

MANIPULATION STRATEGIES FOR MASSIVE SPACE PAYLOADS

Semiannual Progress Report
November 15, 1990 - May 14, 1991

Wayne J. Book
Principal Investigator

The George W. Woodruff
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0405

NASA Grant NAG 1-623
(E25-517)

Technical Officer: Donald Soloway

(NASA-CR-188722) MANIPULATION STRATEGIES
FOR MASSIVE SPACE PAYLOADS Semiannual
Progress Report, 15 Nov. 1990 - 14 May 1991
(Georgia Inst. of Tech.) 121 p CSCL 13I

N91-30524
--THRU--
N91-30530
Unclas
0033712

63/37

TABLE OF CONTENTS

	<u>TOPIC</u>	<u>PAGE</u>
SUMMARY		3
RESEARCH TOPIC:	Control of a Flexible Bracing Manipulator	10
RESEARCH TOPIC:	Control of a Small Working Robot on a Large Flexible Manipulator for Suppressing Vibrations	12
RESEARCH TOPIC:	Control Strategies for Cooperating Disparate Manipulators	14
RESEARCH TOPIC:	Motion Planning for Robots in Free-Fall	16
PUBLICATIONS		17

SUMMARY

Motion planning and control for the joints of flexible manipulators has produced delightful results in the planning of joint motions to achieve specified tip motions. This work is performed principally by Dr. Dong-Soo Kwon, who has just successfully defended his Ph.D. Thesis. Dr. Kwon's work took the specified tip motion and used it as the input to the inverse dynamic model which calculates back to the necessary joint motions. Inverse dynamic models for flexible arms have been developed by other researchers, but our work has overcome significant handicaps of their research and revealed some basic properties of the inverse models of these systems.

The flexible manipulator model of tip motion arising from joint actuation is nonminimum phase system. In linearized form this appears as a system with a zero of the transfer function in the right half plane. The inverse of this system thus has poles in the right half plane, a condition usually associated with instability for physical systems which must be causal. The alternative interpretation of right half plane poles is as an acausal system. An acausal system responds to an input before the input is experienced by the system. The difference in the unstable causal and the stable acausal is the region of convergence of the Laplace transform. While all physical systems, including the flexible arm, must be causal, the inverse dynamic system is not a physical system. It represents a calculation of what joint torques would need to be to achieve tip motion. An acausal inverse system tells us that we must begin actuation of the joint before the tip of the robot begins to move. In fact this motion must begin at a time of negative

infinity for the motion to be exactly as specified. An extremely good approximation for a flexible arm requires only a fraction of a second of hub motion before the tip of the arm begins to move. This advance action bends the link into a shape that at $t=0$ retains the tip at the original location. After $t=0$ the tip begins to move, and can in fact move with an almost arbitrary acceleration profile. To make the motion practical a smooth profile is chosen, which none the less effectively uses the maximum torque of the motor. The smoothness of the tip acceleration profile reduces the peak torques required and the order of the inverse dynamic model necessary to accurately command the physical system. Since Dr. Kwon has produced an extremely efficient linearized form of the inverse dynamics algorithm, it can be applied in real time to the tele-operated case, rather than requiring the complete rest to rest trajectory be precalculated with great computing expense. When an obstacle or the object we desire to contact is sensed, the inverse dynamic model plans the remaining trajectory and can complete the motion for the human operator. After contact the flexible states of the arm can also be prescribed to assume some static deflection. This creates a very natural way to apply force with a flexible arm which is superior to commanding joint motion alone, since joint motion is very small for a large tip force, even with a flexible manipulator. Dr. Kwon has experimentally demonstrated his approach for a single link device, and we plan in the coming grant period to apply it to our large two link arm. Mr. J.D. Huggins is developing a Ph.D. proposal to extend Dr. Kwon's work in that direction. Dr. Kwon will be working for Oak Ridge National Laboratories in the coming year on problems of long arms.

Coordinated motion of a tip manipulator and the large flexible manipulator carrying it enables the inertial forces of the tip manipulator to damp the vibrations of the flexible manipulator. This concept has been demonstrated in simulation and a major thrust of the current grant period has been to experimentally demonstrate it. The small arm we use, SAM (Small Articulated Manipulator), is electrically actuated in three degrees of freedom. We are only using two degrees of freedom in the initial studies of motion in a plane. Initial plans for our research used three Motorola 68000 microprocessors operating in parallel to control SAM. The algorithms were programmed and initial motion of SAM was achieved, but the internally developed control hardware proved to be unreliable. Consequently the controller was moved to a commercially available AT bus, 80386 processor. This unfortunate delay has been compounded by mechanical problems as well, which are presently overcome. SAM is again under control under the new processor and the controller algorithms are being studied experimentally. Simulation results show that the inertial forces are effective in damping the flexible vibrations but cannot act with the control authority that joint actuation can. This approach has the advantage that higher bandwidth may be possible with the small arm than with the large arm actuators. Consequently higher modes can be effectively damped.

