire system.

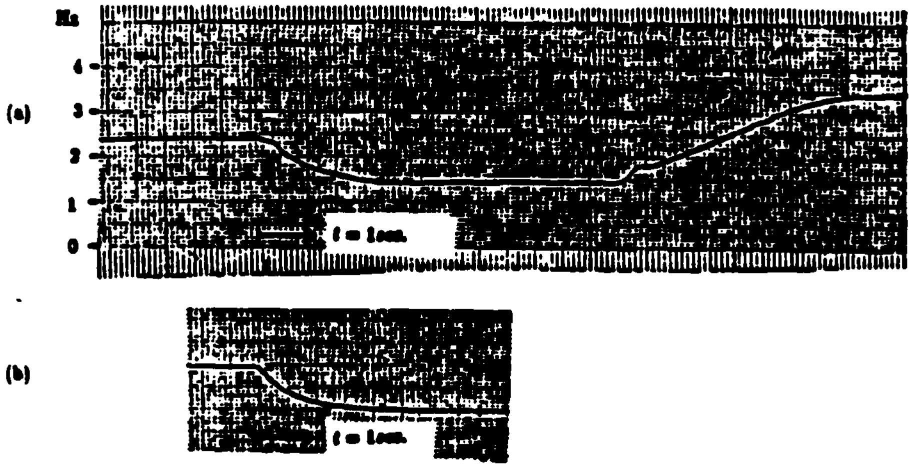


Fig. 3d.-10 **Step response of the FLL (a), compared with that of a first-order filter (b) with the same pole location $s = -1.67sec^{-1}$.**

Using the FLL in an Adaptive Closed-Loop Controller: Experimental Demonstration

The feasibility of using the FLL in adaptive control has been demonstrated in an adaptive closed-loop controller for another mechanical system built in the Stanford Aeronautics and Astronautics Robotics Laboratory. The system consists of two circular disks connected by a torsional rod. The system has inherently extremely low damping, and the actuator (DC motor) and the sensor (RVDT) are mounted on different disks. The natural frequency of the system is 2.11 Hz ($13.3 rad/sec.$) and the damping of that mode is less than 0.004. A high performance controller is designed using LQG algorithm which synthesize a state estimator for state feedback control. The value of the natural frequency is an important parameter in implementing the state estimator, and it is the one to be updated by the adaptation algorithm. An off nominal value of the natural frequency can be used for the initial controller design to test the performance of the adaptation algorithm.

The commanded system output is converted to its corresponding state command x_c, which is then compared with the estimated state $\hat{x}$ to calculate the feedback control effort u. For parameter identification, in principle, any signal containing the vibration mode to be identified can be used to drive the FLL, but it is found experimentally that the estimation error of the system output defined as

$$\tilde{y} = y - \hat{y}$$

is the best one to use since it will not be disturbed by any command change if there is no modeling error in the state estimator. The output of the FLL is compared with the value of the natural frequency used in the state estimator, and the error in the latter is corrected according to a parameter updating algorithm.

The dynamics of the parameter variation, identification, and correction interact with each other when the adaptation loop is closed. The parameter dynamics of the entire system can be expressed as the block diagram shown in Fig. 3d.-9, where ω_n is the open-loop natural frequency of the plant, ω_e is the corresponding value used in the state estimator, ω_c is the most critical closed-loop modal frequency, and ω_o is the output of the FLL. The relationship between ω_c, ω_n, and ω_e can be obtained by analyzing the closed loop system for its modal frequencies. The relation is then linearized to simplify the design for the parameter adaptation algorithm. The difference between ω_o and ω_c is used to correct the error of the parameter, ω_e, in the controller through an integrator, and this parameter updating scheme is designed using a classical control design method, namely the root locus method, because the FLL's behavior is almost linear.

The value of K in the FLL is chosen to be $1.67 rad/sec$ as described before; and the integration constant H is chosen to be 9.9 which results in the fastest response of the parameter correction loop with closed loop poles at $s = -5.8 rad/sec$. The detail of the analysis and design of the adaptive control system is described in a previously published paper by the author. The ratio of the value of the adaptation pole ($-5.8 rad/sec$) to that of the natural frequency of the uncontrolled mechanical system ($13.3 rad/sec$) is so high that the adaptation speed of this algorithm can be considered as very fast. In principle, other linear control design algorithms can also be used to design the parameter updating

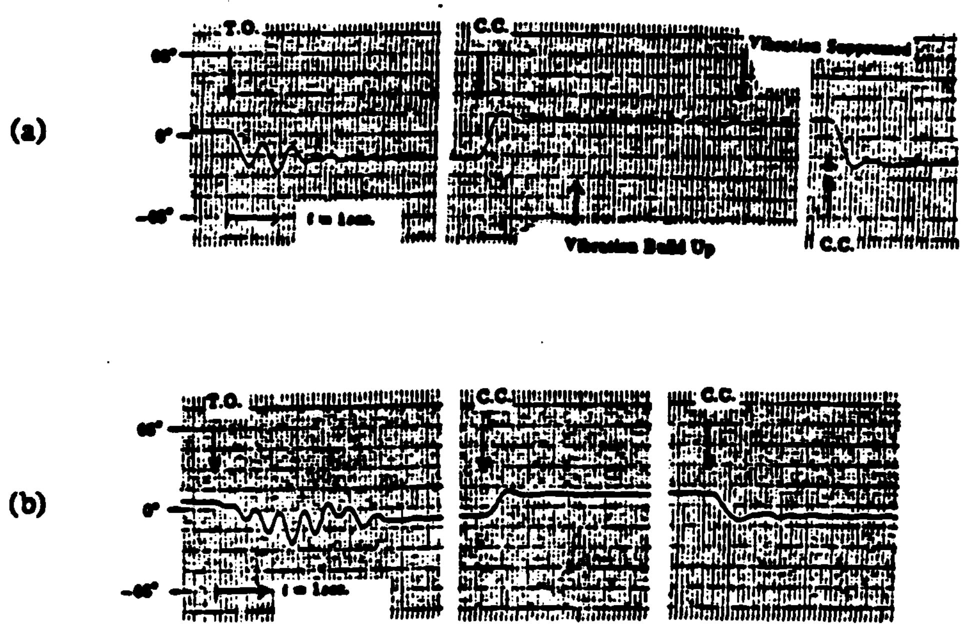


Fig. 3d.-11 Step response of the adaptive controller with FLL detecting initial modeling error in plant frequency; (a) +25% error and (b) -50% error.

scheme, as long as the FLL behaves almost linear.

Figure 3d.-11 shows the tested responses of the mechanical two-disk system under the adaptive closed-loop control with significant initial parameter error in the controller.

The control system becomes unstable if the modeling error in plant's natural frequency is more than 10%. Figure 3d.-11a shows that the system is initially unstable and starts oscillating when the control system is just turned on. However, the adaptation algorithm starts working immediately and stabilizes the system within two cycles of oscillation. The responses for the subsequent step commands look almost identical to those when using a fixed controller with no modeling error. The adaptation takes effect in less than one second on a lightly damped mechanical system which might be the first one, with time constant shorter than one second, having been successfully controlled by a fast adaptive system.

It can be noted from Fig. 3d.-11a that a small amplitude vibration builds up after the system becomes stable. This phenomenon can be attributed to the lack of signal to excite the FLL, and causing its output to drift away. However, the FLL catches the vibration, corrects the parameter error and suppresses the vibration before the instability has the chance to grow up. This interesting behavior can be related to the "minimum excitation" required by other types of adaptive algorithm.

Fig. 3d.-11b shows the test when the initial modeling error is 50% in the plant frequency. Beside making the initial control system unstable, the large parameter error also exceeds the linear design range of the FLL. The test shows that the frequency identification and the parameter adaptation scheme still work; it only takes a little more time to stabilize the system. The responses for the subsequent step commands show no sign of the initially large modeling error.

Judging from the success of the adaptive closed-loop control system for the two-disk mechanical system, we can envision that the FLL will be able to identify the vibration frequency and drive an adaptive filter/controller adequately for a wrist-beam system.

PUBLICATIONS

Ph.D Theses

[Chi T] Chiang, Wen-Wei, "Rapid, Precise End-Point Control of a Wrist Carried by a Flexible Manipulator," Stanford University, SUDDAR 550, January, 1986.

[Gar T] Gardner, Bruce, "Feedforward/Feedback Control Logic for Robust Target-Tracking," SUDDAR 537, December, 1984.

[Ly T] Ly, U., "A Design Algorithm for Robust, Low-Order Controllers, " Stanford University, SUDDAR 536, November 1982.

[Map T] Maples, James, "Force Control of Robotic Manipulators with Structural Flexibility," SUDDAR 549, June, 1985.

[Ros T] Rosenthal, Daniel, "Experiments in Control of Flexible Structures with Uncertain Parameters," SUDDAR 542, June, 1984.

[Sch T] Schmits, Eric, "Experiments on the End-Point Position Control of a Very Flexible One-Link Manipulator," SUDDAR 548, June, 1985.

[Wein T] Weinreb, Abraham, "Optimal Control with Multiple Bounded Inputs," SUDDAR 544, October, 1984.

Other Publications

Cannon, R. H., "Robots with a Light Touch" (Keynote Address to the 1983 Automatic Control Conference), *IEEE Control Systems Magazine,* May 1984, v4, No. 2.

Cannon, R. H., "Freeing the Space Station Crew for Discovery, Commitment to Automation," Editorial in Aerospace America, January, 1986, v24, No. 1.

Chiang, W. and Cannon, R. H., "The Experimental Results of a Self Tuning Adaptive Controller Using Online Frequency Identification," *The Journal of the Astronautical Sciences,* Jan.-March, 1985, v33, No. 1.

Ly, U., Bryson, A. E.,and Cannon, R. H., "Design of Low-Order Compensators Using Parameter Optimization," *Automction, Journal of the International Federation of Automatic Control,* 1985, v21, No. 3, pp. 315-318.

Rosenthal, D. and Cannon, R. H., "Experiments in Control of Flexible Structures with Noncolocated Sensors and Actuators," *Journal of Guidance, Control and Dynamics,* Sept.-Oct., 1984, v7, No. 5.

Schmits, E. and Cannon, R. H., "Initial Experiments on the on the End-Point Control of a Flexible One Link Robot," *International Journal of Robotics Research,* Fall, 1984, v3, No. 3.

TECHNICAL REPORT ON TASK 4:
INTEGRATED TACTILE SENSORS

Objective, Design Goals, and Development Plan

The overall objective of this Task was to explore the use of an integrated circuit pressure sensor as the basis of a tactile sensor for robotics applications.* A tactile sensor is a contact sensor which provides a spatial array of force information. It is anticipated that such sensors will find many applications in the robotics perception of the shape, orientation, texture, and slip of objects.

Most arrays feature random data access. At its simplest, the number of data lines must equal the sum of N columns and M rows plus one. In more complicated configurations, address bus and decoding circuitry allow a data bus for a reduction in sensor connections [Raibert]. Wider busses and increased power dissipation in larger arrays, however, will both limit spatial resolution and reduce reliability. In both cases, additional electronics are required to scan the array and must incorporate an algorithm for the detection of and recovery from defective sensing elements.

Some quantitative specifications for the performance of a tactile array were available in the literature [Harmon]:

1. spatial resolution	2mm
2. array dimensions	10 x 10
3. response time	0.01 - 0.1 s
4. dynamic range	¿500

In addition, for use in the industrial environment, a tactile sensor is required to be:

5. immune to electromagnetic interference
6. insensitive to temperature
7. mechanically rugged
8. stable
9. easy to calibrate

These requirements led to a number of decisions regarding the design goals for the Task. For a dynamic range ¿500 in pressure, a capacitive rather than a piezoresistive transduction technique was selected due to its much greater sensitivity. For immunity to electromagnetic interference, a digital rather than an analog data format was selected. To permit real-time external compensation for manufacturing variabilities and temperature effects, it was decided that the sensor should provide temperature, zero-offset, and gain in addition to pressure information. To maximize reliability, the number of electrical connections to the array was to be minimized. This required that the output signals of the

* *Refer to Section B1, r. of the Contract Statement of Work, which states: "Investigate the use of silicon integrated circuit technology to achieve ultraminiature pressure sensors."*

various pressure sensing elements in the array be multiplexed onto a single serial output line. For 10 x 10 array dimensions and a response time of 0.01 - 0.1 s, therefore, each pressure-sensing element in the array had to update its output signals each 0.0001 to 0.001 s. Finally, to ensure stability of the sensor, it was to be hermetically sealed and comprised of the minimum number of parts.

The Task was divided into three subtasks to be conducted in two stages. The subtasks were:

1. an integrated circuit which would generate pressure and calibration signals. This circuit, which was to transduce capacitance to an asynchronous digital output, was termed a Capacitive Pressure Transducer (CPT);
2. an integrated circuit which would be the array system manager. This circuit, which would multiplex signals from the various array elements, detect and recover from non-functional elements, etc., was termed an Asynchronous Digital Multiplexer (ADM);
3. micromachining. This subtask was to be concerned with the fabrication of integrated pressure-sensitive capacitors, hermetic sealing, electrical contract through the hermetic encloser, etc.

The two stages of the development plan were:

1. separate development of the subtasks; and
2. merger of the subtasks into a functional array, including
   a. merger of the micromachining and CPT subtasks into a functional pressure sensor; and
   b. merger of ADM and many pressure sensors into a functional tactile array.

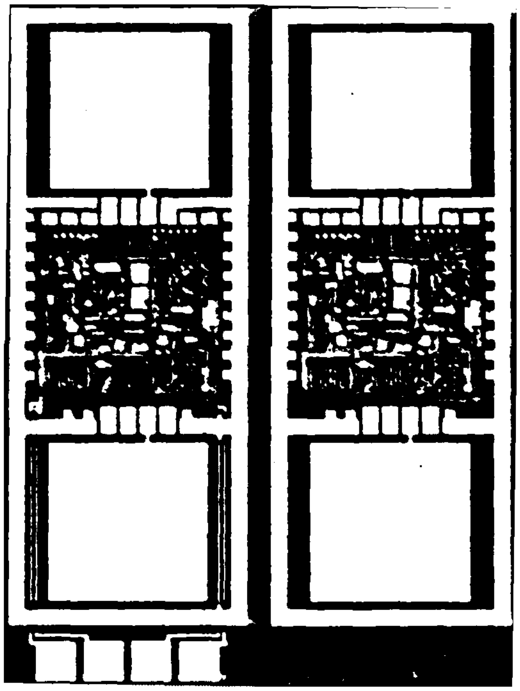
During this funding period, only goals within the first stage were achieved.

Status of the Research Effort

Subtask 1: An Integrated Circuit for a Capacitive Pressure Transducer

The goal of this subtask was to construct a custom integrated circuit for a small, accurate, low-power capacitive pressure transducer which is the key element of a pressure-sensing array.

This subtask has been completed during the current funding period, culminating in the successful demonstration of the custom integrated circuit for use with a capacitive pressure transducer [Smith, May 1985]. A photomicrograph of several of the integrated circuits appears in Figure 4-1.



System Architecture

Figure 4-2 illustrates the block diagram of the integrated circuit. The component subcircuits include an integrated bandgap temperature reference, a Schmitt trigger oscillator, and digital logic. The multiplexed asynchronous digital data format appears in Figure 4-3. The ratiometric calibration scheme which utilizes zero-offset and gain information for the measurement of pressure-sensitive capacitance and temperature are illustrated in Figure 4-4 and 4-5.

System Performance

Results taken from 6 chips show the accuracy of pressure-sensitive capacitance and temperature. Figure 4-6 illustrates the demodulated temperature signal as a function of temperature. Note that the temperature sensitivity of the circuit is almost precisely the theoretical 3333 ppm/degC. Figure 4-7 shows the pressure signal period as a function of capacitance, temperature, and supply voltage. The pressure sensitivity is 3300 ppm/mmHg.

Since zero-offset and gain and effects on accuracy are entirely eliminated by the ratiometric measurement scheme, accuracy is limited only by errors in capacitance and temperature measurements.

Subtask Status

The integrated circuit layout contains all the features necessary to construct a monolithic transducer. The first batch fabrication run containing 135 transducers is currently in process in the Stanford Integrated Circuits Laboratory. These devices will be tested at the circuit level before being packaged as transducers.

In addition, components of a precision calibration apparatus for pressure and temperature were acquired. This apparatus is in the process of being assembled and tested.

Subtask 2: An Integrated Circuit for Tactile Array System Management

Our research in tactile arrays has lead us to a novel system architecture. The custom Asynchronous Digital Multiplexer (ADM) integrated circuit is an implementation of this new architecture. It is a first and necessary prototype to test our methods that will be extended to produce tactile arrays.

System Architecture

The ADM combines the output data signals, emitted by our tactile sensors, onto a single pair of wires [Shapiro]. In addition to the data signal, each pressure sensor also emits a sync signal to serve as a data frame marker.

The ADM operates by using this sync information from the sensors to control the multiplexing of their data signals. This two-tiered multiplexing scheme allows any form of pulse modulation to be used by the sensor as long as the data stream is properly framed by a sync signal. Because the multiplexing rate is determined by the repetition rate of the sync pulses, the sampling frequency will be determined by the particular sensors used and

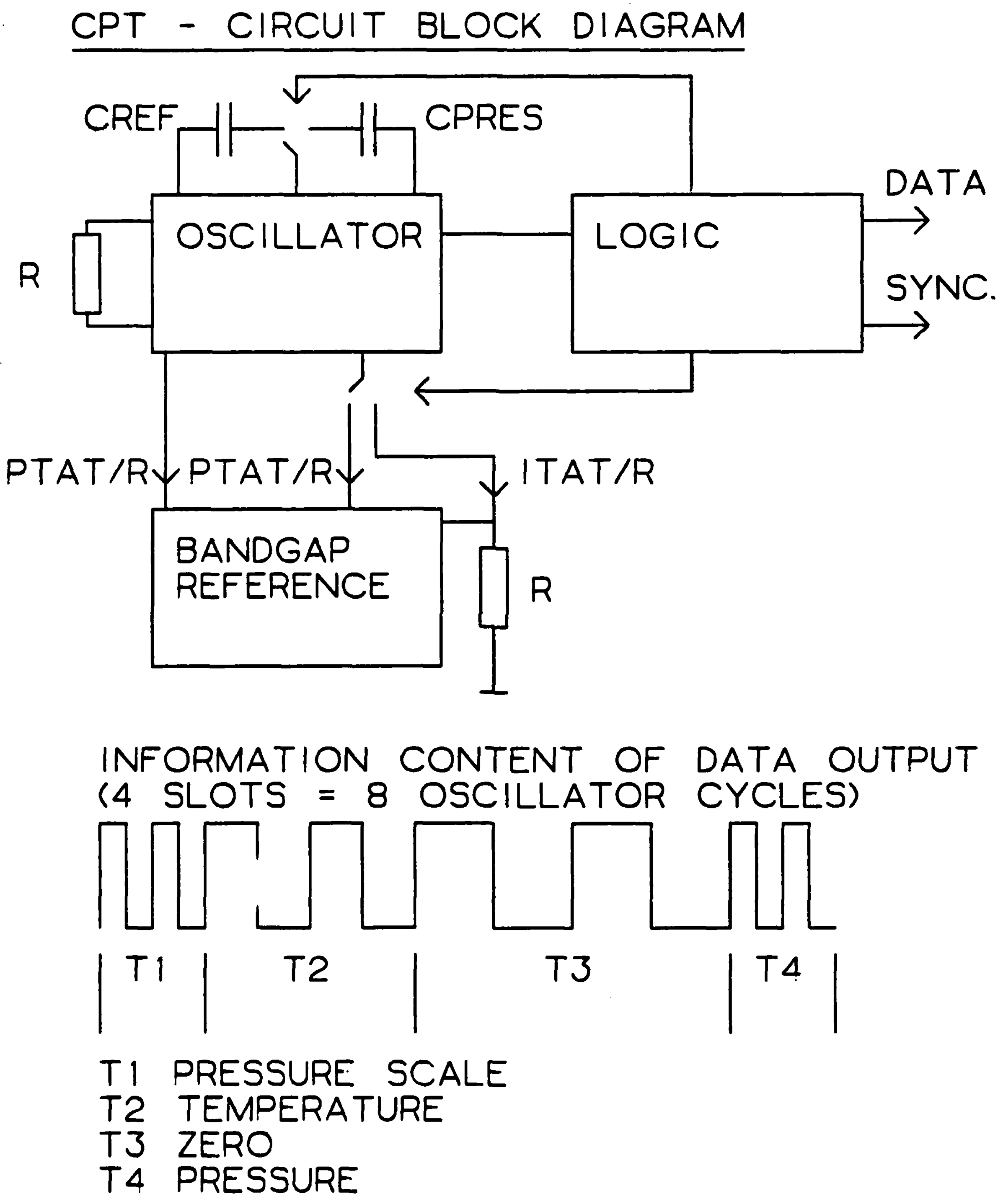


Fig. 4-2 Block Diagram of the Capacitive Transducer IC

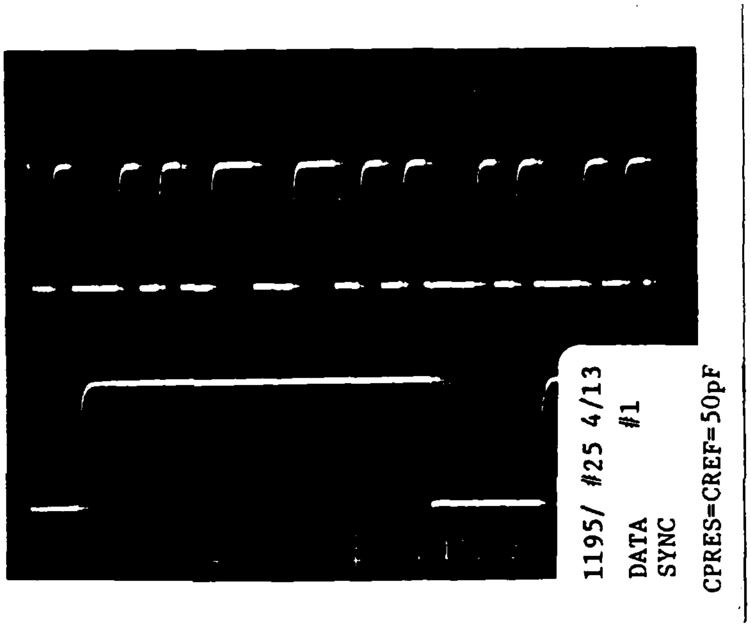


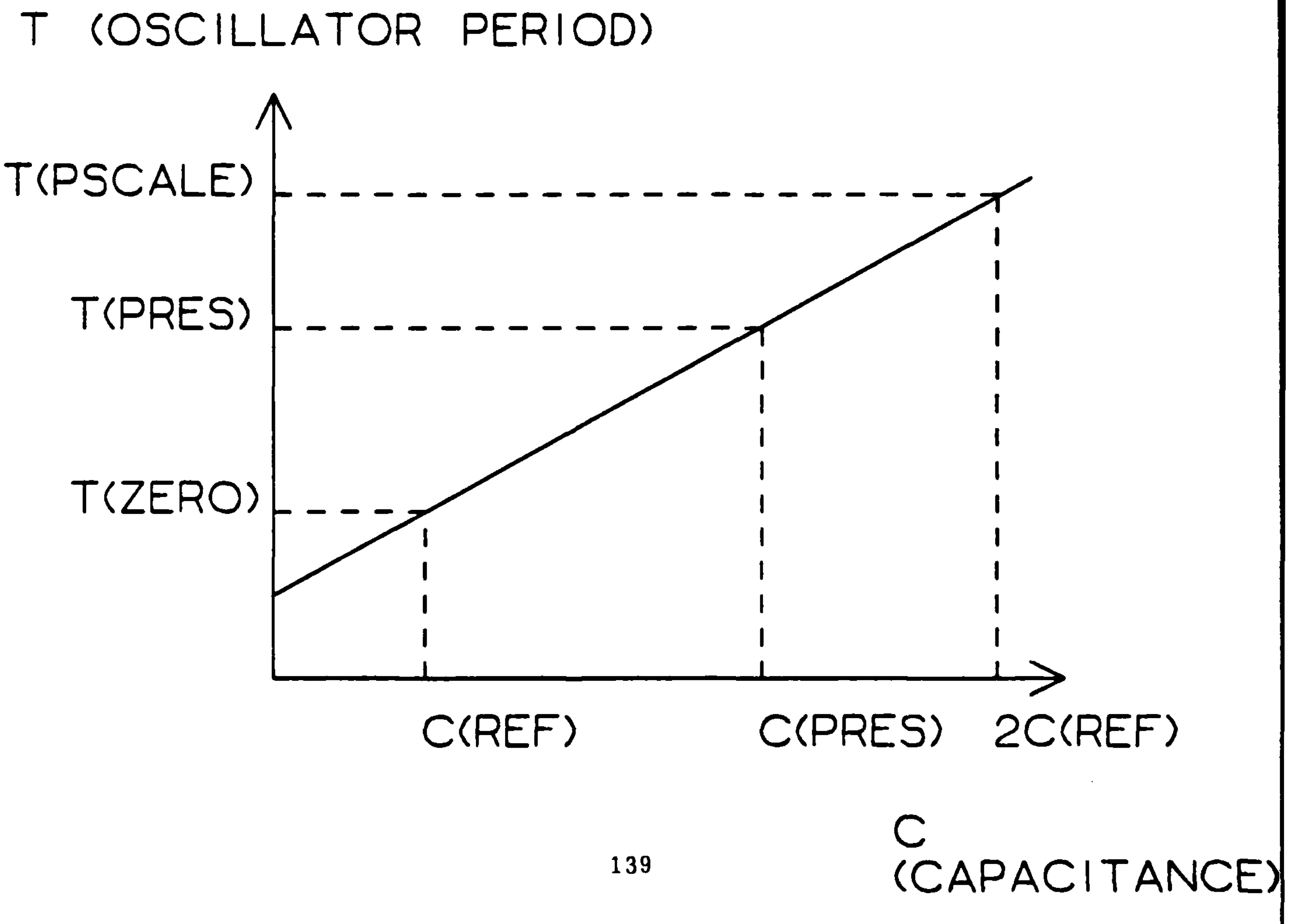
Fig. 4-3 Data Format of the Capacitive Transducer IC