Disparate cooperating manipulators differ in size and other characteristics. Mr. Jae Lew is studying new ways to take advantage of these potentially complimentary features. Under his scenario the large arm carries the payload and the small arm. The small arm, capable of precise position control, pushes on the environment and deforms

the large arm slightly to achieve the final payload placement. The large arm tries to reduce the force that the small arm must apply to it with a force control. In addition to the greater position resolution naturally available from a small arm, its location near the payload avoids the non-minimum phase problems of the noncolocated large arm joints. Non-collocation leads to stability problems when using high gains to achieve a responsive control. A master/slave control law has been shown in simulation to successfully accomplish the task. The control used in the past year's work acted to decouple the large and small arm behavior. The initial simulations assumed that the force between the two arms could be measured, which may not be the most practical mechanical design. Mechanical design similar to our laboratory large/small arm combination makes measurements at the tip of the small arm, where it contacts the environment, more practical. A new round of studies is underway with a control that does not rely on the decoupling control. We expect the simpler mechanical design from relocating the force transducers to justify the control modifications.

Also explored in the past months is the effect of disturbances generated on the large flexible arm (RALF) as a subsequence of completing its task. An abrasive cutoff saw was mounted on RALF and the spectrum of excitation and the resulting motion was studied experimentally.

Motion planning for robots in free fall has obvious implications for all articulated objects in space. The non-holonomic characteristics of these systems results in reorientation of the vehicle (robot) when the joints move through a series of positions even though the joint angles return to their original values. The bad news is that the

normal operation of the robot in free fall will cause the orientation of the robot to shift with respect to the object it works on. The good news is that desired adjustments in orientation in the robot can be achieved using such cyclical motions without the use of thrusters, based on the same principles that a cat uses to reorient itself when dropped upside down. This project thrust seeks to avoid the problems and capitalize on the advantages with proper planning of robot motion. Mr. Jonathan Cameron, on leave from the Jet Propulsion Laboratory, has completed his Ph.D. thesis proposal and is setting up his planning system. His initial approach is to use symbolic processing of the dynamics equations to establish suitable forms. For a given system a catalogue of motion types will be established. The index to this catalogue is crucial to using the results in a planning system. Motion of a given type will allow reorientation from one class of poses to another. The detailed path of motion to bring one from an arbitrary initial orientation to the desired final orientation will depend on refining the parameters of the motion and perhaps iterative simulation. Consequently, very efficient simulation equations are desirable. Robots on orbit are the most obvious application of the research results, but robots on earth that fall are also candidates. A robot that could use dynamic planning to change its landing orientation would be much less susceptible to damage on landing.

In addition to the four research lines above supported by the NASA grant, **additional work under way** is of interest because of its potential future impact on NASA sponsored studies and because these students are exposed to and benefit from the NASA sponsored research in their own graduate education.

A camera system for end point measurement of the position of the robot relative to retroreflective landmarks is being integrated into our laboratory. This has the potential for providing information to be used directly in feedback control. The retroreflective targets selectively placed in the work space are easily and rapidly distinguished from the background even by our inexpensive camera system. We are currently limited to about 10 position measurements per second, fast enough to refine our commanded position values, but not to perform true end point position feedback. Klaus Obergfell is now writing up his Master's thesis on this topic. He is planning to continue at Georgia Tech on a Ph.D. program, in part because of the experience we were able to provide him in the environment of this NASA grant. The integrated camera and processing system is provided on a beta test basis by Dickerson Vision Technologies, Inc., a Georgia Tech spinoff company.

In addition to the use of RALF as an autonomous arm, it is also commanded as a tele-operator. A small kinematic replica master arm has been created and the corresponding control algorithms created. Application of the improved flexible arm servo controls to the tele-operator case are under study. Dave Magee is midway through a master's thesis on this topic. He has just passed his Ph.D. Qualifying Examination and wishes to continue his research in similar areas.

Research using high speed digital signal processing chips for control is also underway. Speeds of 20 MIPS are available, but to take full advantage of this speed some tailoring of the control algorithms is desirable. Lonnie Love is applying a

TMS320C25 processor to robotic exercise devices. Lonnie has also just passed his Ph.D. Qualifying Examination and is beginning his Ph.D. proposal on related areas.

Finally, the effect of tapering the links of the arm are being considered by Doug Girvin. Doug is looking particularly at the nonminimum phase characteristics of the arm, i.e. the zeros of the transfer function from the joint torque to the tip translation. Since the traditional finite element and even assumed modes technique are approximations that are not verified in terms of their accuracy of zero location, this research is using an exact frequency domain technique. The exact solution is perfectly valid for the linear case only. Numerical evaluation of the exact analytical solution still introduces approximations, of course. By looking at various tapers we expect to be able to modify the minimum phase characteristic of the arm.

RESEARCH TOPIC: Control of a Flexible Bracing Manipulator

RESEARCH ASSISTANT: Dong-Soo Kwon

SHORT TERM OBJECTIVE: Integration of current research work to realize the bracing manipulator

The tip position control of a flexible manipulator can be considered as the most general and complicated problem in the control of robotic manipulators, because it is the tracking control of the nonlinear, noncolocated, nonminimum-phase system with uncertainty. All research results about the flexible manipulator control were integrated to show a control scenario of a bracing manipulator. First, dynamic analysis of a flexible manipulator has been done for modeling. Especially, the nonminimum-phase system characteristics have been studied in detail, and the relation between the nonminimum-phase system, and causal and anticausal systems has been analyzed. Second, from the dynamic model, the inverse dynamic equation is derived, and the time-domain inverse dynamic method were proposed for the calculation of the feedforward torque and the desired flexible coordinate trajectories. Third, a tracking controller is designed by combining the inverse dynamic feedforward control with the joint feedback control. The control scheme has been applied to the tip position control of a single-link flexible manipulator for zero and non-zero initial condition cases. Finally, the contact control scheme was added to the position tracking control. Therefore, a control scenario of a bracing manipulator is provided, and evaluated through simulation and experiment on a single-link flexible manipulator.

PUBLICATIONS

1. Book, W.J. and Kwon, D.S., "Contact Control for Advanced Applications of Light Weight Arms", presented at the Symposium on Control of Robots and Manufacturing, Automation and Robotics Research Institute, University of Texas at Arlington, Texas, November, 1990.
2. Kwon, D.S., "An Inverse Dynamic Tracking Control for Bracing a Flexible Manipulator", Ph.D. Thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, to be published in June, 1991.
3. Kwon, D.S., and Book, W.J., "Tracking Control of Nonminimum-Phase Flexible Manipulator", submitted to the 1991 ASME Winter Annual Meeting.

RESEARCH TOPIC: Control of a Small Working Robot on a Large Flexible Manipulator for Suppressing Vibrations

RESEARCH ASSISTANT: Soo Han Lee

SHORT TERM OBJECTIVE: Development of a robust control law for flexible robot and it's stability analysis

1) Development of a robust control law for flexible robot

In order to achieve the accurate trajectory control of a flexible robot, we need to adopt a nonlinear control law. We develop a robust control law for slow motions. The control law does not need larger velocity gains than position gains, which some researchers need to prove the stability of a rigid robot. Initial experimentation for SAM (Small Articulated Manipulator) show that the control law which use smaller velocity gains is more robust to signal noise than the control law which use larger velocity gains. The control law for the fast motion is strain rate feedback.

2) Stability analysis

When we adopt a composite control law for a two time scaled model, the rule of thumb is to separate the controller bandwidth of the two control laws at least factor of 3. However, we may need a systematic approach for determining the controller gains which are related the bandwidth of the controller. Hence, we analyze the stability of the composite control law, the robust control for the slow motion and the strain rate feedback for the fast control. The stability analysis is done by using the quadratic Liapunov function.

3) Phase lag of input force

The force generated by SAM effect the lower link flexible motion of RALF (Robotic Arm Large and Flexible) through upper link when the SAM is located at the tip of the upper link. One relationship between input force and the flexible motion of the root of upper link is nonminimum phase which some researchers have characterized the time delay. The other relationship is minimum phase because the force is parallel to upper link (no phase lag). These relationships are analyzed by assuming a time delayed model.

We can control the flexible motion of links by relating the input force to the flexible signals which are sensed at the near tip of each link. The signals are contaminated by the time delayed input force. However, we can reduce the effect of the time delayed input force by giving a certain configuration to SAM.

4) Present research

The hardware and software are under development for the real time control of RALF and SAM.