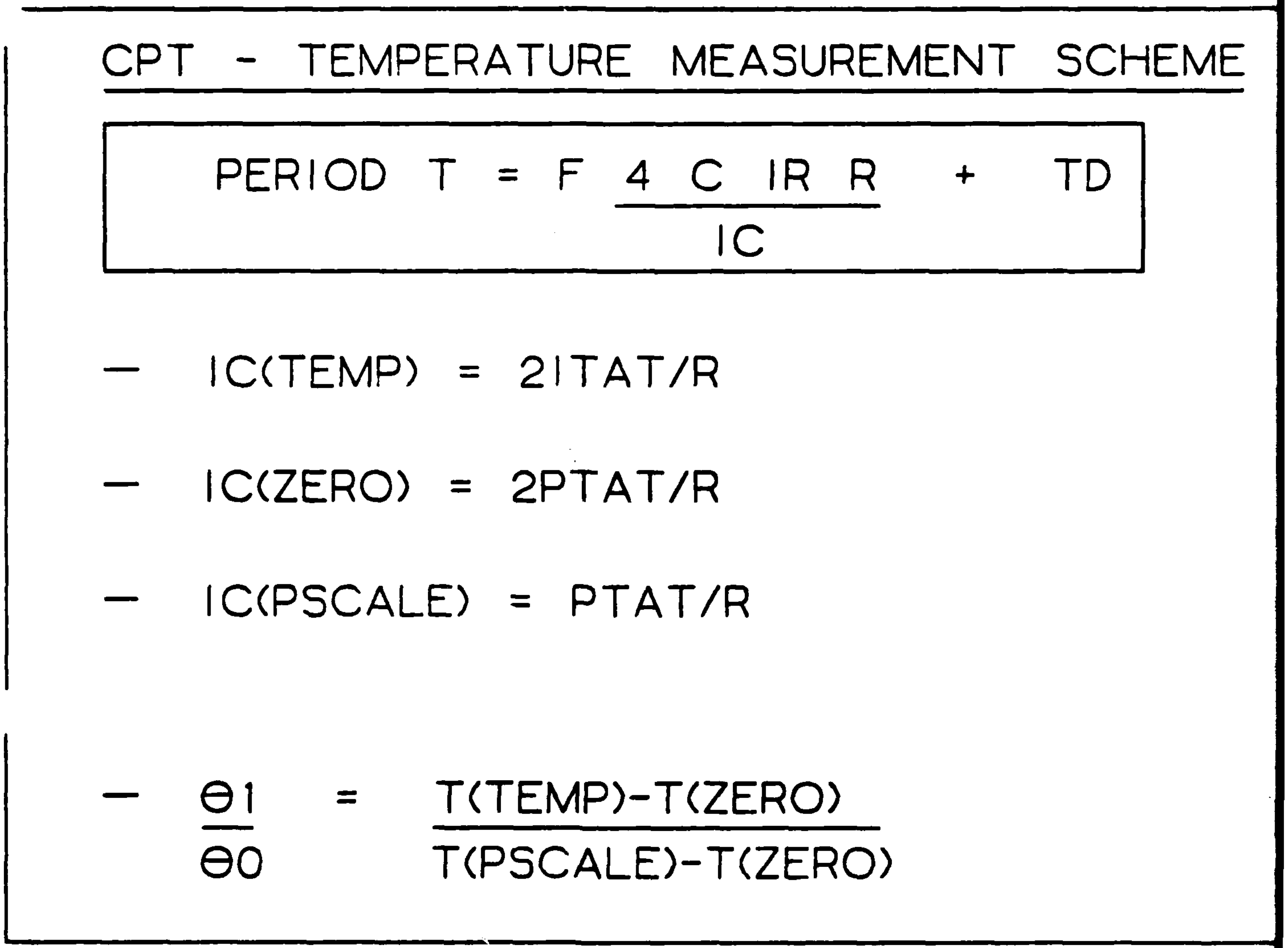
Fig. 4-4 Ratimetric Capacitance Measurement Scheme of CPT IC

CPT - PRESSURE MEASUREMENT SCHEME

$$\text{PERIOD } T = f\,\frac{4\ C\ IR\ R}{IC} + TD$$

- T(PSCALE) = 8 f CREF R + TD IC=PTAT/R
- T(PRES) = 4 f CPRES R +TD IC=2PTAT/R
- T(ZERO) = 4 f CREF R + TD IC=2PTAT/R





CPT - TEMPERATURE MEASUREMENT SCHEME

$$\boxed{\text{PERIOD } T = F\,\frac{4\ C\ IR\ R}{IC} + TD}$$

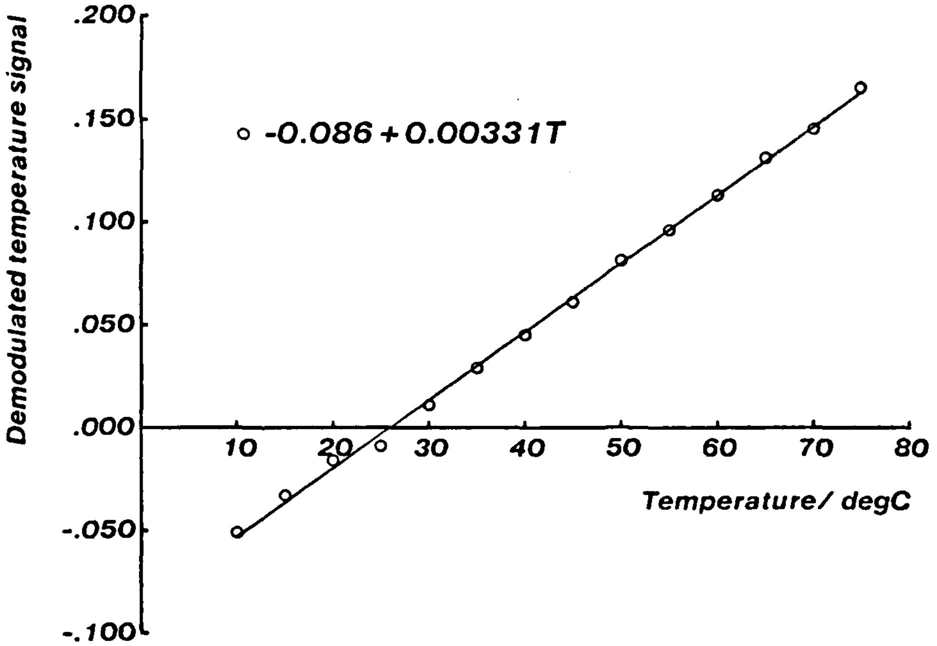
- $IC(TEMP) = 2ITAT/R$

- $IC(ZERO) = 2PTAT/R$

- $IC(PSCALE) = PTAT/R$

- $\frac{\Theta 1}{\Theta 0} = \frac{T(TEMP)-T(ZERO)}{T(PSCALE)-T(ZERO)}$

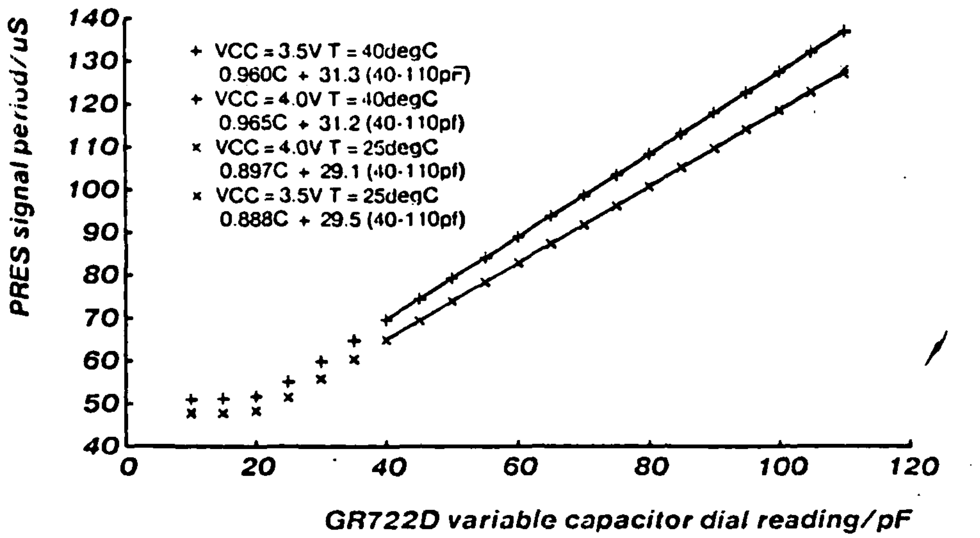
Fig. 4-5 Ratiometric Temperature Measurement Scheme of CPT IC



Demodulated temperature signal vs. temperature

Wafer 1195/5a#19, VCC = 3.75V
Measured using TEK2465
Hot chuck in heat/cool mode
plottempmeas1195-5a-19, 4/8/85

Fig. 4-6 CPT IC Demodulated Temperature Signal vs. Temperature



PRES output signal period vs capacitance

Measured using GR722D, HP5370(10k samples), CREF = 45pF
Wafer 1195/5a#19
plotcaplinearity1195-5a-19, 4/14/85

Fig. 4-7 CPT IC Pressure Signal Period vs. Capacitance, Temperature and Supply Voltage

not by the ADM. A handshake signal for each sensor permits uniform sampling with the synchronization of the sensors. T. is same signal also can be used to initiate power to the sensor in pulse-powering applications where system current drain must be minimized. A block diagram of the ADM integrated circuit appears in Figure 4-8.

A potential disadvantage of a self-clocked multiplexer is that if any sensor fails, the counter in the control logic stops and the ADM doesn't advance to the next channel. To avoid this, a special circuit monitors the sync signals emitted by the sensors and inserts a pulse when one is missing to advance the counter. An added benefit of this circuit is that it aids the start-up of the ADM when the IC is powered. In addition, the ADM logic is always properly initialized on start-up because of the recursion that is inherent in the system architecture: the sensor that is being multiplexed is the same sensor that is clocking the multiplexer.

To test our new architecture, a gate-level logic simulation of the ADM digital design was performed using SALOGS, a program developed by Sandia for the Department of Energy. Four free-running sensors, each with 3 channels of data, were connected to the ADM and the logic design was verified.

ADM Circuitry

The ADM circuitry consists of three main components: a data multiplexer, sync multiplexer and control logic. The sync multiplexer and control logic need not be high speed because both circuits are operated at about 3.33kHz (300us) — the frame rate of the data from the tactile sensor. The data multiplexer must be high-speed, however, to preserve the purity of the pulse edges so that the large dynamic range achieved by the pressure sensor array elements is maintained. Although the repetition rate of the pulses from the pressure sensors is relatively slow (a period of 50us), an RMS jitter of only 15ns would reduce the dynamic range to 300ppm. Because of the special speed and power requirements of the ADM, both I2L and EFL bipolar logic is used. EFL with its multi-emitter input transistor is a natural choice for the high-speed data multiplexer. A simple input stage that connects each sensor data signal to the EFL multiplexer was developed. This input stage interfaces easily with the EFL multiplexer and has an input threshold of about 1.2V. A high-speed low-power driver was designed to connect the output of the EFL multiplexer with the output buffer. A special buffer with bootstrap circuitry has been developed to enable the ADM to drive the capacitance associated with an output cable. A simple bias circuit for the EFL logic generates voltage levels and with tracking current source for improved temperature stability.

The sync multiplexer is implemented in I2L because of the low speeds of the sync signals. The input stage developed is simple and interfaces easily to pulse synchronizers which are required when the sensors are free-running, in the case when handshaking is unavailable. Multiplexing is easy because of the inherent wired-AND in the I2L gate: the outputs from the pulse synchronizers need only be bussed to the input gate of the control logic. The multiplexed sync signals are connected to the missing pulse circuitry that prevents lock-up due to faulty sensors. Missing pulse detection is accomplished with a resettable astable.

Two preliminary versions of the counter in the control logic have been designed. A

Asynchronous Digital Multiplexer (ADM)

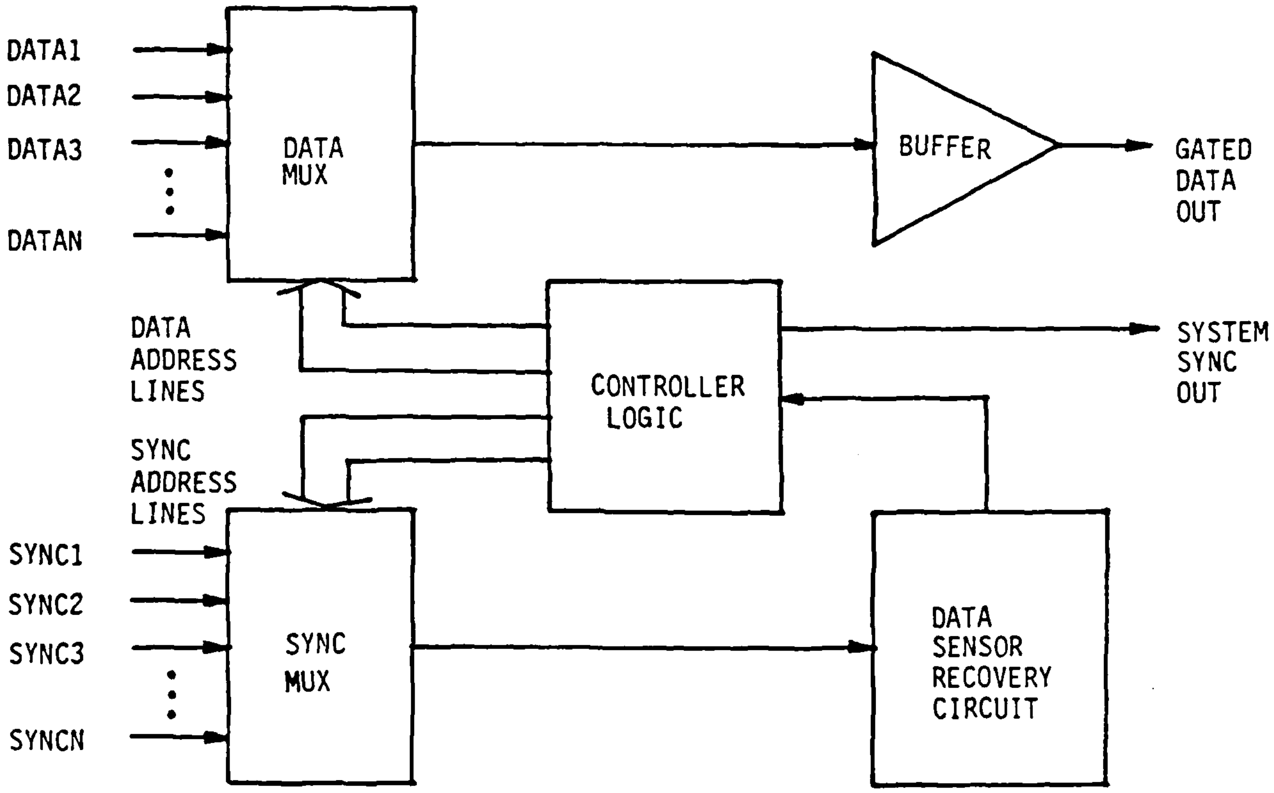


Fig. 4-8 Block Diagram of the Asynchronous Digital Multiplexer IC

programmable version, to insure flexibility, has been designed so that the ADM can multiplex 2, 4, 6, or 8 sensors. An additional feature has been implemented to allow sampling of low-bandwidth signals at a fraction of the rate required for the regular data signals. A simpler non-programmable version has also been designed. When implemented in a tactile array, the counter will be appropriately extended for the array dimensions.

A I2L-to-EFL level-shift circuit, with minimal standby current drain, has been developed to interface the control logic with the data multiplexer. Similar circuitry also is used to drive the output buffers that emit the handshake signals from the I2L control logic.

Subtask Status

The majority of the circuit simulation was performed using HSPICE, donated by Meta-Software of Cupertino. Each of the above mentioned circuit blocks, with the exception of the missing pulse detector, have been simulated successfully. At the time of the writing, all the blocks are being connected together in order to simulate the complete chip: a two channel version of the sync multiplexer has been simulated with the data multiplexer soon to be added. The missing pulse detector must then be completed and added into the ADM simulation.

Layout and fabrication of the ADM integrated circuit follow the completion of the design and simulation of the circuitry. The circuitry will be fabricated using the Standard Bipolar Process, developed by the IC Lab at Stanford University.

As mentioned earlier, the ADM is only a first step in the development of larger tactile arrays for robotics. A generalization of the architecture results when we repartition the ADM. If we use tri-state or open collector outputs to multiplex both the data and sync signals, we can remove these functions from the ADM chip. The control logic also can be partitioned into separate pieces that can be incorporated into each pressure-sensing array element. To do this, we take advantage of the sequential nature in scanning the sensors: we arrange the sensors into a shift-register or "bucket brigade" configuration so that each sensor tells the subsequent one when to power up and start sensing. A common clock line advances the shift-register. With serial rather than random access to each individual sensor in the array, the number of wires required for electrical connection to the array is independent of the array dimensions and reduced from $N + M + 1$ to 2.

Many key ideas in the ADM and the repartitioned ADM are the same. Both feature self-clocked multiplexing, missing-pulse detection of defective elements, and pulse-powering capability. The differences are that the data multiplexer is now accomplished with a wired-AND connection and control logic from CPT and ADM is combined and simplified.

Another potential disadvantage of this architecture is that a break in any of the bussed lines would disable the array. A random access array with address bussing, however, suffers a similar problem and a simple array could still lose a complete row or column of sensors. This potentiality, however, is common to all bussed integrated circuits and rarely presents a problem. The likelihood of this problem is less in rigid arrays, such as the one being developed here, than in flexible arrays.

Subtask 3: Micromachining

The structure of the capacitive pressure transducer prototype for the tactile array element appears in Figure 4-9. Special micromachining techniques were required for forming this three dimensional structure, including chemical etching of a pressure sensitive diaphragm and of a shallow well in glass for and electric shield over the integrated circuit, laser drilling of electrical contact vias through glass, and electrostatic bonding of glass and silicon.

Laser Drilled Electrical Contact Vias through Glass

The conventional way to protect integrated circuits from the corrosive attack of the environment is by means of hermetically sealed packages. An integrated sensor, however, must be exposed to the environment in order to perform its transduction function. For integrated sensors, electrostatic sealing glass and silicon wafers is a demonstrated technique for hermetically protecting on-board circuitry while exposing an on-board pressure transducer [Sander]. Hermetically enclosing a circuit, however, creates the problem of making electrical contact to it. We chose to drill electrical contact vias through the glass wafer.

An automated carbon dioxide laser workstation was designed and built for the purpose of drilling vias through glass wafers. The workstation interfaced a microcomputer to an X-Y Table and to a carbon dioxide laser. The X-Y Table has a spatial resolution of 5 um and the laser has a focussed spot size of 50 um. This workstation was then programmed to produce patterns of holes through the glass wafers which mated with patterns of electrical contact pads on underlying silicon wafers.

Conical vias with an upper diameter of 300 um and a lower diameter of um were produced. Metalization of these holes produced electrical contact with resistances of less than 10 ohms. [Bowman, 84]. It was determined that the following procedures were essential to achieving a low resistance electrical contact:

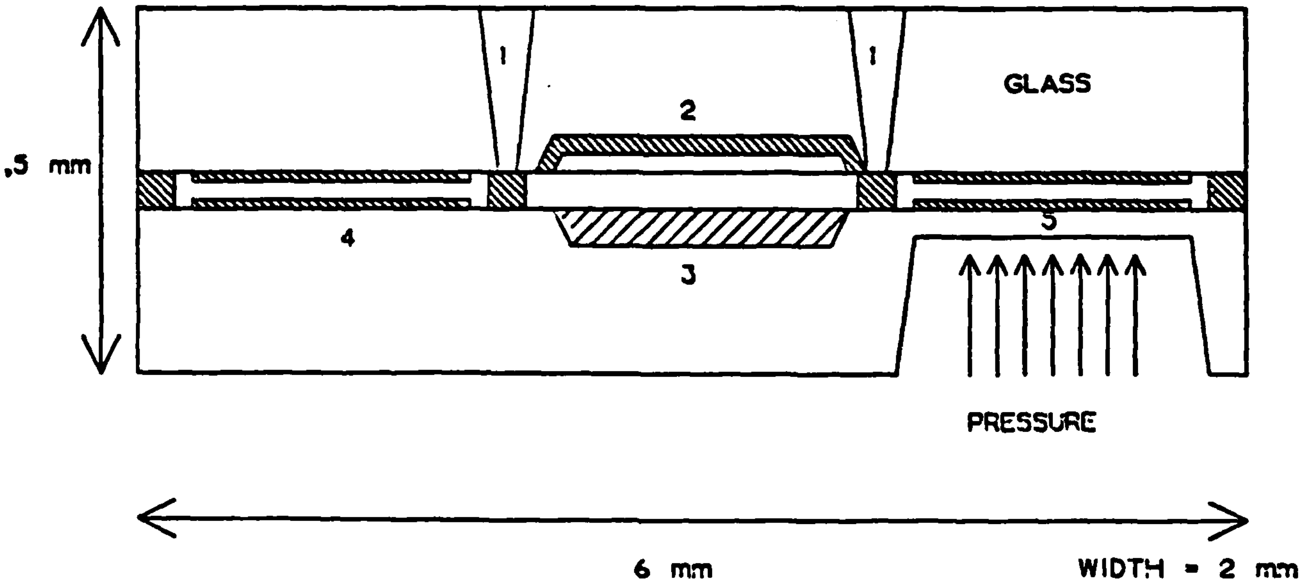
1. electrostatic sealing of the glass and silicon wafers before laser drilling;
2. drilling through the glass with many small laser pulses rather than with one large pulse;
3. sculpting the contour of the drilled hole by moving the workpiece under the laser during drilling; and
4. depositing metal in the drilled hole by sputtering. A thickness of 3 or more um of aluminum was adequate.

Chemical Etching of Pressure Sensitive Diaphragms

The anisotropic etching of pressure sensitive diaphragms is performed extensively commercially. Little information was available, however, concerning the repeatability of the etching process. Processing information from a large number of diaphragms in a tactile array would be greatly simplified if the etch rate were uniform across a wafer and from wafer to wafer.

An experiment was performed to determine the uniformity of silicon etch rate in KOH.

CPT - CROSS SECTION



1 - ELECTRICAL CONTACT VIAS
2 - ELECTROSTATIC SHIELD
3 - CUSTOM INTEGRATED CIRCUIT
4 - REFERENCE CAPACITOR
5 - PRESSURE SENSING CAPACITOR

Fig. 4-9 Structure of Prototype Capacitive Pressure Transducer

V-grooves of various depths were etched and their depths measured. It was determined that the variation in depth was 0.28 um in a mean depth of 40 um. Projected to a total depth of 140 um as would be used in a functional pressure sensor or tactile array, the variation in diaphragm thickness would be less than 1 um. Since the diaphragm thickness itself would be approximately 60 um, this variation in thickness would have little influence upon sensor sensitivity.

Subtask Status

The merger of the micromachining and CPT subtasks into a functional sensor remains to be accomplished.

Related Publications

Bowman, L., Schmitt, J.M., and Meindl, J.D., "Electrical Contacts to Implantable Integrated Sensors by CO_2 Laser-drilled Vias Through Glass", Workshop on Micromachining and Micropackaging of Transducers, Cleveland, Ohio, November 7-9, 1984.

Smith, M.J., and Meindl, J.D., "Integrated Circuits for a Capacitive Pressure Transducer", IEEE Frontiers in Engineering in Health Care Conference, Los Angeles, Sept., 1984.

Shapiro, F.B., Shott, J.D., and Meindl, J.D., "A Custom IC for Multichannel Telemetry with Digital Sensors", Conference Proceedings, 6th Annual EMBS Conference, pp. 715-718, Sept., 1984.

Smith, M.J., Meindl, J.D., "Exact Analysis of the Schmitt Trigger Oscillator", IEEE Journal of Solid-State Circuits, Vol. SC-19, Dec., 1984.

Smith, M.J., Prisbe, M.A., Shott, J.D. and Meindl, J.D., "A Cusotm Analog IC for a Smart Pressure Sensor", IEEE Custom Integrated Circuits Conference, Portland, Oregon, May, 1985.

Smith, M.J., Prisbe, M.A., Shott, J.D., Meindl, J.D., "A Micropower IC for a Biomedical Pressure Sensor", Third International Conference on Solid-state Sensors and Actuators, Philadelphia, June, 1985.

Smith, M.J. and Meindl, J.D., "Analysis and Design of an IC Capacitive Pressure Sensor", accepted for publication in IEEE Transactions on Biomedical Engineering, Dec., 1985.

Smith, M.J. "An Integrated Circuit for a Biomedical Capacitive Pressure Transducer", Ph.D Dissertation, Stanford University, June, 1985.

Bibliography

Raibert, M. and Tanner, J., "Design and Implementation of a VLSI Tactile Sensing Computer", The International Journal of Robotics Research, Fall, 1982.

Harmon, L., "Automated Tactile Sensing", The International Journal of Robotics Research, Vol. 1, No. 2, Summer, 1982.

Sander, C.S., Knutti, J.W., Meindl, J.D., "A Monolithic Capacitive Pressure Sensor with Pulse-period Output", IEEE Transactions on Electron Devices, Vol. ED-27, pp. 927-930, May, 1985.

Appendix A
PERSONNEL

Principal Investigators:
Robert H. Cannon, Jr., Professor and Chairman, Dept. of Aeronautics/ Astronautics
Thomas O. Binford, Professor (Research), Dept. of Computer Science

Center Consultants
Daniel B. DeBra, Professor of Aeronautics and Astronautics
Rodney Brooks, Assistant Professor of Computer Science
Gordon S. Kino, Professor of Applied Physics
Gene F. Franklin, Professor of Electrical Engineering
Lambertus Hesselink, Assistant Professor of Aeronautics and Astronautics
Victor Scheinman, Vice President, Automatix, Inc.

Task Area 2 Research Team
Ulrich Rembold, Visiting Professor
Oussama Khatib, Research Associate
Yutaka Kanayama, Research Associate
Eric Pauchon, Visiting Scientist
Ernst Triendl, Research Associate
Junichero Takamura, Visiting Scientist
Ron Goldman, Post Doctoral Fellow
Leonie Dreschler-Fischer, Visiting Scientist
Ted Selker, Research Assistant
Raju G.V.S., Visiting Scientist
Richard Scott, Master of Science in Artificial Intelligence
Igal Tidhar, Engineer
Avraham Rousso, Engineer
Roy Fuller, Engineer
Ram Rubinstein, Engineer
Avrahain Rise, Engineer
Cregg Cowan, Research Assistant
H. David Goering, Research Assistant
Cary Gray, Research Assistant
John Clayton Hake, Research Assistant
Hong-Seh Lim, Research Assistant
David Chelberg, Research Assistant
Shashank Shekar, Research Assistant
Chun Shen Cai, Research Assistant
Pang Chen, Research Assistant
Glenn Healey, Research Assistant
D. Sathyanarayanan, Research Assistant
Ashok Subramanian, Research Assistant
Nancy Paulikas, Research Assistant
Craig Rublee, Research Assistant
Richard. L. Vistnes, Research Assistant
William Wells, Research Assistant
Mark Nakamura, Student Support Staff
Brad Chen, Student Support Staff
Amy Pearl, Student Support Staff
Wally Mann, Student Support Staff
Elmer Moots, Support Staff

Ernie Hine, Support Staff

Task Area 3 Research Team:

Gad Shelef, Research and Design Engineer
Tzoor Fridman, Research and Design Engineer
Wen-Wei Chiang, Research Assistant
Michael Grant Hollars, Research Assistant
Ross Koningstein, Air Force Training Grant Fellow
Ray Kraft, Air Force Training Grant Fellow
Arie Maharshak, Research and Development
Dennis Morse, TRW Fellow
Celia Oakley, NSF Fellow
Lawrence E. Pfeffer, Research Assistant
Daniel Rovner, Air Force Training Grant Fellow
Stanley Schneider, Research Assistant
Brian Anderson, Research Assistant
Avi Weinreb, Research Assistant
Warren Jasper, Research Assistant
Ellen Crippen, Research Assistant
Michael Sidman, Digital Equipment Corporation Fellow
Marc Ullman, Air Force Training Grant Fellow
John Wilson, Research Assistant

Task Area 4 Research Team:

Lyn Bowman, Research Assistant
Joseph M. Koepnick, Research and Development Engineer
Frederick B. Shapiro, Research Assistant
Michael J. S. Smith, Research Assistant

APPENDIX B

PART I - THE SCHEDULE

SECTION B - SUPPLIES/SERVICES AND PRICES

0001 1. RESEARCH

The contractor shall furnish level of effort specified in Section F, together with all related services, facilities, supplies and materials needed to conduct the research described below. The research shall be conducted during the period specified in Section F.

0001AA a. Conduct a survey to identify key problems in assembly, test and rework in production of airframe, avionics, and propulsion units.

b. Establish mechanisms to maintain a continuous process of technology transfer.

c. Provide a focus for research to solve key problems identified in a.

d. Develop components needed for flexible, integrated systems for programmable automation for assembly, test, and rework.

e. Develop Components needed for an intermediate integrated system for assembly.

f. Analyze selected, typical assemblies in AL, including airframes, propulsion systems, and avionics.

g. Design and implement additions to AL, to simplify programming and overcome limitations identified in task d.

h. Augment development of the portable AL programming system and aid in distribution of AL.

i. Develop preliminary specifications for a family of standardized electronic and software interfaces for programmable devices, especially robots.

j. Develop adaptive self-calibration and self-alignment modules which adapt to setup errors to cut setup time, modules which track variations and maintain calibration during production to improve reliability; develop optimization modules to move along the learning curve in speeding up assembly operations.

k. Develop ACRONYM connected to AL as an integrated assembly system with vision, force control, geometric modeling, manipulation, simulation, and planning.

The Board of Trustees of the

l. Develop visual control of assembly, using ACRONYM; use visual tracking of robot and parts.

m. Extend ACRONYM to aid in programming and integrating multiple test functions, and aid in integrating test with assembly.

n. Extend the end-point control laws developed for single link flexible arm operation to two link operation in the horizontal plane and investigate extension to the vertical plane dimensions where gravity affects the system.

A+A

o. Investigate the capabilities and limitations of lumped-compliance robot arms. Specifically, determine the slew speed, end-point settling time and positioning accuracy, proximity motion, and contact-force performance for a range of: flexibility, payload mass and inertia, actuator power, controller bandwidth; and study inherent actuator and sensor non-linearities.

p. Extend the work in tasks o and p to robot arms with distributed flexibility.

q. Investigate the performance to be expected of the complete flexible robot arm with end-effector for various real-life manufacturing, assembly, and inspection tasks.

Meindl

r. Investigate the use of silicon integrated circuit technology to achieve ultraminiature pressure sensors.

0002 REPORTS

Reports identified below shall be prepared in accordance with Exhibit A to this contract and delivered in accordance with Section F.

0002AA FUNDS EXPENDITURE AND PROJECTION REPORT

0002AB ANNUAL REPORT

0002AC FINAL REPORT

Other reports which are or may be required under this contract are identified in Sections H and I.

See Section H, Paragraph 3 for pricing information.

APPENDIX C

PREDICTING SPECULAR FEATURES

Glenn Healey and Thomas O. Binford

Artificial Intelligence Laboratory
Stanford University
Stanford, California 94305

Abstract

We show that highlights in images of objects with specularly reflecting surfaces provide significant information about the surfaces which generate them. A brief survey is given of specular reflectance models which have been used in computer vision and graphics. For our work, we adopt the Torrance-Sparrow specular model which, unlike most previous models, considers the underlying physics of specular reflection from rough surfaces. From this model we derive powerful relationships between the properties of a specular feature in an image and local properties of the corresponding surface. Careful experiments with specularly reflecting objects establish the merit of these relationships.

1. Introduction

Shiny surfaces give us specular reflections (highlights). A perfectly smooth shiny surface (e.g. a perfect mirror) reflects light only in the direction such that the angle of incidence equals the angle of reflection. For rougher shiny surfaces (e.g the surface of a metal fork), specular effects are still observable. In this paper we analyze the properties of specular reflection from rough shiny surfaces.

There are several basic reasons why the study of specular reflection deserves serious attention in computer vision. Specular features are almost always the brightest regions in an image. Contrast is often very large across specularities; they are very prominent. This makes them easy to locate. In addition, the presence or absence of specular features provides immediate constraints on the positions of the viewer and light sources relative to the specular surface. Also, as we will show, the properties of a specularity constrain the local shape and orientation of the specular surface.

An ability to understand specular features is valuable for any vision system which is required to analyze images of shiny objects. This work, for example, began as an attempt to allow ACRONYM [3] to reason about specular reflections from shiny mechanical parts in the ITA project [4]. Images of these parts typically contain large specular regions. The recognition task becomes considerably easier if the system is able to predict the characteristics of these specular regions.

In this paper we examine what information can be inferred from an image of a rough shiny surface by considering only the physics of specular reflection. Particular emphasis is placed on finding symbolic quasi-invariant relationships which will hold in many different situations (e.g. different source, viewer configurations). In contrast to many intensity-based vision algorithms, our relationships are based on the properties of a relatively large number of pixels in an image. This allows us to observe predicted features and infer local surface shape even in noisy intensity images or in cases where available specular models do not completely characterize the physics of specular reflection.

2. Review of Previous Work

Researchers in computer graphics have used increasingly realistic specular models. Several of these models will be discussed in the next section. In computer vision, however, relatively few attempts have been made to exploit the information encoded in specularities. Ikeuchi [9] employs the photometric stereo method [15] and uses a distributed light source to determine the orientation of patches on a surface. Grimson [8] uses Phong's specular model [10] to examine specularities from two views in order to improve the performance of surface interpolation. Coleman and Jain [5] use four-source photometric stereo to identify and correct for specular reflection components. In more recent work, Blake [1] assumes smooth surfaces and single point specularities to derive equations to infer surface shape using specular stereo. He shows that the same equations can be used to predict the appearance of a specularity on a smooth surface when using a distributed light source.

2

Takai, Kimura, and Sata [13] describe a model-based vision system which recognizes objects by predicting specular regions. As specular models and insights improve, we expect to see more work which makes use of the properties of specular reflection.

3. Specular Reflectance Models

Given a viewer, a surface patch, and a light source, a reflectance model quantifies the intensity the viewer will perceive. The most general reflectance models represent the perceived intensity I as a sum of three independent reflection components

$$I = I_A + I_D + I_S \qquad (1).$$

Here I_A represents the ambient reflection, I_D represents diffuse (Lambertian) reflection, and I_S represents specular reflection. In this paper we restrict our attention to the I_S reflection component.

We note that it is typically very easy to separate the I_S reflection component from the I_A and I_D reflection components in an image. There are two distinctive properties of specular reflection. First, over most of a surface I_S is zero, but in specular regions I_S is usually very large relative to I_A and I_D. Secondly, in regions where the specular component is nonzero, I_S changes much more rapidly than either I_A or I_D.

Before discussing the various specular reflectance models, we introduce the reflection geometry (Figure 1). We consider a viewer looking at a surface point P which is illuminated by a point light source. Define

$\bar{V}$ = unit vector from P in direction of viewer
$\bar{N}$ = unit surface normal at P
$\bar{L}$ = unit vector from P in direction of source
$\bar{H} = \frac{V+\bar{L}}{\|\bar{V}+\bar{L}\|}$ (unit angular bisector of $\bar{V}$ and $\bar{L}$)
$\alpha = \cos^{-1}(\bar{N}\cdot\bar{H})$ (the angle between $\bar{N}$ and $\bar{H}$)
$\theta = \cos^{-1}(\bar{N}\cdot\bar{V})$ (the angle between $\bar{N}$ and $\bar{V}$)

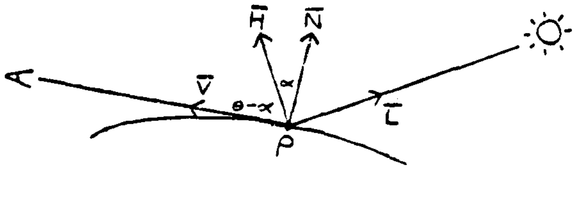


Figure 1. The Reflection Geometry

Throughout this paper, we consider only illumination from a single point light source. In principle, we lose no generality using this kind of an approach since we can describe distributed light sources as arrays of point sources. Thus to handle situations involving distributed light sources we only need to integrate the effects of an equivalent array of point sources.

The simplest specular model assumes that specularities *only* occur where the angle between $\bar{L}$ and $\bar{N}$ equals the angle between $\bar{N}$ and $\bar{V}$ and $\bar{L}$, $\bar{N}$, and $\bar{V}$ all lie in the same plane. This corresponds to the situation $\alpha = 0$ in Figure 1. Unless the surface is locally flat, this model predicts that specularities will only be observed at isolated points on a surface. A few experiments, however, show that this model is inadequate for most real surfaces. Not only are observed specular features usually larger than single points, but highlights often occur in places which are not predicted by this model.

An empirical model for specular reflection has been developed by Phong [10] for computer graphics. This model represents the specular component of reflection by powers of the cosine of the angle between the perfect specular direction and the line of sight. Thus Phong's model is capable of predicting specularities which extend beyond a single point. While Phong's model gives a reasonable first approximation which is useful in many practical situations, it is possible to develop more accurate models by examining the physics underlying specular reflection.

The Torrance-Sparrow model [14], developed by physicists, is a more refined model of specular reflection. This model assumes that a surface is composed of small, randomly oriented, mirror-like facets. Only facets with a normal in the direction of $\bar{H}$ contribute to I_S. The model also predicts the shadowing and masking of facets by adjacent facets using a geometrical attenuation factor. The resulting specular model is

$$I_S = \frac{FDG}{\bar{N}\cdot\bar{V}} \qquad (2)$$

where

F = Fresnel coefficient
D = facet orientation distribution function
G = geometrical attenuation factor

We will analyze the effects of each factor in the model in the next few paragraphs. The results we present in this paper are derived from (2).

The Fresnel coefficient F models the amount of light which is reflected from individual facets. In general, F depends on the incidence angle and physical properties of the reflecting surface. Cook and Torrance [6] have shown that to obtain realistic graphics images, F must characterize the color of the specularity. For metal surfaces F is approximately a constant.

The distribution function D describes the orientation of the micro facets relative to the average surface normal $\bar N$. Blinn [2] and Cook and Torrance [6] discuss various distribution functions. In agreement with Torrance and Sparrow we use the Gaussian distribution function given by

$$D = Ke^{-(\alpha/m)^2} \quad (3)$$

where K is a normalization constant. Thus for a given α, D is proportional to the fraction of facets oriented in the direction $\bar H$. The constant m indicates surface roughness and is proportional to the standard deviation of the Gaussian. Small values of m describe smooth surfaces for which most of the specular reflection is concentrated in a single direction. Large values of m are used to describe rougher surfaces with large differences in orientation between nearby facets. These rough surfaces produce specularities which are spread out on the reflecting surface.

The expression for the geometrical attenuation factor G is derived by Torrance and Sparrow in [14]. They assume that each specular facet makes up one side of a symmetric v-groove cavity. From this assumption, they examine the various possible facet configurations which correspond to shadowing or masking. They quantify the geometrical attenuation factor as

$$G = min\left\{1, \frac{2(\bar N\cdot\bar H)(\bar N\cdot\bar V)}{(\bar V\cdot\bar H)}, \frac{2(\bar N\cdot\bar H)(\bar N\cdot\bar L)}{(\bar V\cdot\bar H)}\right\} \quad (4).$$

We will show that in applications it is often possible to use a simpler expression for G.

As θ increases from 0 to $\frac{\pi}{2}$, the viewer gradually sees a larger part of the reflecting surface in a unit area in the view plane. Therefore, as θ gets larger, there are correspondingly more surface facets which contribute to the intensity perceived by the viewer. We take this phenomenon into account in (2) by dividing by $\bar N\cdot\bar V$.

4. Inferring Surface Properties

In this section, we demonstrate how we can use (2) to determine local surface properties from specularities. In almost all situations we do not require the full generality of (2) to infer these local properties. Our first assumption is that F is constant. This is a very good approximation for most specularly reflecting surfaces. We can further simplify (2) by observing that except for a small range of angles near grazing incidence, the value of G is unity. We will discuss this result and the exceptional cases later. Hence the form of (2) used to determine local surface properties is

$$I_S = \frac{Ke^{-(\alpha/m)^2}}{\cdot\,(\bar N\cdot\bar V)} \quad (5).$$

Referring again to the geometry of Figure 1, we assume that the viewer and light source are distant relative to the surface. Therefore $\bar V$ and $\bar L$ are essentially constant and hence their angular bisector $\bar H$ is essentially constant. We assume that the positions of the viewer and light source are known. Finally, since the distance from the viewer to the surface is large, we can approximate the perspective projection of the imaging device with an orthographic projection.

Doubly Curved Surfaces

For a surface which is doubly curved at a specularity (i.e. both principal curvatures are different from zero) we will be able to locate a single point P_0 of maximum intensity in the image of the specularity. From (5) we see that this point corresponds to the local surface orientation $\bar N = \bar H$ (i.e. $\alpha = 0$). Given a doubly curved surface where $\bar H$ is known, we can very quickly determine the surface orientation at P_0.

Figure 2 shows a typical specular image generated about an elliptic point on a surface.

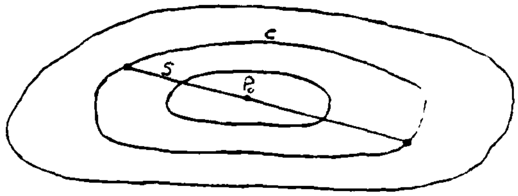


Figure 2.

Specular Intensities on a doubly curved surface

The closed curves are image curves of constant intensity. P_0 corresponds to $\alpha = 0$. As predicted by (5), intensity decreases as we move away from P_0. Now suppose we examine a straight segment S in the image which intersects P_0 and which begins and terminates on a constant intensity curve C. Let the measured intensity at P_0 be I_0 and let the measured intensity on C be I_1. Since the exponential factor in (5) will change very fast relative to the change in $(\bar{N} \cdot \bar{V})$, we can consider $(\bar{N} \cdot \bar{V})$ to be constant on S. Define $d\alpha$ to be the change in α as we move from the surface point imaging to P_0 to the surface point imaging onto C along S.