RESEARCH TOPIC: Control Strategy for Cooperating Disparate Manipulators

RESEARCH ASSISTANT: Jae Y. Lew

SHORT TERM OBJECTIVE: Non-Colocated Control of Disturbances of a Flexible Arm

The long term objectives for this research use a small arm (SAM) to compliment a large arm (RALF). As a step on this path the disturbances generated by the tasks of the robot were experimentally studied during this period. In general, disturbances could arise from the motion of the payload (or another arm it carries), from planned or unplanned contact with the environment, or from processes at the tip of the arm (grinding, cutting, spraying, etc). The available equipment made it convenient to study disturbances generated by an abrasive cut off saw mounted on the tip of RALF and used to cut through rods and pipes in the range of 0.5 in. to 1.5 in. The abrasive process was modeled as a friction operation.

The operation of the cut off saw was performed autonomously and under tele-operated control. The behavior was stable in both cases for a P-D joint control algorithm. The critical aspect of operation was low speed on initial contact. The angle of approach relative to the stiffness ellipsoid showed some significance, but since the ellipsoid was not greatly elongated leads to only minor differences for different approach directions. The broad band excitation during the cutting operation is capable of exciting arm natural frequencies to above the third natural frequency of the arm. The direction of cut changed the modes excited due to the change in coupling coefficients.

PUBLICATION

1. Lew, J., Huggins, J. D., Magee, D. and Book, W. , "Experimental Investigations of the Effects of Cutting and on Chattering of a Flexible Manipulator." 1991 ASME Winter Annual Meeting, Atlanta, GA 30332.

RESEARCH TOPIC: Dynamics of Multibody Systems in Free-fall

RESEARCH ASSISTANT: Jonathan M. Cameron

SHORT TERM OBJECTIVE: Complete and Defend Ph.D. Proposal

PERIOD OF COVERAGE: January-March, 1990 (Winter Quarter)

During this period, the researcher completed and presented his Ph.D. proposal. The researcher was only supported during winter quarter. After winter quarter, he was transferred to another project and was not supported by this contract.

The Ph.D. dissertation proposal was titled, "Planning and Executing Motions for Multibody Systems in Free-Fall." It was presented and successfully defended on March 22, 1991. The proposal is attached. Although an extensive literature search was reported in earlier reporting periods, more was done in preparation of the thesis proposal during this quarter.

The researcher also supported activities in the Mechanical Research Building by assisting with computer system management required for several DEC VAX computers.

PUBLICATION

1. Cameron, Jonathan, "Planning and Executing Motions for Multibody Systems in Free-fall", Ph.D. Dissertation Proposal, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

PUBLICATIONS

1. Book, W.J. and Kwon, D.S., "Contact Control for Advanced Applications of Light Weight Arms", presented at the Symposium on Control of Robots and Manufacturing, Automation and Robotics Research Institute, University of Texas at Arlington, Texas, November, 1990.
2. Kwon, D.S., "An Inverse Dynamic Tracking Control for Bracing a Flexible Manipulator", Ph.D. Thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, to be published in June, 1991.
3. Kwon, D.S., and Book, W.J., "Tracking Control of Nonminimum-Phase Flexible Manipulator", submitted to the 1991 ASME Winter Annual Meeting.
4. Lew, J., Huggins, J.D., Magee, D. and Book, W., "Experimental Investigations of the Effects of Cutting and on Chattering of a Flexible Manipulator." 1991 ASME Winter Annual Meeting, Atlanta, GA 30332.
5. Cameron, Jonathan, "Planning and Executing Motions for Multibody Systems in Free-fall", Ph.D. Dissertation Proposal, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

Contact Control for Advanced Applications of Light Weight Arms

Wayne J. Book
Dong-Soo Kwon
The George W. Woodruff School
of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0405

ABSTRACT

Many applications of robotic and teleoperated manipulator arms require operation in contact and non-contact regimes. This paper deals with both regimes and the transition between them with special attention given to problems of flexibility in the links and drives. This is referred to as contact control. Inverse dynamics is used to plan the tip motion of the flexible link so that the free motion can stop very near the contact surface without collision due to overshoot. Contact must occur at a very low speed since the high frequency impact forces are too sudden to be affected by any feedback generated torques applied to a joint at the other end of the link. The effect of approach velocity and surface properties are discussed. Force tracking is implemented by commands to the deflection states of the link and the contact force. This enables a natural transition between tip position and tip force control that is not possible when the arm is treated as rigid. The effect of feedback gain, force trajectory, and desired final force level are of particular interest and are studied. Experimental results are presented on a one link arm and the system performance in the overall contact task is analyzed. Extension to multi-link cases with potential applications are discussed.