If we let $K' = \frac{K}{(N \cdot V)}$, then from (5) we have

$$I_0 = K', \qquad I_1 = K'e^{-(\alpha/m)^2} \quad (6)$$

which give us

$$d\alpha = m\sqrt{\ln I_0 - \ln I_1} \quad (7)$$

To determine the surface curvature along S we need to compute the arc length of the curve on the surface which generated S in the image. This will depend on $\bar{V}$. We do this by introducing an x,y,z coordinate system such that the surface curve S' imaging to S lies in the x-y plane (Figure 3). Denote by P_0' the point imaging to P_0 and place P_0' at the origin. Further arrange the coordinate system so that $\bar{H}$ is parallel to the y axis. Let $\bar{V}'$ be the projection of $\bar{V}$ onto the x-y plane.

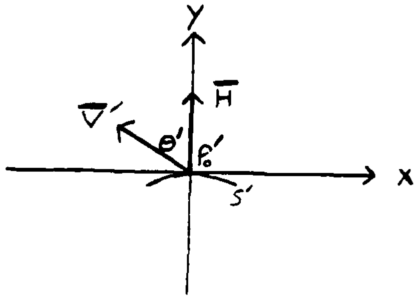


Figure 3. The Surface Geometry

Since the specularity will only be observable for a small range of α, we can approximate the arc length of S' by the length of its projection onto the tangent plane to the surface at P_0'. If l_S denotes the arc length of S' and l_I denotes the length of S in the image then letting θ' be the angle between $\bar{V}'$ and the y axis we have

$$l_S = \frac{l_I}{\cos\theta'} \quad (8).$$

Using (7) and (8) we can estimate the curvature κ on S' at P_0' as

$$\kappa = \frac{2d\alpha}{l_S} \quad (9)$$

Note that here we are using the fact that since the length of S' is small the surface normal at any point on S' lies approximately in the same plane as S' and $\bar{H}$ in Figure 3.

For any segment like S which intersects P_0 and begins and terminates on C we can compute a corresponding value of κ. If we examine these line segments for every direction in the image, then on the tangent plane to the surface at P_0' we will be examining the corresponding line segments $l_1, ..., l_n$ in every direction through P_0'. Consequently, on the surface we will compute κ for all curves formed by intersecting the surface with planes containing $\bar{N}$ and l_i (for i=1,...,n). The largest and smallest computed values of κ will give the principal curvatures of the surface at P_0' [12]. Therefore we can determine the principle curvatures and the principle directions at P_0'.

Singly Curved Surfaces

If one principal curvature of a surface is zero in a specular region we will not be able to immediately infer the local orientation as we did for a doubly curved surface. To understand why, consider Figure 4. Figure 4 shows a viewer looking at a tilted cylinder. To make the example concrete, assume that $\bar{L}$ is such that $\bar{H} = \bar{V}$.

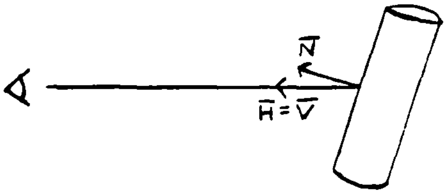


Figure 4. Viewer Observing a Tilted Cylinder

For this configuration there will be no point on the surface for which $\alpha = 0$ (recall that $\bar{H}$ is essentially constant), yet we will still observe a specularity in the image if at some point α is small enough to give a significant value for I_S in (5). The lines of constant I_S in the image appear as in Figure 5. Here C_0 corresponds to the smallest α and therefore the largest specular intensity. As predicted by the model, I_S decreases as we move away from C_0.

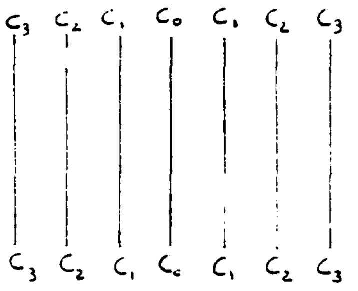


Figure 5. Lines of Constant I_S for a Cylinder

We observe that it is typically easy to detect the fact that a surface is singly curved at a specularity. This is because we will observe a line of maximum intensity (along the line of zero curvature) instead of the point maximum we observe for the doubly curved case.

Next we examine how we can infer curvature and orientation from an image of a singly curved surface. We cannot simply apply the analysis for the doubly curved case since in general we do not know α at surface points on the line of maximum specular intensity. In Figure 3 it is reasonable to assume that $\alpha = 0$ at P_0'. For the singly curved case, however, the relationship will be more complicated. Modify the geometry of Figure 3 so that $\bar{H}$ lies in the y-z plane and makes an angle ϕ with the x-y plane. The normal to S' at P_0' is still considered to be parallel to the y axis. If the curvature of S' at P_0' is $1/r$ then locally we have

$$\bar{N} = (\frac{x}{r}, \frac{\sqrt{r^2 - x^2}}{r}, 0) \quad (10)$$

$$\bar{H} = (0, \cos\phi, \sin\phi) \quad (11)$$

$$\alpha(x) = cos^{-1}(\frac{\cos\phi}{r}\sqrt{r^2 - x^2}) \quad (12).$$

This expression allows us to predict the appearance of I_S on a singly curved surface as a function of curvature. Figure 6 shows how I_S changes for a cylinder of fixed curvature as we change ϕ. It is worth noting that both the magnitude and shape of I_S change as ϕ increases.

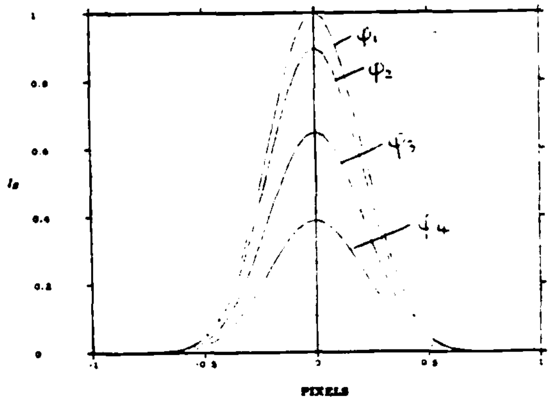


Figure 6. I_S for different values of ϕ

Planes

For a planar surface, $\bar{N}$ is constant. Hence recalling our basic assumptions, I_S is constant across a plane. If the plane is oriented such that α is small enough, then a viewer will perceive elevated intensity reflected from the plane. As with the singly curved surface, the magnitude of the perceived intensity will depend on α. If α is not sufficiently small, then I_S will be zero at all points on the plane. These observations provide us with two useful pieces of information:

1. Shiny surfaces which don't generate specularities over a range of orientations are probably planar.
2. Surfaces which produce a specularity of constant intensity over a 2-D region in the image are locally planar.

5. Predicting G

In the previous analysis we have used the fact that over most viewer, source, surface configurations the geometrical attenuation factor G of (4) will have the constant value 1. For large angles of incidence, however, the character of G changes remarkably. In particular, for large angles of incidence (glancing incidence) we see that

1. G can become as large as 10.
2. G causes a shift in the peak of the specular profile toward larger angles of incidence.
3. G causes the specular profile to be unsymmetric as a function of α.

It is not surprising that when these effects are present in an image, they are rather easy to detect. For this reason, it is probably profitable to make qualitative predictions about G in applications where large angles of incidence are possible.

6. The Laboratory Setup

A laboratory arrangement has been set up to test the derived relationships (Figure 7). In this section of the paper, the laboratory setup is described. In Section 7, we share insight gained about how to best use the input image data to infer surface properties from specularities in a working system. Experimental results are presented in Section 8.

Figure 7. The Specularity Lab

To insure accurate measurements, the experiments are conducted on a 4x6 foot optical table. High precision rotation and translation stages are used to position the objects being viewed. A halogen light source with a 5 mm wide filament is placed 20 feet from the object surface to approximate a point source. Monochromatic image data is obtained using a video camera and an image digitizer. A 210 mm lens is used with the video camera to obtain high resolution across the specularity. The resulting images are in the form of 256x256 arrays of pixels. Each pixel has eight bits of gray level resolution. A precise positioning device has been built to position the camera relative to the surface. Camera-object distances of at least 24 inches are enforced to insure that the assumed distant object condition is met. Using this setup, it is possible to obtain more than 40 pixels across a specular feature which is less than a centimeter wide on the surface.

Aluminum cylinders of diameter 3.5, 2.5, 1.5, and 0.75 inches are used to test the predicted relationships (Figure 8). The cylinders have been carefully machined to achieve uniformity of surface roughness on individual cylinders and between different cylinders. The length of each cylinder is divided into 4 sections of different measured roughness. From this we are able to study the effect of varying surface roughness on images of specularities. In the future we plan to experiment with different kinds of specular surfaces and also with surfaces which are doubly curved.

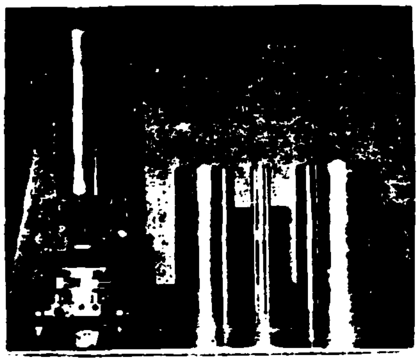


Figure 8. Experimental Specular Surfaces

7. Interpreting Real Images

In this section we discuss practical considerations related to using the results from Section 4 to infer surface properties from real images. For the first set of experiments, each cylinder is oriented such that $\alpha = 0$ on the line of maximum perceived intensity. For this special case, the doubly curved surface analysis applies to our singly curved surfaces (cylinders). Figure 9 shows a typical image obtained using this configuration. As

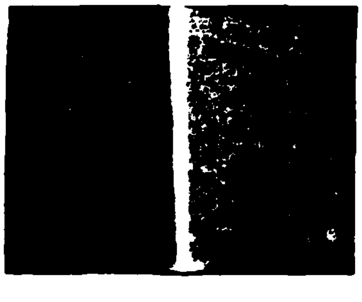


Figure 9. A Specular Image

7

previously discussed, the specularities are easily located in an image. It is reasonable to assume that $(I_A + I_D)$ is constant in the small neighborhood of the specularity. Thus we compute I_S by subtracting a constant from the pixel values on the specularity in an input image. I_S is set to zero elsewhere. Figure 10 shows a plot of I_S along a horizontal row of pixels taken from Figure 9 after the subtraction of $(I_A + I_D)$.

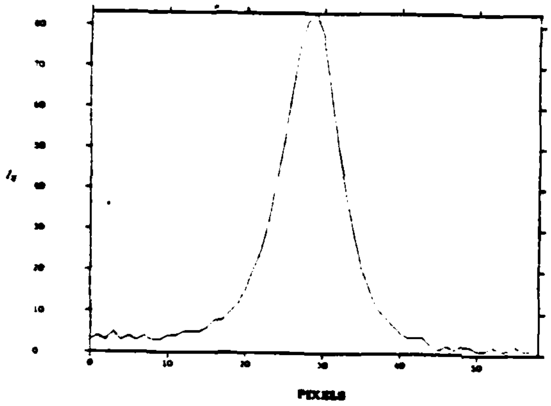


Figure 10. Plot of I_S

Computing the Curvature

Suppose now that we want to make the curvature computation described in Section 4 on the horizontal image line plotted in Figure 10. The problem is to select an appropriate pixel segment S in this image row which intersects the line of maximum I_S and begins and terminates on pixels of equal specular intensity. An equivalent problem is to draw a horizontal line in Figure 10 which intersects the I_S curve at 2 points of equal specular intensity. In principle, any image segment which satisfies the stated conditions will suffice. In practice, however, some choices for S are better than others. Figure 11 shows the case where S is chosen to be only in the high intensity part of the I_S curve. For this case S will only be a few pixels long and any error in the measurement will cause a large relative error in $|S|$. Figure 12 depicts a different kind of problem. Here S is drawn to connect two pixels of small specular intensity. Now S is many pixels long, but choosing the terminating points is difficult, if not impossible. This is because I_S changes only slightly from pixel to pixel on the fringes of the specular profile. Hence for this case there will probably be a large spatial range of pixels which are reasonable terminating points for S. Hence by choosing one of these, we risk introducing a large error in $|S|$.

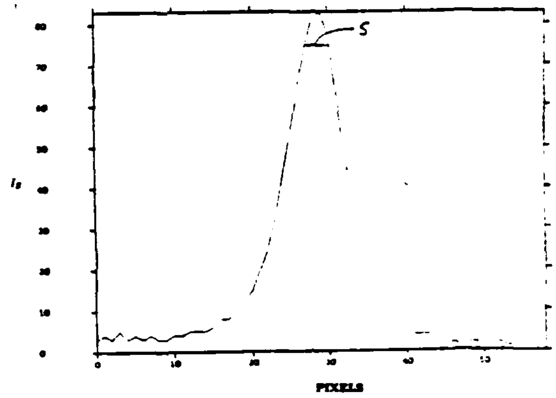


Figure 11. S Chosen Too High.

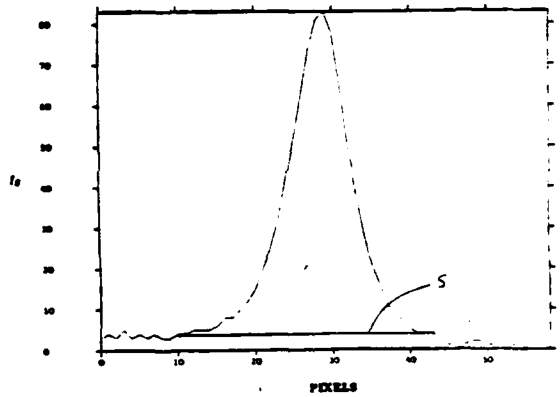


Figure 12. S Chosen Too Low

The solution, of course, is to have S begin and terminate at pixels of intermediate specular intensity. Since I_S is approximately Gaussian, we can determine a Gaussian fit for I_S and compute its standard deviation σ. Let μ be the pixel of maximum specular intensity (the mean of the Gaussian) and let T be the symmetric about μ pixel interval

$$T = [\mu - w, \mu + w] \quad (13)$$

over which S is defined. Consistent with the heuristic arguments given above, experiments have shown (see

Section 8) that an accurate curvature computation will result if we have

$$\sigma \leq w \leq 2\sigma \quad (14)$$

By imposing this restriction on T, we guarantee that we will choose an S which avoids either of the previously discussed pitfalls. We may improve our accuracy by performing the computations of Section 4 for several different S segments for which w satisfies (14). The curvature κ can then be obtained by averaging the curvatures computed for the different segments.

Truncation Effects

Since specularities are usually the brightest features in images, specular intensities are often too large to be represented in the number of bits per pixel allowed by the digitizing hardware. If this is the case, the specularity is said to be truncated. Figure 13 shows I for a truncated specularity. The obvious way to deal with this situation is to avoid it. One avoidance technique is to take multiple images in which differing amounts of light are allowed to pass through the lens. This can be achieved either by adjusting the lens aperture or by using filters. Another possible solution is to control the illumination to eliminate the possibility of truncation.

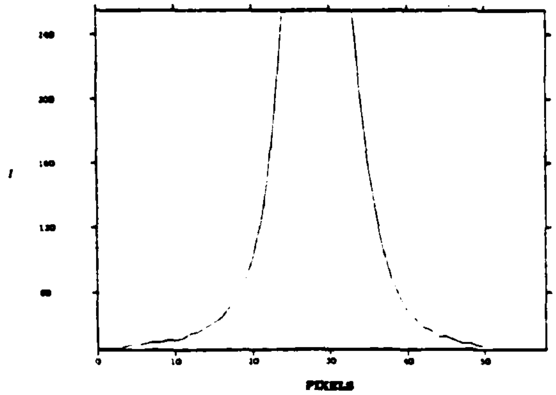


Figure 13. A Truncated Specularity

If inferences must be made from a single image, then it is arguably better to allow truncation to occur. In the case where input images have eight bits per pixel, intensities will range from 0 to 255. In many applications it is possible to weaken the incident illumination so that no truncation occurs. In doing this, however, we cause pixels on the I_S curve which previously had significant specular intensities (on the truncated specular feature) to have negligible specular intensities. The net effect of eliminating truncation is to decrease the width of the specular feature and make width measurements more susceptible to small errors.

8. Experimental Results

The technique described in Section 4 has been applied to the task of determining the curvature of each of the four cylindrical surfaces. Figure 14 shows the I profile for each of the four surfaces c_1, c_2, c_3, and c_4 taken along the line of maximum curvature. Note that we allow truncation. Table 1 displays the results of the experiments. For these measurements, w (see Section 7) is taken to be 1.85σ. We have observed that for the larger (less curved) cylinders c_1 and c_2 the computed curvature varies by less than 10% as we let w vary in the interval $\sigma \leq w \leq 2\sigma$. The reason for this desirable behavior is the fact that we are able to measure many pixels across specularities on these cylinders. On the other hand, if we attempt to measure the specular widths for the smaller cylinders c_3 and c_4 near the spec-

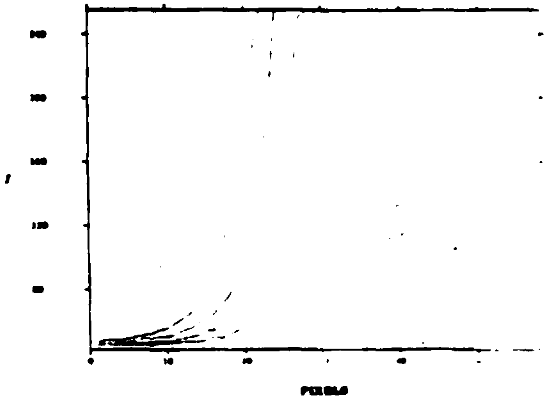


Figure 14. I for Four Different Surface Curvatures

Object	Specular Width in Pixels	κ	Computed κ	Error
c_1	26.9	0.2865	0.2965	3.4%
c_2	17.7	0.4016	0.4161	3.5%
c_3	11.4	0.6711	0.6369	5.1%
c_4	6.0	1.3514	1.2284	9.1%

Table 1. Curvature Computations for the Four Surfaces

9

ular intensity peaks, it is possible to get errors larger than 15% in the computed curvature. This is because near these peaks, we are only able to measure a few pixels across the specularities. Therefore for high curvature surfaces, it is advisable to measure the specular widths as near to the base of the curve as possible without falling victim to the problem discussed in Section 7. As is evident from Table 1, the more pixels which can be measured across a specular feature, the more accurately we can compute the curvature. Hence one immediate way to improve results is to digitize higher resolution images.

An interesting observation can be made from Figure 14. The specular model (2) described in Section 3, predicts that each of the four surfaces should generate the same maximum value for I_S when $\alpha = 0$. This prediction is intuitively appealing, since it seems that if we examine a small enough patch on any of the surfaces that patch should be approximately planar. But a cursory glance at Figure 13 seems to imply that a highly curved surface produces a smaller maximum value for I_S than a flatter surface. The model, however, is correct. The problem is that for the highly curved surfaces we are unable to shrink a pixel down to where the surface area it images is approximately planar. Even within the single maximum pixel, α is changing and cannot be considered to be constant zero. Hence the intensity value at the maximum pixel will be some kind of average specular intensity over a range of α and will not give us the true maximum I_S. Thus it is understandable that maximum measured intensity seems to increase as surface curvature decreases. It follows that to compute the maximum specular intensity in applications, we should use a surface of small curvature.

9. Summary and Implications

Understanding specular reflections is important for any computer vision system which must interpret images of shiny objects. Using a model developed by optics researchers, we have shown that the local orientation and principal curvatures of a specular surface can be determined by examining image intensities on a specularity. Unlike previous work, our derivations have included the effects of surface roughness and microstructure on the appearance of specular features.

A laboratory setup has been described which allows us to test our theoretical relationships. Very good results have been achieved despite the fact that the high intensity and small spatial extent of specularities make measurements difficult. Practical issues related to the implementation of our analytical results have been discussed. These issues have been addressed by the presentation of tested methods which successfully apply our results to the problem of interpreting real images.

The ability to predict intensity patch features such as specularities opens up interesting possibilities for model-based vision. Previous model-based vision systems have restricted their predictions to the shapes of image contours which will be observed for a given model. An ability to predict intensity patch features will significantly enhance the capabilities of a model-based vision system. Clearly it is advantageous to be able to make stronger predictions about an image by using additional information about the imaging process. A perhaps more important advantage of predicting intensity patches is that this prediction can provide strong guidance to low level intensity based visual processes such as edge detection. By making predictions about the appearance of intensity patch features we can hope to further unify the goals of the low level and high level mechanisms of a model based vision system.

10. Future Work

One plan for future work is to continue conducting experiments and working out details associated with inferring local surface properties from specularities. In the near future experiments are planned to study the effects of different specular materials, different kinds of roughness, and different surface orientations. We also plan to experiment with doubly curved surfaces and to develop a model which predicts the combined effects of rough surfaces and distributed light sources. It is expected that as experiments and analysis continue, we will be able to develop more refined algorithms for using specularities to understand images.

Another plan for future work is to formulate a general framework which will allow the qualitative prediction of the structure of intensity patches for a model. The primary challenge will be to isolate singularities in the structure of the intensity patches. As an example, a singularity occurs when we rotate a singly curved object from an orientation where a specularity is visible to an orientation where the specularity disappears. An ability to predict the structure and singularities of image intensity patches will open up significant new possibilities for model based vision systems.

Acknowledgements

This work has been supported by an NSF graduate fellowship, AFOSR contract F49620-82-C-0092, and ARPA contract N000-39-84-C-0211. The authors would like to thank Professor Bert Hesselink for generously providing laboratory space and equipment. We would also like to thank Rami Rise for his work in designing

the mechanical parts for the experiments.

References

[1] Blake A., "Specular Stereo," *Proceedings of IJCAI-9* (Los Angeles: August 1985), 973-976.

[2] Blinn, J., "Models of Light Reflection for Computer Synthesized Pictures," *Computer Graphics*, 11(2) (1977), 192-198.

[3] Brooks, R., "Symbolic Reasoning Among 3-D Models and 2-D Images," *Artificial Intelligence*, 17 (1981), 285-348.

[4] Chelberg, D. and Lim, H. and Cowan, C., "ACRONYM Model-based Vision in the Intelligent Task Automation Project," *Proceedings of Image Understanding Workshop* (1984).

[5] Coleman, E.N. Jr. and Jain, R., "Obtaining 3-D Shape of Textured and Specular Surfaces Using Four-Source Photometry," *Computer Graphics and Image Processing*, 18 (1982), 309-328.

[6] Cook, R. and Torrance, K., "A Reflectance Model for Computer Graphics", *Computer Gruphics*, 15(3) (1981), 307-316.

[7] Foley, J. and Van Dam, A., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, 1982.

[8] Grimson, W.E.L., "Binocular Shading and Visual Surface Reconstruction," MIT AI Memo 697 (1982).

[9] Ikeuchi, K., "Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method", *IEEE PAMI* 3(6) (1981), 661-669.

[10] Phong, B., "Illumination for Computer Generated Pictures," *Communications of the ACM* 18 (1975), 311-317.

[11] Shafer, S., "Optical Phenomena in Computer Vision," Univ. of Rochester TR 135 (1984).

[12] Spivak, M., *Differential Geometry*, Publish or Perish, Inc., 1979.

[13] Takai, K. and Kimura, F. and Sata, T., "A Fast Visual Recognition System of Mechanical Parts by Use of Three Dimensional Model," source unknown. (First author is with CANON INC. in Tokyo.)

[14] Torrance, K. and Sparrow, E., "Theory for Off-Specular Reflection from Roughened Surfaces," *Journal of the Optical Society of America*, 57 (1967), 1105-1114.

[15] Woodham, R., "Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from Image Intensity," Proc. SPIE, vol. 155 (1978).

APPENDIX D

On the Design of a Geometric Modeling System*

Pang-Chieh Chen

Introduction:

The goal of this project is to provide means of building up complex geometric objects using primitives and set operators such as union, intersection, and set difference. To achieve this goal, it is necessary to find a computer representation that can model the objects efficiently both in time and space to an arbitrary degree of accuracy. It is also necessary to devise efficient algorithms to manipulate the representations so that the set operations can be realized. Certainly there are many different ways to approach the problem, depending upon the types of primitives one wishes to consider. For our purpose, restrictions on the class of primitives should be minimal. With these specifications in mind, we proceed to discuss the topics of primitives, data structures, and algorithms.

Primitives:

Let $\vec{x}$ and $\vec{u}$ stand for vectors in world and object space. To avoid complications we shall work with orthogonal coordinate spaces only. For each primitive P, we shall stipulate that the mapping from object space $\vec{u}$ to world space $\vec{x}$ can be represented analytically in the form $\vec{x} = M(\vec{u})$. Furthermore, the surface of each primitive shall be piece-wise planar, with each piece constant in one object coordinate.

As an example, consider a cylinder as a primitive. Let $\vec{x}$ and $\vec{u}$ stand for the Cartesian coordinates (x, y, z) and the cylindrical coordinates (r, θ, z). The mapping could be

$$(x, y, z) = (r\cos\theta, r\sin\theta, z), \text{where}$$

$$r \in [0, r_0],\ \theta \in [0, 2\pi),\ z \in [0, z_0].$$

The surface of this primitive consists of three pieces, corresponding to $r = r_0$, $z = 0$, and $z = z_0$.

It is in general impossible to compute M^{-1}. So to represent the surface in world space we shall use a subdivision of the surface in object space as an approximation. Of course, finer subdivision leads to closer approximation, and such refinement process could be performed on-line to meet proper specifications. To formalize the subdivision process, let us introduce some terminology.

1. A vertex v is a point on the surface with its object coordinates known.
2. An edge e is a double-entity with two vertices as its endpoints. In both object and world space, it represents the straight line segment between the two vertices. Note that

[*] This work is supported in part by a fellowship from the Hertz Foundation

the two representations are not necessarily equivalent, because a straight line in the object space may not map to a straight line in the world space.

3. A **face** f is, like an edge, a double-entity. It has a boundary composed of edges and vertices. In both object and world space, it represents the planar surfaces generated by the surrounding edges and a center point. The center point in object space is the centroid of the surrounding vertices. Its world space coordinates is simply the image of the object space coordinates under M. Thus, f is a set of triangles in world space. In object space, f is a set of coplanar triangles. As a side note, the convexity of f in object space will be maintained throughout the program.

4. A **net** $N = (V, E, F)$ for a surface ∂P is a subdivision of ∂P. V, E, and F are the vertex, edge, and face set, resp. Note that N represents two different surfaces N_o and N_w, piece-wise planar in object and world space, resp. N_o corresponds to the exact surface ∂P, while N_w is only an approximation.

5. A **rope** $R = (V, E)$ for a curve L is a subdivision of L. It is merely a one-dimensional analog of a net.

We shall use a net N in world space to model the actual object surface. Similarly, the curve of intersection between two objects shall be modeled by a rope R. The fitness of the net approximation can be measured by the value

$$\max_{\vec{p} \in N_o} \min_{\vec{q} \in N_w} |\vec{p} - \vec{q}|,$$

evaluated in world space. However, because this value may not be easily computable, we employ an estimated measure, the **tightness** τ. The better the approximation, the smaller the tightness value. By definition,

$$\tau(N) = \max_{t \in f \in N} \hat{\tau}(t),$$

with t a planar piece of face f in N. Let a, b, c be the three vertices defining t, and n be the normal function so that $n(a)$ is the normal vector at a. Denote $A(t)$ the triangular area of $\Delta(a, b, c)$ in world space. Also denote the pair-wise function $\Theta(c)$ the angle between $n(a)$ and $n(b)$. We define

$$\hat{\tau}(t) = A(t) * \sum_{a} \Theta(a)^2.$$

This formula can be derived through a series of approximation. Roughly, $\hat{\tau}(t)$ represents the extremal distance between $\Delta(a, b, c)$ and a fitting spherical surface with solid angle defined by the three normals. Thus, intuitively, we are replacing the exact surface N_o by spherical surface patches.

To illustrate further, let us derive a plausible formula for $\hat{\tau}$ in the *one-dimensional* case. Given two points in space and a curve going through them, we wish to determine the fitness of the approximating straight line segment. Rather than using the actual curve for comparison, we use a circular arc instead. Let Θ be the angle between the two normals at the two points. Let d be the distance between the two points, and r be the radius of the circular arc. We now make the assumptions that the actual curve does not twist too much between the two points and that Θ is sufficiently small. We then have the following approximate relations.

$$d \approx 2r\sin(\Theta/2) \approx r\Theta,$$

$$\hat{\tau} \approx r - r\cos(\Theta/2) \approx r\Theta^2/4$$

$$\Rightarrow \hat{\tau} \approx r\Theta(d/r)/4 = \Theta d/4$$

Thus, within a constant factor, we can define $\hat{\tau} \equiv \Theta d$.

The algorithm for refinement, i.e. the process of reaching a sufficiently fine subdivision, is now obvious. We simply subdivide the faces in the object space until the tightness of the net becomes smaller than some specified value T. A plane $u_i = constant$ is used to split the face into two parts. The direction and the constant are determined from the original τ value so that the resulting τ values would be decreased in a favorable way. With proper initialization we see that the faces will always remain convex in object space under the splitting process. Furthermore, by the property of τ, the refining procedure will always terminate as long as T is positive.

To illustrate the refinement process, we have the orthographic projections of the nets representing a cylinder and a sphere. In figure 1, we have the initial net representing a cylinder of radius 50 and height 80. After refinement with $T = 100$ we have the net in figure 2. Figure 3 and 4 show the similar view of a sphere with radius 50.

An object can be described by the specification of the surface and the volume of space it occupies. Using the refinement algorithm we can construct a sufficiently tight net to model the object's surface. To represent the volume on the other hand, we shall use an oct-tree data structure. Now we discuss the data structures used in the system.

Data Structures:

To implement the net structure each primitive has access to doubly linked lists of faces and edges. Each face consists of a circular list of vedges, with each vedge composed of a vertex and a edge. The vedges, when accessed in the forward manner, form a CCW order with respect to the outside of the surface containing the face. By definition, each face belongs to one particular primitive and can have one of the six possible orientations (three directions with parity in each direction) in object space.

Each edge consists of two vertices as its endpoints. It also has two neighbors which consist of two faces or one face and one edge. The neighbor of an edge can be another edge belonging to a different primitive. It is this property that allows linkage of different primitives to form complex objects. The two edges that are neighbors of each other together model a piece of intersection curve between two primitives.

Besides the object and world coordinates, a vertex also contains a pointer to an ordered list of fedges describing its neighborhood. A fedge consists usually of an edge and a face. The fedges form a CCW order with respect to the vertex as viewed from above the surface. The fedge list contains only neighboring elements belonging to the same primitive containing the vertex. Thus, a fedge list may not completely describe the neighborhood of a point. However, at the end of the list where a face is usually expected, another vertex belonging to a different primitive may be in place instead. The set of vertices linked up as above would represent an intersection point that has the various object coordinates as well as the common world coordinates.

We use an oct-tree to model the volume of an object. Each cell c of a tree is either a node or a leaf. For simplicity we use a sphere to represent each cell. Note that the spheres would have to overlap and thus cause some wastes in space. The advantages of using spheres instead of cuboids or other types of volume are implementation simplicity and time efficiency. It is extremely simple to answer sphere-sphere and face-sphere intersection queries. Now, the content of each cell is the set of faces intersecting with the associated sphere and is denoted $C(c)$. We color a leaf black or white depending upon whether the sphere is totally inside or outside the object. We color it gray when $C(c)$ is nonempty. As one can see, using the tree structure we may place the elements of the net in a hierarchical organization to support efficient intersection algorithms.

We now come to the method of constructing the tree. For convenience let us call the process **solidification**. Clearly the iterative part of solidifcation is simple. With proper initialization, we simply recursively split a cell into eight equal parts and update the appropriate data structures. The splitting of a sphere is done by dividing the bounding cube into eight equal size cubes and then associate the descendent cells with the bounding spheres of these smaller cubes. The main problem is the termination condition. Intuitively, we wish to split the cells until we have separated the face elements into small groups. However, we cannot use $|C(c)|$ directly because a cell can intersect with the same face elements no matter how small the cell size become. What we need is some function to measure the amount of faces in a cell. Using this function we can split each cell until the amount of faces it contains is sufficiently small. On the other hand, we do not wish to have each face contained in too many cells, thereby causing both time and space inefficiency. To balance the trade-off we use the fraction of a face in a cell as the *amount* parameter for that particular face and cell.

To solve the above problem we construct the following functions.

$$\eta(c) = \sum_{f \in C(c)} \hat{\eta}(f,c),$$

$$\hat{\eta}(f,c) = \delta(r_i/r_f),$$

$$\delta(d) = \begin{cases} 0 & \text{if } d \le \hat{d}; \\ 1 & \text{otherwise,} \end{cases}$$

$$r_i = \min\{r_f, r_c, r_f + r_c - |v_c - v_f|\},$$

where r_f and r_c are the associated length of face f and cell c, resp. More explicitly, r_f is the square root of the area of f, while r_c is the radius of the sphere. v_c and v_f stand for the centers of c and f, resp. δ is a threshold function with threshold level $\hat{d}$. Essentially, $\hat{\eta}$ measures fractional face content of f in c. However, measuring the two-dimensional intersection area between f and c is not too simple. Instead, we use an one-dimensional analog for calculating the size of intersection. Segments of length r_f and r_c are appropriately placed on the line joining v_f and v_c. We then evaluate the length of intersection r_i. Using the ratio r_i/r_f we determine if f is *contained* in c. Here the parameter $\hat{d}$ comes into play. A value of 1/2 is probably a reasonable value for $\hat{d}$.

The intuitive meaning of η should now be clear. Basically, η measures the face containment c has. c is said to contain f if the size of intersection between f and c is large enough compared to the size of f. Now the termination condition for solidification can be stated. Each leaf cell shall have its η value less than some specified cell capacity S. It is recommended to have $S \approx 10$. Observe that as long as S is positive, termination of solidification is guaranteed. This is by noting the property that $\eta(c) \searrow 0$ as $r_c \searrow 0$.

Algorithms:

We have already discussed two algorithms in constructing the primitives, namely refinement and solidification. To perform set operation on two objects, we need a net-joining procedure to compute a sufficiently tight net for the resulting object. Thus, it is necessary to compute the intersection curves. One way to obtain the curves is to find the intersection points between edges of one net and faces of the other. For convenience, we shall refer to these points as knots. The knots constitute rope(s) for the intersection curves(s). The resulting net with the deletion of appropriate faces and introduction of new faces and sufficiently tight ropes for the intersection curves is the desired net representation for the final object.

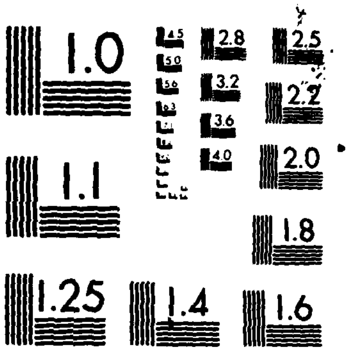
Finding the knots requires an iterative intersection algorithm. The idea of our algorithm is to use the intersection between e and f in world space as the initial point $\vec{p}$. We continue the initialization process by computing $\vec{q}_e$ and $\vec{q}_f$ in object space on e and f, resp., using a linear approximation. The linear approximation simply uses the associated weights $\vec{p}$ has on the vertices of e and/or f, and applies them in the corresponding object space. To iterate, we use the tangent line on $\vec{q}_e$ and tangent plane on $\vec{q}_f$ to compute the new $\vec{p}$ in world space. Let $\vec{p}_e$ and $\vec{p}_f$ be the image of $\vec{q}_e$ and $\vec{q}_f$ under M, resp. We use the difference $\vec{p}-\vec{p}_e$ and $\vec{p}-\vec{p}_f$ to perturb the object coordinates $\vec{q}_e$ and $\vec{q}_f$. By differentiating the equation $\vec{x} = M(\vec{u})$, we obtain the linear approximation

$$\Delta(x_i) = \frac{\partial u_j}{\partial x_i}\Delta(u_j).$$

Inverting the matrix we then express $\Delta(\vec{u})$ in terms of $\Delta(\vec{x})$. Using the perturbation formula we compute the new $\vec{q}_e$ and $\vec{q}_f$. The proper termination condition for the iteration is simply that distance $|\vec{p}_e - \vec{p}_f|$ be less than some user-specified tolerance. Occasionally, however, the iterated $\vec{q}_e$ or $\vec{q}_f$ would be pushed out of the interval specified by the vertices of e or f. In that case, appropriate refinement of e and f is taken before the iteration halts.

One should note that the above algorithm would work only if the edge has only one varying object coordinate. In the case of two varying coordinates, the edge together with its edge neighbor form an intersection curve. Thus, the problem of three primitives intersecting together is at hand. A modification of the algorithm is required. Instead of

AD-A172 266 CENTER FOR AUTOMATION AND MANUFACTURING SCIENCE(U) 3/3
STANFORD UNIV CA R H CANNON ET AL SEP 86
SPO-13649-01-00 AFWAL-TR-86-4053 F49620-82-C-0092
UNCLASSIFIED F/G 13/9 NL



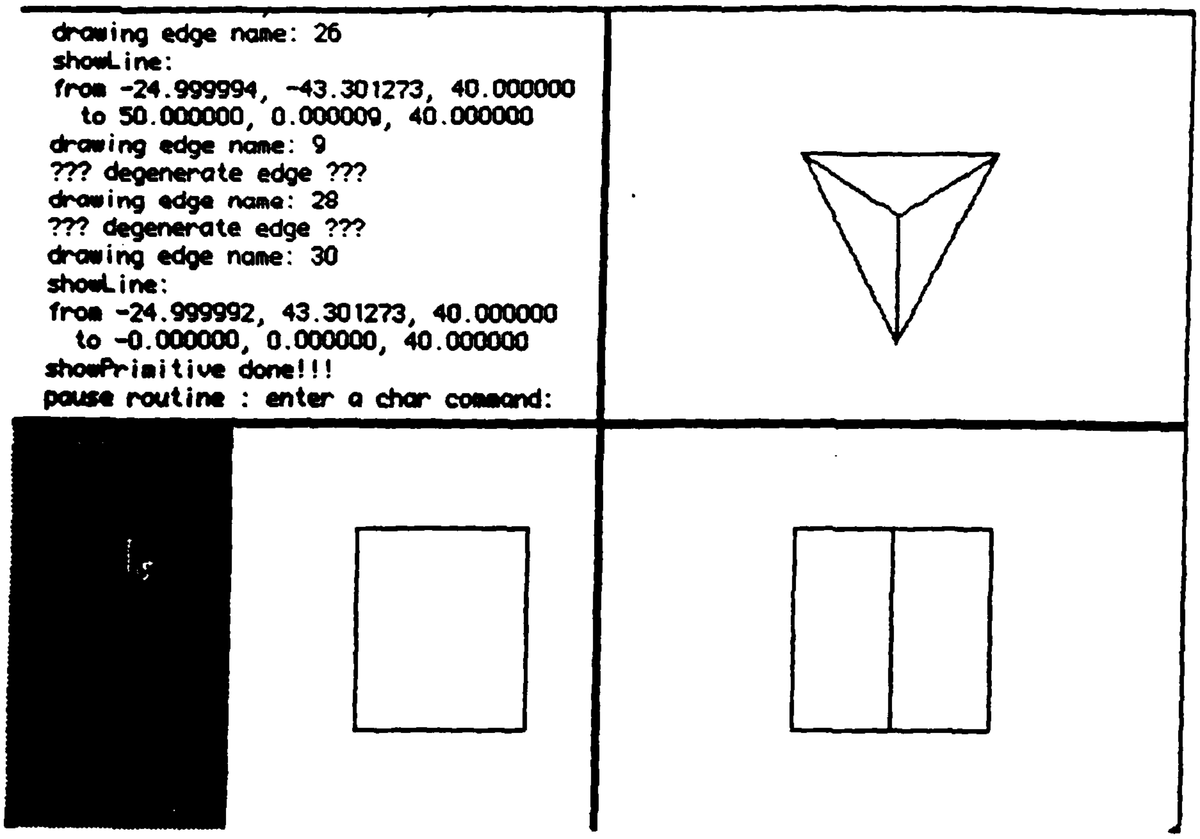
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

a tangent line on $\vec{q_e}$ as above, we use two tangent planes on $\vec{q_e}$ and its neighboring point. Thus, $\vec{p}$ is now the intersection between three planes instead of between a line and a plane.

Finding the knots can now be done easily by inquiring possible intersections between gray cells of the two objects. Each knot splits the corresponding edge and face. The original edge splits into two edges with one *visible* and the other *hidden*, depending upon the set operator and the orientation of the face. (Note: if the face does not have an inside-outside orientation, then the two edges would both be visible.) After the splitting process is done, the parts of the nets that are hidden may be removed using a seed-fill algorithm.

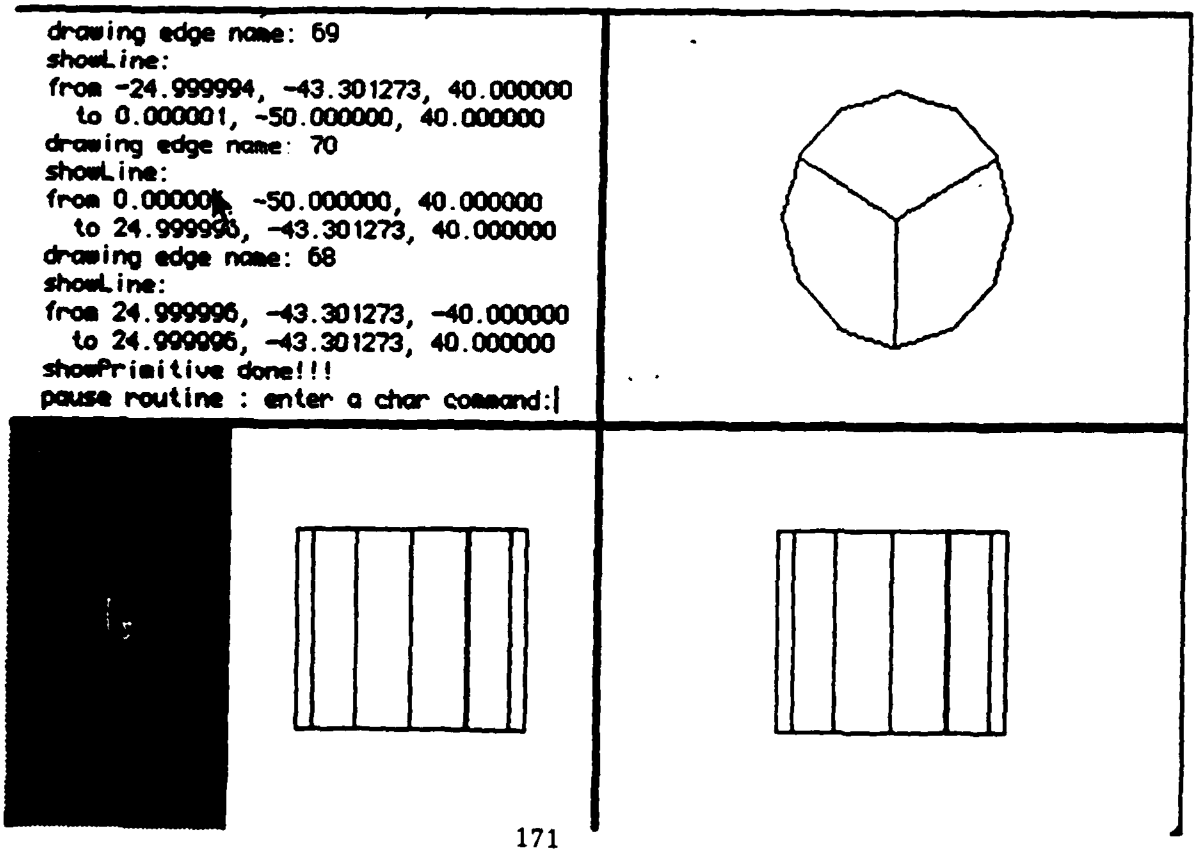
Connecting the knots to form ropes for the intersection curves is the next task in the net-joining procedure. This can be done by using the neighborhood information of the knots as stored in the fedge lists. The rest of the procedure is simply cleanup work involving further refinement of the ropes.

Implementation: The algorithms up to knot-finding have been implemented on a 512K MacIntosh. The rest of the work is being continued on the SUN workstation.



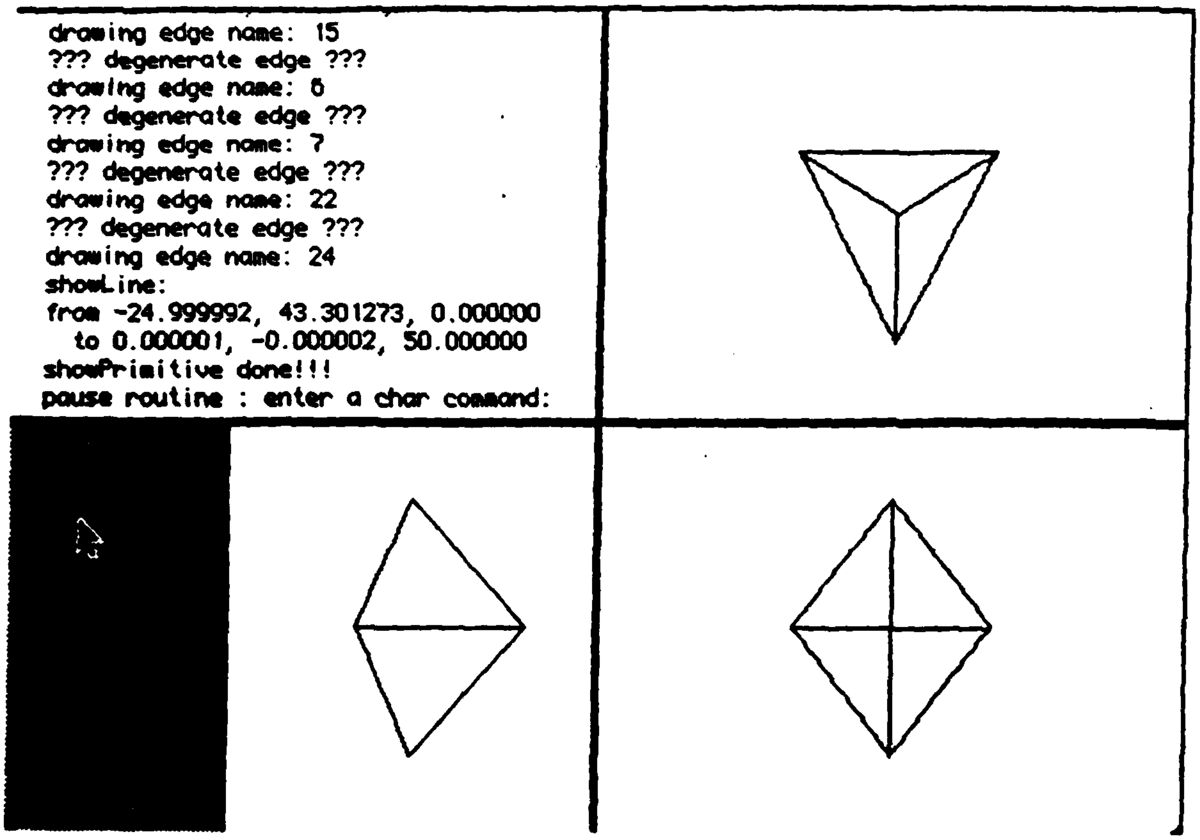
```
drawing edge name: 26
showLine:
from -24.999994, -43.301273, 40.000000
  to 50.000000, 0.000009, 40.000000
drawing edge name: 9
??? degenerate edge ???
drawing edge name: 28
??? degenerate edge ???
drawing edge name: 30
showLine:
from -24.999992, 43.301273, 40.000000
  to -0.000000, 0.000000, 40.000000
showPrimitive done!!!
pause routine : enter a char command:
```

figure 1



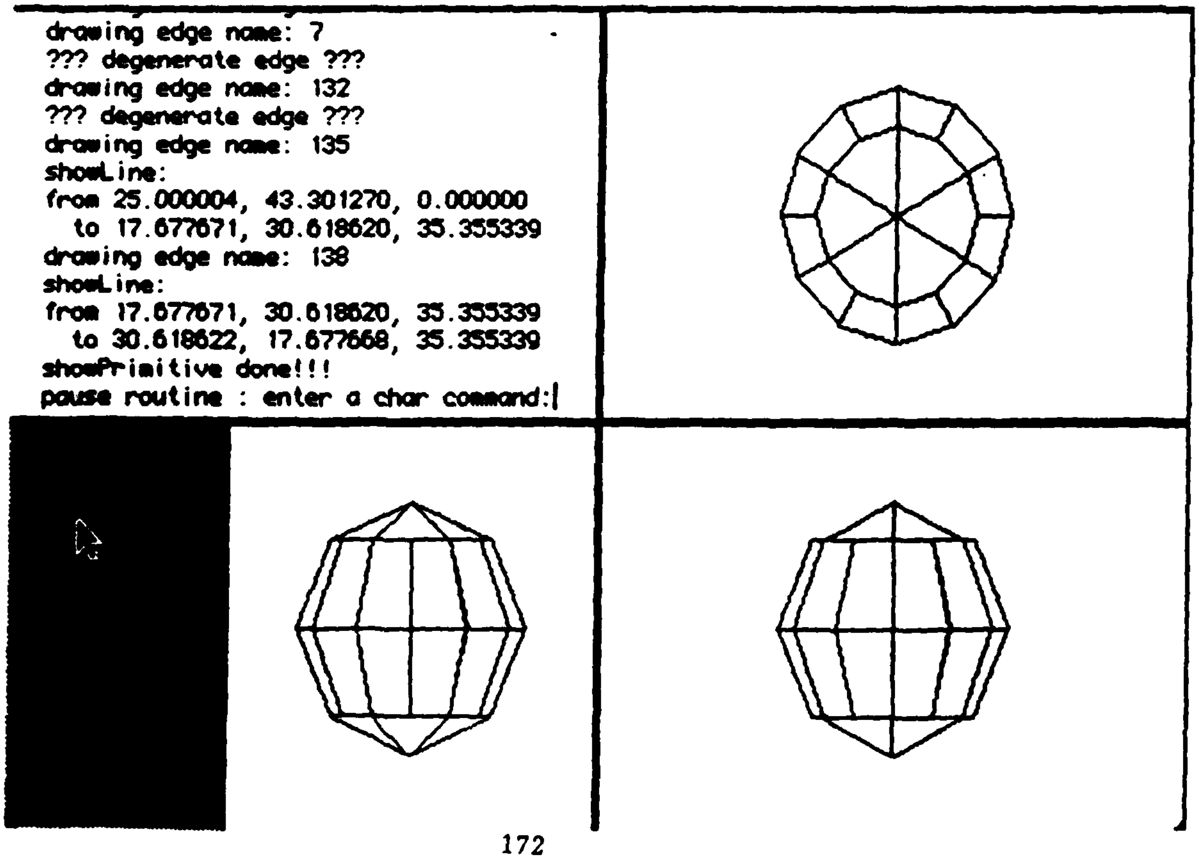
```
drawing edge name: 69
showLine:
from -24.999994, -43.301273, 40.000000
  to 0.000001, -50.000000, 40.000000
drawing edge name: 70
showLine:
from 0.000001, -50.000000, 40.000000
  to 24.999996, -43.301273, 40.000000
drawing edge name: 68
showLine:
from 24.999996, -43.301273, -40.000000
  to 24.999996, -43.301273, 40.000000
showPrimitive done!!!
pause routine : enter a char command:|
```

figure 2



```
drawing edge name: 15
??? degenerate edge ???
drawing edge name: 6
??? degenerate edge ???
drawing edge name: 7
??? degenerate edge ???
drawing edge name: 22
??? degenerate edge ???
drawing edge name: 24
showLine:
from -24.999992, 43.301273, 0.000000
  to 0.000001, -0.000002, 50.000000
showPrimitive done!!!
pause routine : enter a char command:
```

figure 3



```
drawing edge name: 7
??? degenerate edge ???
drawing edge name: 132
??? degenerate edge ???
drawing edge name: 135
showLine:
from 25.000004, 43.301270, 0.000000
  to 17.677671, 30.618620, 35.355339
drawing edge name: 138
showLine:
from 17.677671, 30.618620, 35.355339
  to 30.618622, 17.677668, 35.355339
showPrimitive done!!!
pause routine : enter a char command:|
```

figure 4

APPENDIX E

by David M. Chelberg

Introduction

This appendix describes a design for a system to model sensors and vision algorithms and describes a partial implementation of the system. Given sensor and algorithm models and mathematical models of objects we want to recognize, the system has the capability to make predictions about which combinations of sensors and algorithms will be most likely to meet a set of user-specified constraints. This makes it possible to determine appropriate sensor and algorithm combinations for visual tasks, e.g. to locate an object or class of objects.

Background

Current computer vision systems are dependent on the type of input data. Their dependence on a particular sensor type is usually embedded implicitly throughout the system where it is difficult to eradicate. The same is true for vision algorithms. High level systems do not consider how data was acquired, and do not have means to account for systematic and random errors of sensors and low-level algorithms.

Typically, high level systems cannot make accurate use of data. For example, the point scattering error of edge elements depends on contrast [Binford 81]. Without a model of this dependence, high level systems must assume some safe worst cast which is likely to be a factor of 10 worse than necessary. Where there are multiple parameters, high level systems have no method for parameter adjustment which leaves them with non-optimal settings. Even though the data from sensors and low-level algorithms have severe faults, high-level systems could perform substantially better by making better use of these low-level data.

For example, in [Brooks 81], an example of aircraft recognition was given. In figure 9.6, a view of a terminal at San Francisco airport with 6 aircraft, the limitations of the ribbon finder on small figures are apparent. The ribbon finder found ribbons suitable to recognize only one aircraft. In a subsequent experiment I carried out using ribbons fit by hand to the edges from the Nevatia-Babu edge finder, ACRONYM found all 6 aircraft. It is impossible to separate out the powerful human perceptual capabilities in this simulation of a ribbon finder, but prediction of the performance of the Nevatia-Babu edge finder was crucial in this intermediate level function.

Another problem of current vision systems is their lack of ways of operating in multi-sensor environments. To hadle this situation adequately, it is necessary to characterize the nature of each sensor, including the type of information each sensor provides as well as how the sensor obtained the information [Henderson 84a, 84b]. The type of information necessary is often characterisable as a vector of a few properties of the sensor, such as type of output (e.g. gray level resolution, spectrum), what it measures (range, image intensity), standard deviation, gain and bias, repeatability, dynamic range, optics, and the image input to the sensor. Measures of digitization and sampling are included because we are dealing with digital data.

Sensor Modeling

The first goal of this system is to model various sensors and algorithms adequately. To do this, a framework for modeling was developed. the most important aspects of a sensor are what it senses and what type of information it returns. For example, a camera senses spatial light intensity and returns a set or field of vectors giving position and intensity information. Thus a camera transforms its input data – the light incident in its field of view, into density of dye components at a point on the film plane. Another example is a laser range finder. One type measures the phase of a return signal, proportional to the distance to a reflecting surface along the sensor direction. This information is a triple of range (modulo wavelength) and two angles for the direction.

A uniform method of modeling sensors and algorithms must describe the transformations that sensors and algorithms perform. We achieve this by thinking of both algorithms and sensors as functions which transform their inputs to produce outputs.

This uniform approach models composing an algorithm with a sensor as a composition of their functions. Predictions abot the performance of various combinations is a major reason for our modeling of them, since this will allow determining the best selection for a vision task.

The properties of sensor and algorithm must be modeled with statistical and parametric models. Models must interact within the overall system and relate to other models. The faithfulness of models to the real world is one criterion for choice of model; the simplicity of the model for symbolic prediction is another. The design philosophy thus far has been to begin with simple models but to design the system to allow more complex models as necessary. The framework encompasses these models naturally. It is intended that the system choose model complexity to fit the requirements of the system measurement.

Prediction System

The prediction system is intended to produce grey scale imagery, color imagery, depth maps, edge maps, and other outputs. Outputs can be produced from the polyhedral approximations to generalized cylinders, or symbolic descriptions of various output types may be generated for classes of generalized cylinders. For example, for cylinders, the system would predict step edge intensity profiles for the ends of the cylinder, and curved edge intensity profiles for the limbs (apparent edges).

The prediction system will have to determine the performance of sensors and algorithms given their models, task models, and object models. This system will be complex. The current design calls for implementation of this system as a search over the graph of sensor and algorithm combinations, supplemented by a suite of functions which can determine input/output characteristics of a sensor or algorithm from the model of the sensor.

A difficult part of this may be making the models of algorithms. The behavior of algorithms is not modeled in most publications. I currently plan to start with a few algorithms that have been developed at Stanford since I have access to people who know the behavior of the algorithms.

Implementation of the Sensor Modeling System

The current implementation is in Frans Lisp with a record package to define structured data representations. A sensor or algorithm is defined as an object with the following slots: a unique name; a set of input types – which can be the real world for sensors; a description of the output; a set of properties of the sensor, such as resolution, sensitivity, etc.

Using this framework, a program can form a graph with all possible combinations of sensors and algorithms. The graph can be partitioned on the basis of final output types. It is not necessary or likely to build this entire graph. Parts of it may be generated in response to different request of the system and then stored for later use. It is useful to think of prediction as being a search of a portion of the graph, using constraints to guide the search.

Object Modeling

The current system relies on the ACRONYM modeling system. This system uses generalized cylinder primitives with restrictions to implement object classes. An interface to this system to allow generation of polyhedral approximations to generalized cylinders was written. Extensions were made to the modeling system to allow specification of surface properties of objects, such as color and specular and diffuse ????? The prediction part of the system has yet to be implemented. components of reflectivity.

References

[Brooks 81] Brooks, R.,
"Symbolic Reasoning Among 3-D Models and 2-D Images"; PhD Thesis, Stanford University, 1981.

[Henderson 84a] Henderson, T.C., Shilcrat, E.,
"Logical Sensor Systems";
Journal of Robotic Systems v1, p169, 1984.

[Henderson 84b] Henderson, T.C., Fai, W.,
"MKS: A Multi-Sensor Kernel System";
IEEE Transactions on Systems, Man, and Cybernetics v14, p784, 1984.

APPENDIX F

SURFACE MODELS

Richard Scott

Computer Science Department
Stanford University, Stanford, California 94305

Abstract

The purpose of this work is to provide the elements of a symbolic graphics system: modeling language, geometric models and splines plus algorithms to represent the visible and hidden surfaces. It extends previous work on a display program.

Introduction

The input to the graphics system consists of of combinations of generalized cylinders defined using a modeling language. A new formulation for a wide class of cylinders is used, based upon curves defined by splines. The spline method which is not yet completely derived, takes point, tangent and curvature boundary conditions and interpolates with continuous tangent and curvature.

The output consists of a description of the model surfaces and a description of the image. It divides the model surfaces into their forward facing, back facing and occluded regions which are paired with the areas of image on which they project.

Many of the intervening algorithms such as limb following, special cases, and formulae for derivatives of surface normals are simplified and formalized from the previous display program.

The remaining sections of the report are:

1. Class of generalized cylinders
2. Modeling language
3. Classification of limb points
4. Spline method
5. Extensions to Graphics
6. References

1. The Class of Generalized Cylinders

1. Class of generalized cylinders
1.1 Overview
1.2 Examples
1.3 Spline with frames
1.4 Loops and singular points
1.5 Evaluation, compliling, and discontinuities

1.1 Overview

(See [1] and [2]). Members of the class of generalized cylinders are defined by three successive sweeps: from point to curve to ribbon to generalized cylinder. The terms used here are that a "cross-section" is swept along a "spine" curve to form a "section". The cross-section can be a point, curve or area and the section is a curve, ribbon, or generalized cylinder respectively. (Section n) is the class of an object formed by n sweeps. At the lowest level of the modeling system the curves are generated by splines.

The bottom and top cross-sections are positioned relative to each other by placing the lower end of the spine at an attachment point on the bottom cross-section and then placing the top cross-section on the upper end of the spine.

The top cross-section is a deformation of the bottom one. Intermediate cross-sections are produced by partially deforming the bottom cross-section.

The spine is a moving frame, whose axes at a point orient the cross-section. For example to twist a cross-section, the axes of the moving frame rotate about the spine tangent. Having a frame at every point allows the special cases of translation and rotation to be treated uniformly: For translation, the frames are parallel and the cross-section is not deformed as it is swept: There is no need to have an attachment point between the spine and cross-section, because all attachments produce the same figure. Rotation about a point occurs when all the points of the spine are coincident.

The sweeping rule is defined completely by the spine, deformation and attachment point; these elements are independent of each other and of the cross-section, so that components from different GCs can be recombined.

The deformation method follows after some examples defining a cuboid and a right circular cone.

1.2 Examples

A cuboid with sides l1, l2, l3:

```
((Class GC) (Name cuboid)
        (Variables (l1) (l2) (l3))
        (Spine (straight (length (l3))
                    (direction zhat)))
        (Cross-area rectangle)
  )
((Class ribbon) (Name rectangle)
          (Variables (l1) (l2))
          (Spine (straight (length (l2)) (direction yhat)))
          (Cross-curve (straight (length (l1)) (direction xhat)))
)
```

The top and bottom cross-sections are equal, and no attachment point is needed to connect the spine and cross-section because it is a translation.

A generic cone:

```
((Class GC) (Name cone)
          (Variables (radius) (height))
          (Spine (straight (length (height)) (direction zhat)))
          (Cross-area
           (Bottom (circle-area (center origin) (radius)))
           (Top (Deform bottom (shrink-to-point center))))
          (Attach (frame center unit-axes))
          )
```

Here the bottom cross-section is a circle area, the top cross-section is the singular center point, while intermediate cross-sections shrink the circle radius linearly down to zero.

```
((Class ribbon) (Name circle-area)
            (Variables (radius) (center))
            (Spine (center))
            (Cross-area
             (Bottom (circle (radius) (center)))
             (Top (Deform bottom (shrink-to-point (center)))))
            (Attach (frame (center) unit-axes))
            )
```

A particular cone:

```
((Class cone) (Name cone1)
           (Radius 1.0 (unit inch))
           (Height 5.0 (unit inch))
    )
```

A cuboid with the cross-curve of its rectangular cross-section warped into a circular arc of angle two theta:

```
((Class GC) (Variables (theta) (l1) (l2) (l3))
          (Deform (ref (cuboid cross-curve))
                (Deform (ref (top tangent))
                      (Rotate (angle (theta)) (axis zhat)))
                (Deform (ref (bottom tangent))
                      (Rotate (angle (minus (theta))) (axis zhat))))
   )
```

1.3 Splines with Frames

A moving coordinate-frame along the spine determines the orientations of cross-sections.

Given a directed curve in the xy plane, natural axes can be assigned uniquely by rotating the xyz unit frame about the z axis to align the x axis with the curve tangent. The axes can be reoriented by twisting around the tangent direction; in general there are axes boundary conditions in the form of orientation, instantaneous axis of rotation and rate of rotation.

In addition a curve with axes along its length can be deformed by changing the position, tangent and curvature boundary conditions. Two things can happen to the axes. They can either be kept at constant absolute orientation or they can be rotated so that their orientation relative to the natural frame is kept constant. Corresponding points on the original and deformed curves have the same ratio of arc length to total curve length. For example a helix formed by a straight spine with twist:

```
((Class GC) (name helix)
        (Variables (height) (pitch) (big-radius) (small-radius))
        (Spine (Deform (straight (direction zhat)
                          (length (height))
                          (axes unit-axes))
                   (Deform (top axes)
                        (Rotate (axis zhat)
                             (Angle (* (pitch) (height)))))))
        (Cross-section (circle-area (radius (small-radius))
                            (center (big-radius) 0.0)))
        (Attach (frame unit-frame)))
```

1.4 Loops and Singular Points

It is important to have topological features repesented by the model; eg. curves which form loops or singular points instead of a closed interval. This allows things like the hole in the torus and the point on top of a cone to be deduced. The axes along a looped spline must rotate an integer number of times. There are two ways to form a volume or ribbon with a hole; one is:

```
(Spline loop (boundary-conditions)),
```

the other is when the two spine sweeping method has the bottom end of one spine positioned at the top of the other.

The only way that points and lines can be singular *i.e.*, have a dimension smaller than expected, is when the deformation shrinks to zero, or in the two spine case, when the spines cross each other, eg. in the dumb-bell definition above.

1.5 Evaluation, Compiling and Discontinuities

Evaluation:

(Section n) is generated by up to four sweeps with parameters named t1, t2, t3 and t for sweeping through time. By setting values to a subset of these parameters, (Section n) is evaluated to yield an object of class (Section m) formed by the sweeps of the remaining unset parameters. For instance,

(Section 3)(t1 t2 t3) = point on generalized cylinder,
(Section 3)(t3) = a ribbon,
(Section 3)(t1) = a ribbon: t2 spine swept along the t3 spine.

The formulae for the first and second derivatives are computed automatically. Eg. $\frac{dp}{dt1}$ and $\frac{d2p}{dt2dt3}$:

```
(Derive1 (parameter t1) (Section 3) (t1 t2 t3))
(Derive2 (parameters t2 t3) (Section 3) (t1 t2 t3))
```

The bounds on the parameters form a cuboid of parameter space. If (t1 t2 t3) lies on the surface, then the surface normal, given by (Normal (Section 3) (t1 t2 t3)) equals the dot product of the parameter space surface normal with,

$$((X \frac{dp}{dt2} \frac{dp}{dt3}) (X \frac{dp}{dt3} \frac{dp}{dt1}) (X \frac{dp}{dt1} \frac{dp}{dt2})),$$

where X means cross product and p is the position on the GC surface.

Discontinuities:

When the parameters lie on a corner or edge of the parameter cube, or when some of the defining splines have a tangent discontinuity at that parameter point, then the cylinder has a surface tangent discontinuity. Then the formula above gives a list of normals from which any sum of positive multiples is perpendicular to the surface.

Compiling:

The three main parts to compiling a generalized cylinder definition are: calculating the spline parameters; forming lisp functions for position and its derivatives; and building a data structure to keep track of where the present parameter point is located compared to the discontinuities.

2. Modeling Language

Examples of the modeling language were used to describe the class of generalized cylinders, so this section is brief.

2.1 Top Level
2.2 External input format: Names, definitions, references
2.3 Incomplete definitions and quoting
2.4 Name environment

2.1 Top Level

At the highest level the modeling system is a network of nodes, each of which has a class rule and a value rule associated with it, which must be known beforehand to access the node. Different kinds of nodes have different rules.

Class-rule: Node ↦ Class
Value-rule: Node ↦ Value

The nodes and rules of immediate interest are those of the external input format of the modeling system. The input function takes a list of definitions, which it first stores exactly as is. Then it goes through and expands the definitions.

2.2 External Format: Names, Definitions, References

There are three basic formats for an input node, examples of which are:

```
((Class C) (Value V)), and
((Value V) (Class C)), and
(C V)
```

where C and V are names of other nodes. To give the node (C V) the name "N", insert "(Name N)" to the list, or alternatively put (C V) into the value position and (Name N) in the class position:

```
(C (Name N) V), or
((Name N) (C V))
```

If also the node

```
((Name C) (Class D) (Value W))
```

is input, then the system links to form the internal equivalent of:

```
((Name N) ( ((Name C) (Class D) (Value W)) V)).
```

Apart from a name, two other kinds of reference are possible. The most common one is a nested definition, which takes the form of a list of nodes; for example the two equivalent forms below. The field classes A and B can be omitted when their values appear in the same order as they were given in the definition of C.

```
((Class C) (A (X Y)) (B Z)).
(C (X Y) Z)
((Class record) (name C) (Fields (A) (B)))
```

The final kind of reference is a path to a previously defined node eg:

```
((Class O) (reference (N (A X)))), or
((Class O) (reference (C (A X))))
```

which are both converted internally to,

```
((Class O) (X Y)).
```

An alternative form is to reverse the reference path, as in

```
((Class O) (reference-of (X (A N)))).
```

Also the forms,

```
(Value-of N) and
(Class-of N)
```

yield V and C respectively.

2.3 Incomplete Definitions, and Quoting

A definition is incomplete when a node has a class but no value; for example a variable in a function definition, or a field in a record definition; see the definition of a generic cuboid in section 1.2. It is indicated by brackets around the class eg.

(C) instead of (C V). Sometimes parts of a definition are quoted to prevent them from being expanded upon input, for example when a value does not have the input format. The class will contain instructions on how to interpret the value but the input method does not know that. (If the class is missing from the modeling system then the class is the programmer himself, who had better know how to interpret the value structure. For want of a better name this class is called God.) The other main use of quoting is in "(Name N)" which quotes "N" automatically. If this were not so the name would be recursively expanded for ever.

3. Classification of Limb Points

A classification of seven types of limb point, their properties and their possible orders is part of a robust limb following algorithm [10] and also of theoretical interest. It is derived by considering five kinds of surface patch and the ways that limbs can cross them.

The surface patches are:
A smooth,
B crossed by an edge,
C with a junction point of three or more edges,
D with a singular point like the top of a cone, and
E with the end point of a crease.
A must have a smooth limb crossing it, (1).
B has three types of limbs: Straight along the edge (2), crossing over the edge (3), and half off the edge and half on it (4). Types (1) and (2) have no tangent discontinuities on the surface, in the image or in their parameter values. Type (3) has surface, image and parameter tangent discontinuities. Type (4) differs from (3) by being continuous in the image.
C has three types of limb: (3), (4) and one which switches edges (5), having surface, image and parameter discontinuities.
D has one limb type (6), with surface, image and parameter discontinuities.
E has two types, (6), and (7) which falls off the end of the crease, giving surface, image and parameter discontinuities.

The ordering constraints are obvious, eg. (1) must be linked to (2) via (4) or (7).etc.

Acknowledgements

This research is supported by the Air Force Office of Scientific Research F49620-82-C-0092

6. References

[1] Binford, Thomas O., "Generalized Cylinder Representation," 1985.

[2] Brooks, Rodney A., "Symbolic Reasoning among 3D Models and 2D Images," *Phd. Thesis, Stanford,* 1981.

[3] Goldberg, and Robson, "Smalltalk80: The Language and its Implementation."

[4] Edelsbrunner, and Guibas, and Stolfi, "Optimal Point Location in a Monotone Subdivision," *Stanford,* 1984.

[5] Hilbert, "Geometry and the Imagination," *Chelsea* 1954.

[6] Jeffreys, and Jeffreys, "Methods of Mathematical Physics."

[7] Marimont, David "A Representation for Image Curves," *Proc. AAAI* 1984.

[8] Pavlidis, Theo "Curve Fitting with Conic Splines," *ACM Trans. on Graphics. Vol2. No.1* Jan. 1983.

[9] Salmon, "Conic Sections," *6th Ed. Chelsea, New York* 1954.

[10] Scott, Richard S.F., "Graphics and Prediction from Models," *Proc. IU Workshop,* New Orleans, October 1984.

APPENDIX G

MINIMUM-TIME CONTROL OF A TWO-LINK ROBOT ARM

Abraham Weinreb and Arthur E. Bryson Jr.

Stanford University, Stanford, California

ABSTRACT

- Minimum time paths for a two-link robot arm are determined so that the tip moves a specified distance, starting from rest and ending at rest. Two torquers are used, one at the shoulder and one at the elbow, each having bounded torque. The optimal locations of the two end points in the robot coordinate system are determined as part of the solution. The paths are found to be "bang-bang" in both controls. For distances that are small compared to the reach of the arm, the minimum time paths are asymmetric, changing to symmetric paths when the distance is just slightly less than the maximum reach of the arm.

- A new gradient algorithm for systems with multiple bounded controls was developed to solve the problem (cf. Ref. 1).

INTRODUCTION

- Robot arms are now used in many industries. As of now, they are massive and move relatively slowly. To move the tip along a prescribed path, appropriate joint angle histories are determined and servo-actuators at each joint cause these histories to be followed as closely as possible. The limits on speed are (1) the maximum available torques at the joints, and (2) vibrations of the links. The faster the arm moves, the more work it can do, so minimum-time paths are of considerable interest.

- We consider here minimum-time paths to move the tip of a rigid two-link robot arm with a tip mass (the "payload") through a specified distance, starting and ending with zero velocity (a "pick-and-place" movement), assuming actuators with bounded torque at the "shoulder" and the "elbow" (see Figure 1). A unique feature of our formulation of the problem is that the optimum orientation of the arm with respect to the line connecting the two end points is determined. Previous work on this type of problem is contained in Refs. 3 through 5.

EQUATIONS OF MOTION

- The equations of motion are

$$J_{11}\dot{\omega}_1 + (J_{12}\cos\alpha)\dot{\omega}_2 = J_{12}\sin\alpha\,\omega_2^2 + Q_1 - Q_2 \;, \qquad (1)$$

$$(J_{12}\cos\alpha)\dot{\omega}_1 + J_{22}\dot{\omega}_2 = -J_{12}\sin\alpha\,\omega_1^2 + Q_2 \;, \qquad (2)$$

$$\dot{\alpha} = \omega_2 - \omega_1 \;, \qquad (3)$$

$$\dot{\theta} = \omega_1 \;, \qquad (4)$$

where

$[\omega_1, \omega_2]$ = angular velocity of link [1, 2],

α = elbow angle,

θ = angle of link 1 relative to its initial orientation,

$[Q_1, Q_2]$ = actuator torques at [shoulder, elbow],

and

$[m_1, m_2]$ = masses of [link 1, link 2 + tip mass],

$[l_1, l_2]$ = length of link [1, 2],

a = distance from elbow to center-of-mass of (link 2 + the tip mass).

$J_{11} = J_1 + m_2(l_1)^2$,

J_1 = moment-of-inertia of link 1 about the shoulder,

$J_{12} = m_2 l_1 a$,

J_{22} = moment-of-inertia of (link 2 + tip mass), about the elbow.

▪ It is convenient, for numerical integration, to put Eqns. (1) and (2) into the form:

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{bmatrix} = F(\alpha)\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \end{bmatrix} + G(\alpha)\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \qquad \begin{matrix} (5) \\ (6) \end{matrix}$$

where

$$F(\alpha) = \frac{J_{12}\sin\alpha}{\Delta(\alpha)}\begin{bmatrix} J_{12}\cos\alpha, & J_{22} \\ -J_{11}, & -J_{12}\cos\alpha \end{bmatrix},$$

$$G(\alpha) = \frac{1}{\Delta(\alpha)}\begin{bmatrix} J_{22}, & -J_{22} - J_{12}\cos\alpha \\ -J_{12}\cos\alpha, & J_{11} + J_{12}\cos\alpha \end{bmatrix},$$

$$\Delta(\alpha) = J_{11}J_{22} - (J_{12})^2\cos^2\alpha.$$

THE MINIMUM-TIME PROBLEM

▪ We wish to find the initial elbow angle $\alpha(0)$, and the control torques $Q_1(t)$ and $Q_2(t)$ to move the tip through a specified distance L in minimum time, starting and ending with zero tip velocity with the control torques bounded, i.e.

$$|Q_1(t)| \leq Q_{1M}, \tag{7}$$

$$|Q_2(t)| \leq Q_{2M}, \tag{8}$$

▪ The distance traveled by the tip may be expressed as

$$D(t) = [(x(t) - x(0))^2 + (y(t) - y(0))^2]^{1/2}, \tag{9}$$

where

$$x(t) = l_1\cos\theta(t) + l_2\cos[\theta(t) + \alpha(t)], \tag{10}$$

$$y(t) = l_1\sin\theta(t) + l_2\sin[\theta(t) + \alpha(t)] \; . \qquad (11)$$

- The initial conditions are

$$\omega_1(0) = 0, \;\; \omega_2(0) = 0 \; , \;\; \theta(0) = 0 \; , \qquad (12)$$

with $\alpha(0)$ to be determined.

- The final conditions are

$$D(t_f) = L, \;\; \omega_1(t_f) = 0, \;\; \omega_2(t_f) = 0. \qquad (13)$$

- An integral of the motion is obtained by observing that the total angular momentum of the arm is equal to the angular impulse imparted by the shoulder torquer, i.e.

$$(J_{11} + J_{12}\cos\alpha)\,\omega_1 \; + \; (J_{22} + J_{12}\cos\alpha)\omega_2 \;\; = \;\; \int_0^t Q_1 dt, \qquad (14)$$

Since the final angular momentum is also zero, it follows that the time integral of $Q_1(t)$ over the whole path must be zero.

THE ACW GRADIENT ALGORITHM

- It is difficult to use the standard forms of the gradient algorithm with a problem having bounded controls. A new gradient algorithm was developed here to solve such problems, which we call the Adjustable Control-Variation Weight (ACW) algorithm. We describe it briefly here for the case of a single control $u(t)$ and a fixed final time t_f; the more general case is described in Refs. 1 and 2.

- Consider the problem of finding a scalar $u(t)$ to minimize

$$J = \phi[x(t_f), t_f], \qquad (15)$$

where vector $x(t)$ is determined by

$$\dot{x} = f(x,u), \qquad (16)$$

with

$$x(t_o) \text{ specified}, \qquad (17)$$

and

$$\psi_i[x(t_f), t_f] = 0, \quad i = 1,---,p, \qquad (18)$$

subject to

$$|u(t)| \le 1. \qquad (19)$$

- We guess a nominal $u(t)$ that satisfies the constraints (19). Then variations in J and ψ_i are given by

$$\delta J = \int_0^{t_f} H_u^{(\phi)}(t)\,\delta u(t)\,dt, \qquad (20)$$

$$\delta\psi_i = \int_0 H_u^{(i)}(t)\,\delta u(t)\,dt, \quad i = 1,---,p, \qquad (21)$$

where $[H_u^{(\phi)}(t), H_u^{(i)}(t)]$ are the impulse response functions for

$[\delta J, \delta\psi_i]$, determined by integrating the adjoint equations backward over the nominal path (cf. Ref. 2, Section 7.4). We choose $\delta u(t)$ to be proportional to $H_u^{(0)}(t)$ plus a linear combination of the other $H_u^{(i)}(t)$ for the terminal constraints:

$$\delta u(t) = -k[H_u^{(0)}(t) + \nu_i H_u^{(i)}(t)]^T, \tag{22}$$

where k is positive and the ν_i's will be determined to improve satisfaction of the terminal constraints on the next iteration.

- Substituting (22) into (21) gives

$$\delta\psi_i = -P_{ij}\,\nu_j - g_i, \tag{23}$$

where

$$P_{ij} = \int_0^{t_f} \{k * H_u^{(i)}(t) * [H_u^{(j)}(t)]^T\} dt, \tag{24}$$

$$g_i = \int_0^{t_f} \{k * H_u^{(0)}(t) * [H_u^{(i)}(t)]^T\} dt. \tag{25}$$

The linear equations (23) are solved for the ν_i's, which are then used in (22) to form the constrained impulse response function

$$H_u(t) = H_u^{(0)}(t) + \nu_i H_u^{(i)}(t). \tag{26}$$

- The new feature is that we use k = k(u), where u is the nominal u(t), in order to limit $\delta u(t)$ so as to stay inside the constraint boundary and never quite reach it. The k(u) that we use in (24) and (25) is:

$$k(u) = c[1 - |u(t)|], \tag{27}$$

where c is a positive constant (see Fig. 2).

A slight change in k(u) is used in the forward integration to avoid "trapping" the control near a boundary. If $|u(t)| \geq 1 - \epsilon$ and $u(t) * H_u(t) > 0$, then use

$$k(u) = c * \epsilon \;, \tag{28}$$

where $\epsilon \ll 1$ (see Fig. 2). Note (28) is used only when u(t) is far from the optimal u(t).

- The choice of the proportionality constant c has the same difficulty as it does in the conventional gradient procedure. If it is chosen too large the algorithm becomes unstable; if it is chosen too small, the convergence is very slow.

MINIMUM-TIME PATHS FOR A TYPICAL ARM/PAYLOAD COMBINATION

- We shall consider the case where

$$m = m_1 = m_2 = m_p,$$

$$l = l_1 = l_2,$$

$$Q_M = Q_{1M} = Q_{2M},$$

where

$$m_P = \text{mass of payload (the "tip mass").}$$

We shall measure the quantities J_{ik} in units of ml^2, the time in units of the characteristic time $\tau = [ml^2/Q_M]^{1/2}$, the angular velocities in units of $1/\tau$, the torques in units of Q_M, and the distances (D,L) in units of l. Using these units for the case considered we have

$$J_{11} = 7/3, \quad J_{12} = 3/2, \quad J_{22} = 4/3, \tag{29}$$

and

$$|Q_1(t)| \le 1, \quad |Q_2(t)| \le 1. \tag{30}$$

- Fig. 3 shows the minimum-time path for L/l = 0.4. The path of the tip is almost a straight line; there is only one switch in the shoulder torquer and one switch in the elbow torquer. From the integral (14), the shoulder switch must occur at $t = t_f/2$.

- Fig. 4 shows the path for L/l = 2.5. The shoulder torquer switches three times while the elbow torquer still switches only once.

- Fig. 5 shows the path for L/l = 3.7. The tip path is now symmetric with respect to the half-way point. The shoulder torquer still switches three times and the elbow torquer still switches only once. Surprisingly, the time required is less than for L/l = 2.5, where the path is not symmetric.

- Fig 6 shows the path for L/l = 4.0 which is the maximum "reach" of the robot arm. The tip path is still symmetric but, surprisingly, there are now three elbow-torquer switches and only one shoulder-torquer switch. The angular velocity of the lower arm (link 2) is very low as the tip passes through the shoulder, whereas it was very high for L/l = 3.7.

- Fig. 7 shows the minimum time and the switching times vs. distance traveled by the tip. The character of the paths changes from asymmetric to symmetric for L/l > 2.6. The symmetric paths change from "elbow snaps" to "shoulder snaps" for L/l > 3.7, i.e. the maximum angular velocity occurs for the upper arm (link 1) rather than for the lower arm (link 2).

- Fig. 8 shows the optimum initial elbow angle vs. distance traveled by the tip.

IS THE SOLUTION ALWAYS BANG-BANG?

- Most of the numerical solutions converged to paths that looked nearly bang-bang. However, there were two paths (for L/l = 2.5 and 3.3) where the constrained impulse response function H_{u1} (cf. Eqn. (26)) for the shoulder torque was very close to zero for a finite length of time, indicating the possibility of a singular arc or non-uniqueness. These are shown in Fig. 9 along with the shoulder torque determined by the ACW gradient algorithm.

- H_{u1} close to zero indicates that the final time is indifferent to changes in the value of the shoulder torque. In other words, the shoulder torque could be reduced during this period while the elbow torque is still maximum. Whether the shoulder torque should be interior to the constraints (a singular arc), or whether it has some extra bangs that reduce the average torque, was unclear from our numerical results; obviously, the final time will not be affected significantly one way or the other.

- Also, the periods where $H_{\omega 1}$ is close to zero correspond to periods where the arm is folded ($\alpha \approx 180$ deg), which causes Eqns. (5) and (6) to be one-way coupled since $F(\alpha) \approx 0$ and $(J_{22} + J_{12}\cos\alpha) \approx 0$. In Ref. 1 it is shown that this leads to non-uniqueness of one of the controls, which seems to be what is happening here.

- Physically, the extra switches (or the singular arc with intermediate torque) appear to be an attempt to keep the arm close to its folded position, which keeps the moment of inertia near its minimum value.

CHECKING THE SOLUTIONS

- As demonstrated in Fig. 9, the ACW gradient algorithm did not give the bang-bang control histories shown in Figures 3 through 6. These were inferred from the steep (but not abrupt) changes produced after many iterations of the ACW gradient algorithm. They were then checked and "fine-tuned" as described below.

- For the symmetric paths, only the first half of the path has to be calculated. However this means taking $\theta(0)$ not equal to zero; $\theta(0)$ is computed from the expression for the distance traveled for a symmetric path,

$$L = 2\cos\theta(0) + 2\cos[\theta(0) + \alpha(0)], \qquad (31)$$

using the $\alpha(0)$ determined by the ACW algorithm (for the first try). For the paths with one elbow and three shoulder switches ($L/l = 3.3$ and 3.7), the elbow switch and one of the shoulder switches must occur at $t = t_f/2$. The other two shoulder switches must occur symmetrically about $t = t_f/2$. Thus the first shoulder switch was found by linear interpolation so that $\theta = -\pi/2$ when α reached π; the time at which this occurs is $t_f/2$. The whole procedure is then repeated with slightly different values of $\alpha(0)$ to determine the $\alpha(0)$ that minimizes t_f. For the path at $L/l = 4.0$, the first elbow switch was found in the same manner.

- For the asymmetric paths with one elbow and one shoulder switch, ($L/l = .4$ and 1.0), the shoulder switch must be at $t = t_f/2$ because of the integral (14). Thus we pick a t_f and find the elbow switch time by linear interpolation to cause $\omega_2(t_f) = 0$. The corresponding value of D is then calculated. The whole procedure is then repeated with slightly different values of $\alpha(0)$ to determine the $\alpha(0)$ that maximizes D for the chosen t_f.

- No simple procedure was found to check the asymmetric path for $L/l = 2.5$ with one elbow switch and three shoulder switches. It is a four parameter optimization problem for a given t_f, (three switch times and $\alpha(0)$), since one shoulder switch time can be found given the other two shoulder switch times using the integral (14).

DISCUSSION OF SOLUTIONS

- All of the minimum time paths start with a relatively extended arm configuration and the tip motion is directed toward the shoulder. For $L/l \geq 2.0$, the tip path passes through the shoulder. Initially the torques are of opposite sign, causing a flipping motion of the tip.

- These observations are consistent with the fact that the maximum tip acceleration starting from rest is obtained with the extended arm configuration and it is directed approximately toward the shoulder (discussed in more detail in Ref. 1).

- For the longer paths, the arm tends to fold quite rapidly. This is consistent with the fact that the moment of inertia of the arm about the shoulder is minimized when the arm is folded, i.e. when the elbow angle $\alpha = \pi$.

SUGGESTIONS FOR FURTHER RESEARCH

- These solutions yield open-loop torque histories for particular mass characteristics of the arm and tip mass. We would like to have feedback logic and a capability for handling different tip masses. This may be possible using a form of "neighboring optimum" feedback based on small changes in the switch times proportional to small changes in final location desired and small changes in tip mass (cf. Ref. 6).

- It seems likely that for practical pick-and-place robot arms, some form of closed-loop proportional (as opposed to bang-bang) terminal control will be required for precision work.

- It seems possible that a gradient algorithm based on switch times (rather than entire control histories), would find the optimal paths much faster than the ACW gradient algorithm. A problem here is to determine the number of switches in each control.

- The effects cf flexibility (vibrations) must be taken into account for the faster, lighter links envisioned for future robots.

- Minimum-time torque histories for robot arms to follow a desired path are of interest (note the path was open here, only the distance between the end-points was specified).

- Minimum-time paths for robot arms with more than two links are of interest.

- Minimum-time paths to more than one point are of interest.

REFERENCES

(1) Weinreb, A, "Optimal Control with Multiple Bounded Inputs", Stanford Univ. Dept. Aero./Astro. Report 344, October, 1984.

Min. Time --- Robot Arm (p. 8)

(2) Bryson, A.E. and Ho, Y. C., Applied Optimal Control, Hemisphere, Washington, D.C., 1975.

(3) Bobrow, J. E., Dubowski, S., and Gibson, J.S., "On the Optimal Control of Robotic Manipulators with Actuator Constraints", Proceedings of American Automatic Control Conference, San Francisco, 1983, p. 782.

(4) Kahn, M. E., and Roth, B., "The Near-Minimum-Time Control of Open-loop Articulated Chains", Jour. Dynamic Systems, Measurement, and Control, Trans. ASME, Sept. 1971, p. 164.

(5) Lin, C.S., Chang, P.R., and Luh, J.Y.S., "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots", IEEE Trans. on Auto. Control, Dec. 1983, p. 1066.

(6) Foerster, R. E. and Flugge-Lotz, I., "A Neighboring Optimal Feedback Control Scheme for Systems Using Discontinuous Control", Journal of Optimization Theory and Application, Vol. 5, 1971, p.367.

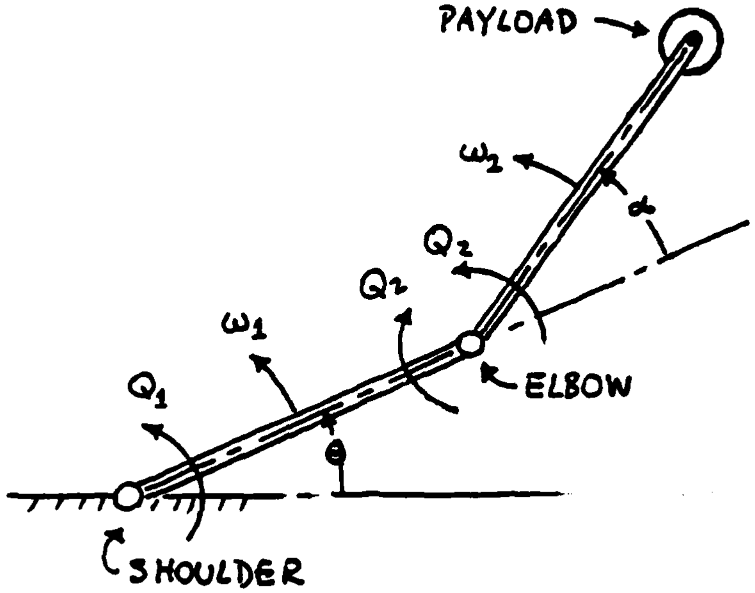


FIG. 1 - TWO-LINK ROBOT ARM - NOMENCLATURE

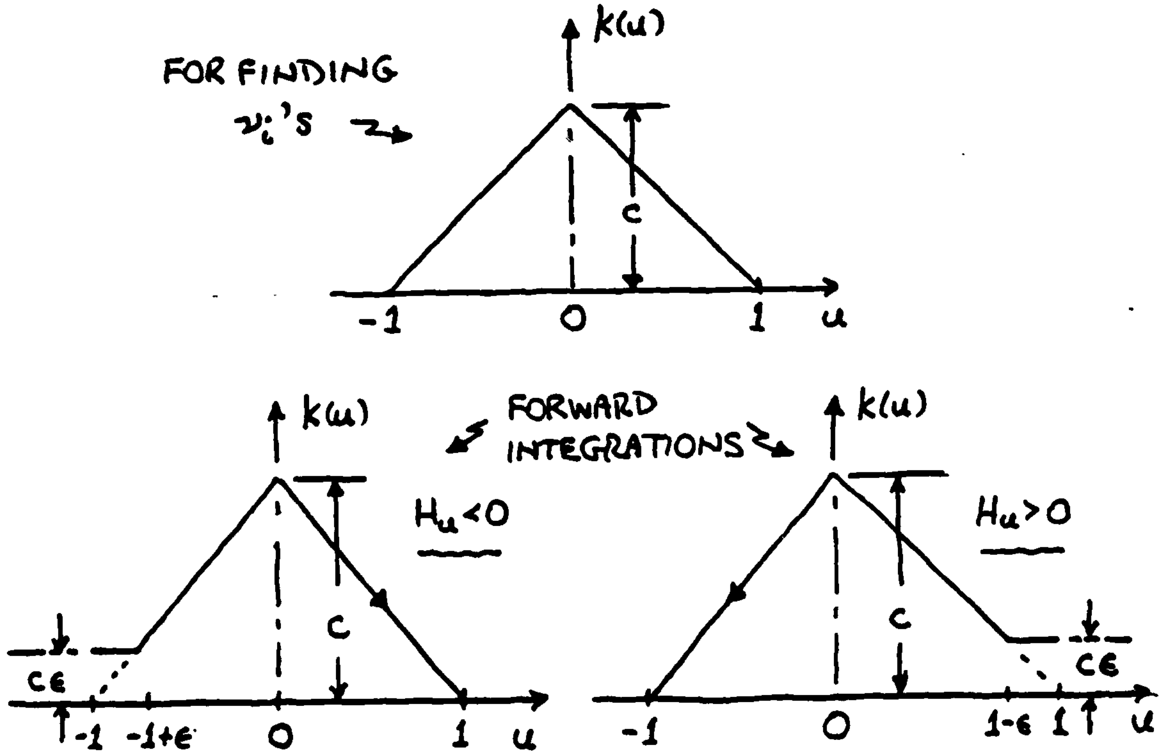
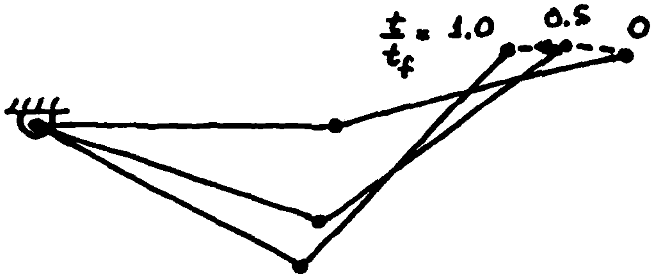
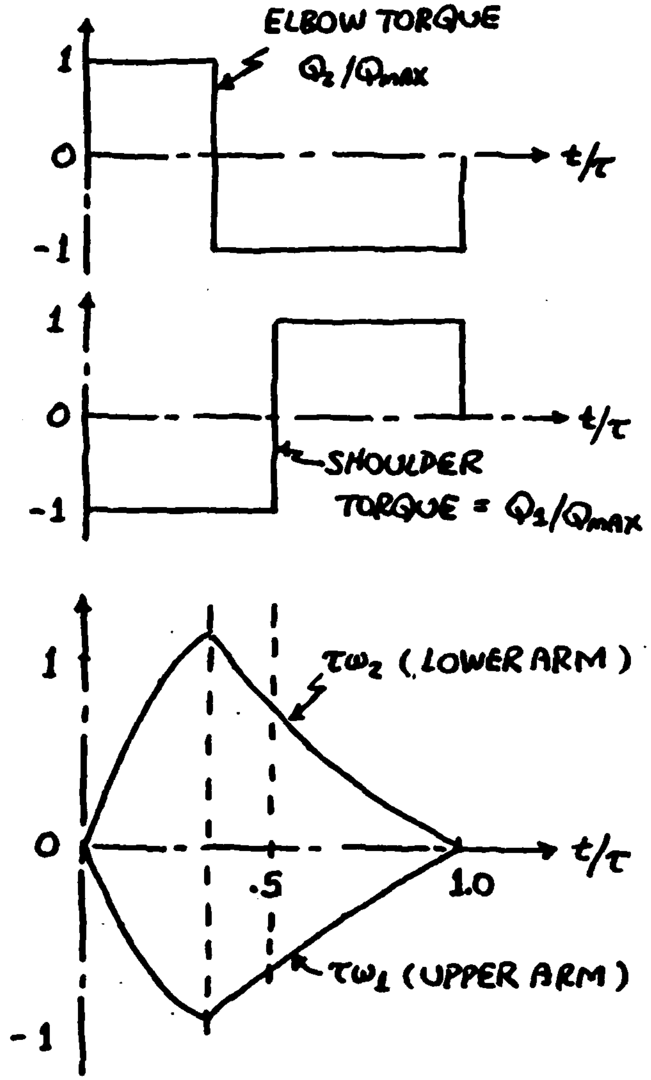


FIG. 2 - CONTROL-VARIATION WEIGHTS



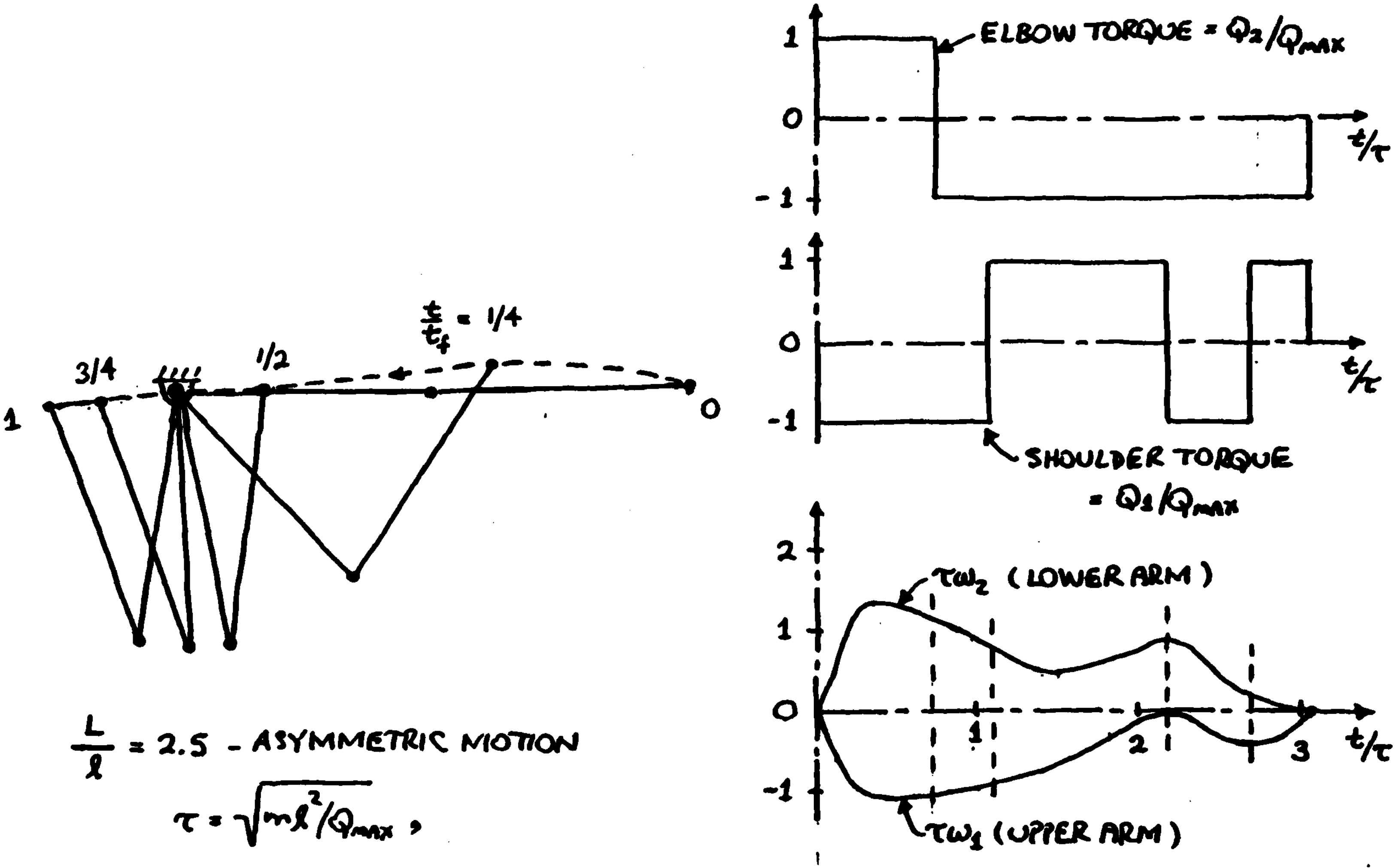


$\frac{L}{\ell} = 0.4$ - ASYMMETRIC MOTION

$\tau = \sqrt{m\ell^2/Q_{max}}$,

$\frac{t_f}{\tau} = 1.01$.

FIG. 3



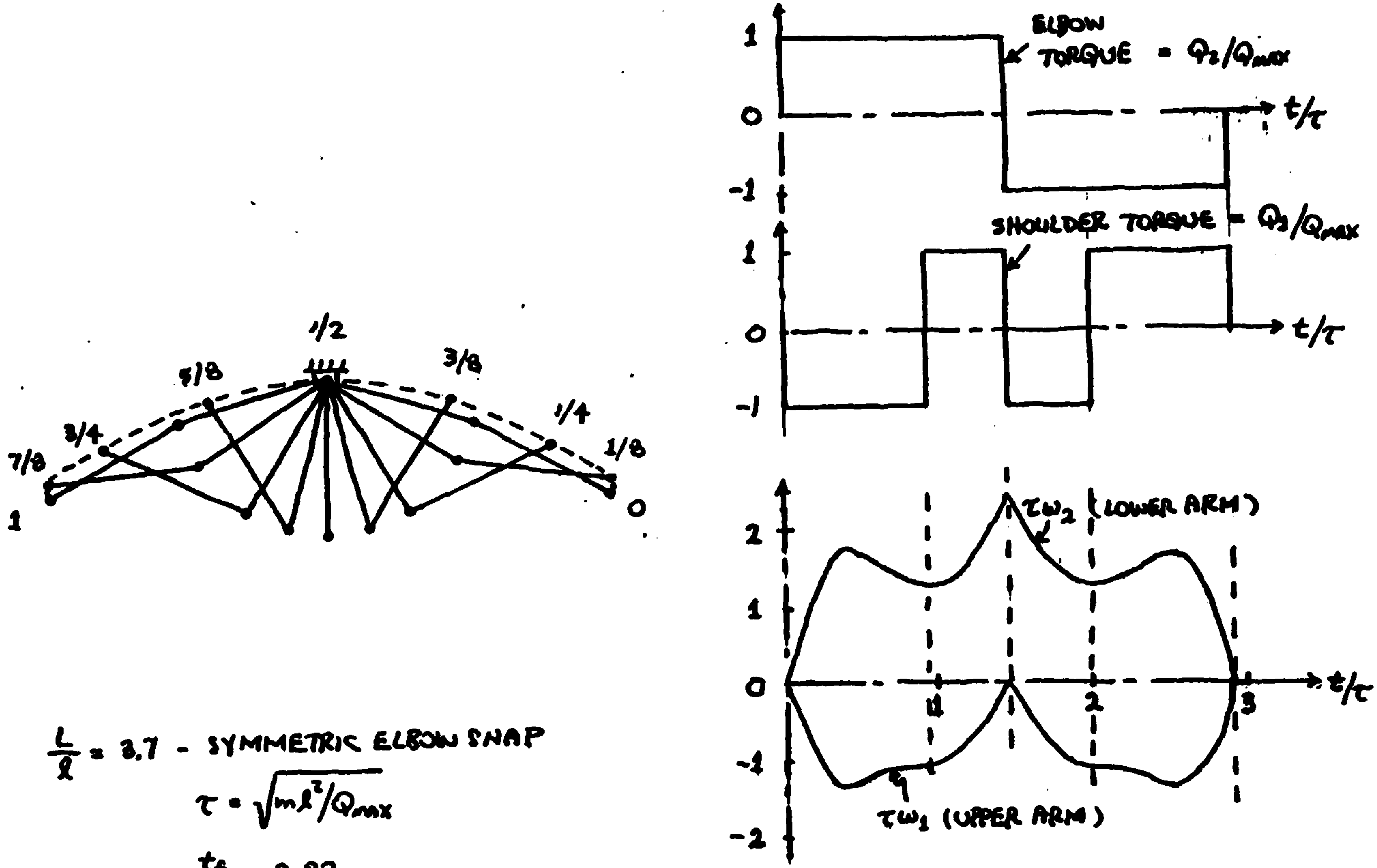
$$\frac{L}{\ell} = 2.5 \text{ - ASYMMETRIC MOTION}$$

$$\tau = \sqrt{m\ell^2/Q_{max}},$$

$$\frac{t_f}{\tau} = 3.05.$$

FIG. 4

⑧

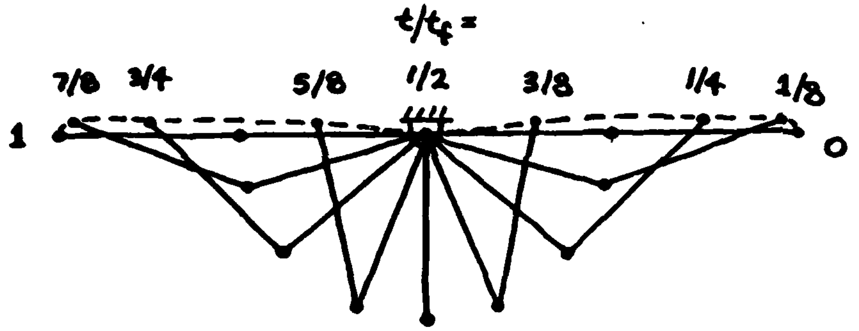


$\frac{L}{\ell}$ = 3.7 - SYMMETRIC ELBOW SNAP

$\tau = \sqrt{m\ell^2/Q_{max}}$

$\frac{t_f}{\tau} = 2.92$

FIG. 5



$\frac{L}{\ell} = 4.0$ - SYMMETRIC SHOULDER SNAP

$\tau = \sqrt{m\ell^2/Q_{max}}$,

$\frac{t_f}{\tau} = 2.97.$

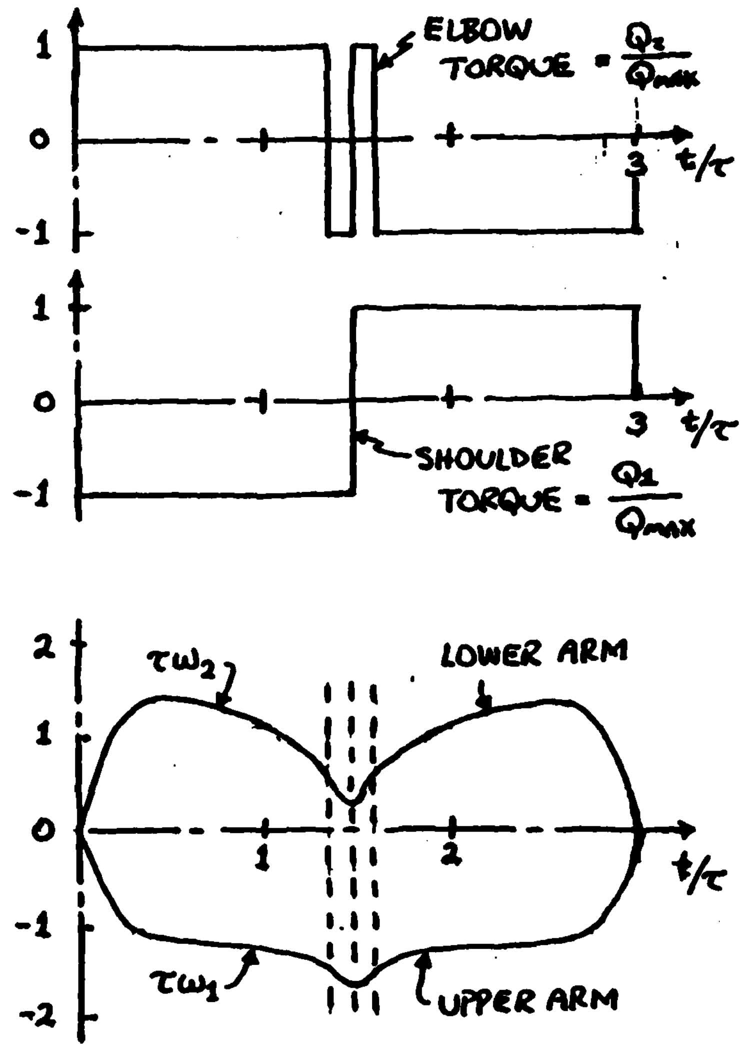
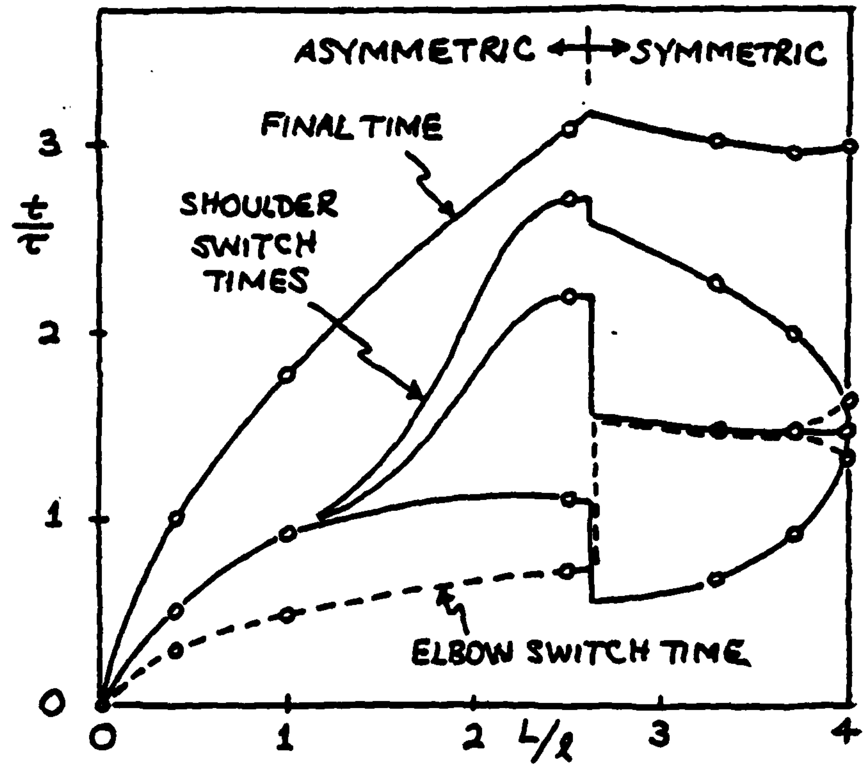


FIG. 6

12/23/84 AEB

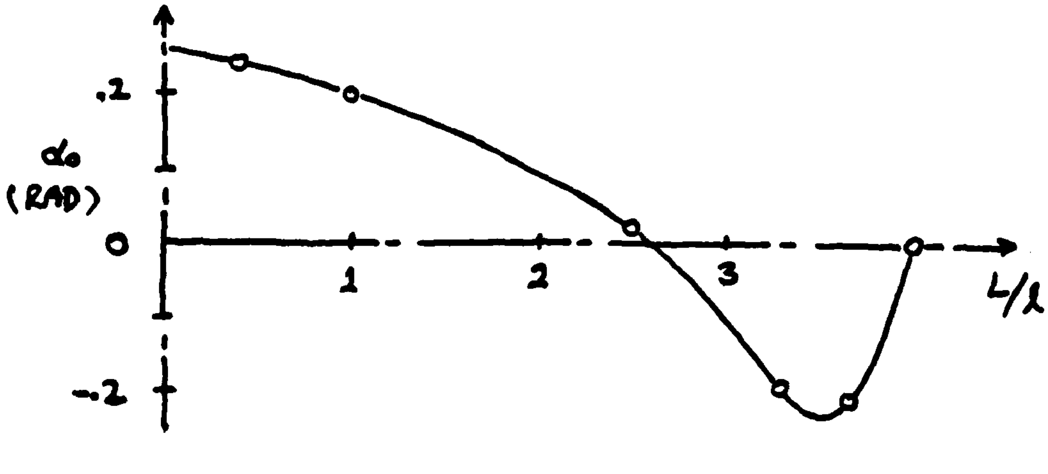
10



$\tau = \sqrt{\frac{ml^2}{Q_{MAX}}}$

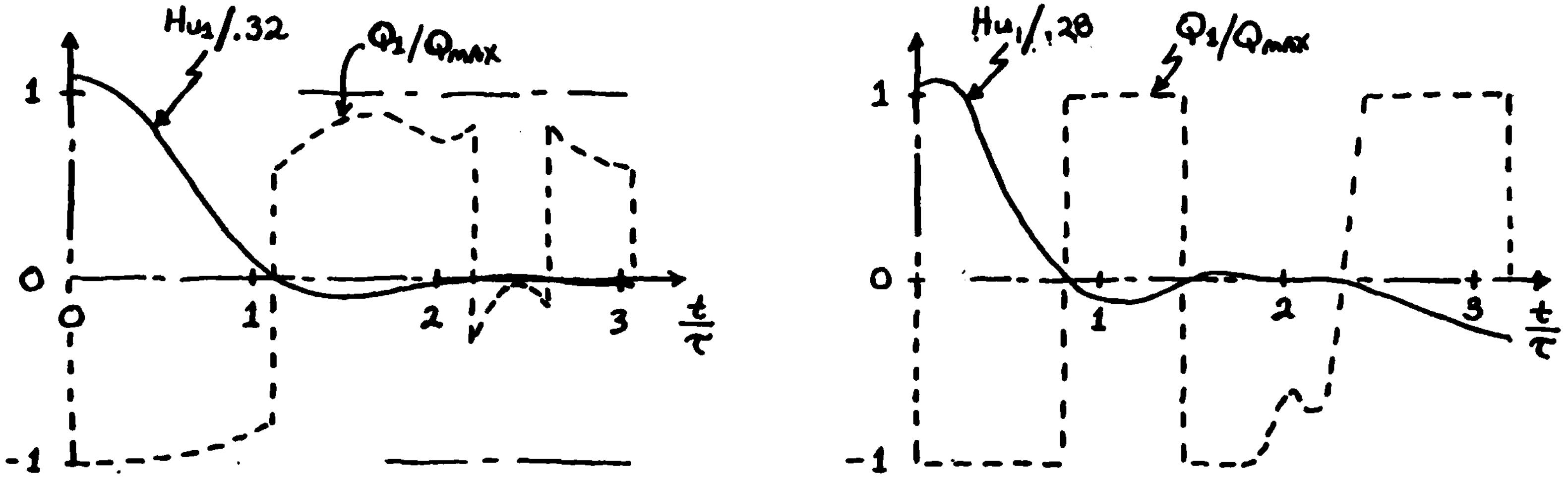
FIG. 7

MINIMUM TIME & SWITCHING TIMES VS. DISTANCE



OPTIMUM INITIAL ELBOW ANGLE VS. DISTANCE

FIG. 8



$\frac{L}{\ell} = 2.5$

$\frac{L}{\ell} = 3.3$

SHOULDER TORQUE Q_1 AND SWITCHING FUNCTION H_{u_1} VS. TIME

FIG. 9

13

END

11-86

DTIC