1. INTRODUCTION

Control of the impact between the arm's tip and its environment has been largely ignored in prior control research. Since contact forces may be high, it is important to understand the behavior well. This is especially true for operation with repeated contact such as when employing a bracing strategy or for material handling applications. Compared to the behavior before and after the impact, behavior during impact is more complex in several ways: Impacts give high frequency excitation to the arm and to the environment. Modeling these behaviors requires models for the distributed mass and compliance of the system. The contact force exhibits a highly nonlinear nature because it is unidirectional and dependent on relative tip and environment position. However, other complex phenomena important in free motion are not so important in contact control, for example the nonlinearities of the dynamic equations. Since impact intentionally involves low speeds and small position changes the dynamic nonlinearities are often insignificant.

First, this paper presents a practical inverse dynamics approach for planning trajectories that will bring the tip of a flexible manipulator to a precise stop with no overshoot from a

still or moving initial condition. Tip position as a function of time is used as an input to the inverse dynamic equations, which generate the remaining states and joint torque input needed to achieve the desired tip position.

After the planned trajectory is executed, contact with the surface is initiated. Planning of the tip position will allow the contact to be initiated from a point very near the surface. Since the deflection variables of the arm are modeled and controlled along with joint variables, force and tip position control are both naturally accommodated by the same model. A smooth time history of tip force is readily converted to a static displacement of the flexible modes of the links.

The overall system performance involves time penalties, actuator constraints, and force constraints on the tip and the environment. Several characteristics of the application will favor the present more advanced approach. Applications with long arms and penalties on the structural size and mass clearly favor this advanced approach, especially when the surface contacted is prone to damage. Airplane washing, painting and depainting robots have been constructed and have these characteristics. Nuclear waste remediation inside large single shell storage tanks are another clear example, as are required decontamination and decommissioning of nuclear facilities. Manufacturing and construction of large items such as aircraft, power generation equipment, bridges, and buildings also lend themselves to this technology.

The organization of the paper will now be described. The inverse dynamics approach will be presented in summary form with reference to details in section 2. It forms the core of the first phase of contact control, motion control. The second phase is the impact phase, covered in section 3. The final phase of force tracking is then described, followed by a section presenting the experimental results on all phases. Section 5 collects the results from the various phases to show the effect on the overall task performance time. Section 6 presents the issues involved in extending the approach to a multi-link case. The final section on conclusions summarizes the results.

2. AN INVERSE DYNAMICS APPROACH WITH PRACTICAL COMPUTATIONAL EFFICIENCY

To minimize the impact force and the task completion time, the position control should bring the tip near the contact wall precisely without overshoot. A practical and simple inverse dynamic method, which can make that motion possible, was proposed by Kwon and Book [4]. The inverse dynamic method generates the torque profile and all state trajectories that can make

the tip of a flexible manipulator follow the desired trajectory. The trajectories are implemented with a tracking controller. The relative position of the surface is assumed to be poorly known when the motion begins. The initial condition depends on the state of the arm when a position sensor on the tip of the arm determines the distance to a surface in the environment. Upon determining the position of the environment the fast inverse dynamic algorithm allows the remainder of the arm's trajectory to be planned on-the-fly, in time to complete the motion of the tip with precise position and no overshoot. This section describes the modeling of a flexible manipulator and the inverse dynamic method briefly.

2.1. Modeling

A single link flexible manipulator having planar motions is described as shown in Figure 1. The rotating inertia of the servo motor, the tachometer, and the clamping hub are modeled as the hub inertia I_h . The payload is modeled as the end mass M_e and the rotational inertia J_e . Though structural damping exists in the flexible link, it is ignored in modeling.

To derive equations of motion of the manipulator, we describe the position of a point on the beam with virtual rigid body motion and flexible deflection using a Bernoulli-Euler beam model. The virtual rigid body motion is represented by the motion of a moving coordinate, which is attached to the beam. The flexible deflection is described by a finite series of assumed modes with respect to that moving reference frame. The rigid body motion can be defined differently according to the choice of the rigid body coordinate. Several authors [2,3] used the rigid body coordinate that is attached at the base hub or that passes through the center of mass of the beam. In this paper, the rigid body mode coordinate that passes through the end point of the beam is selected [1,4], and the mode functions of pin hub-inertia, pin end-mass boundary conditions are used to describe the displacement of the beam. Because of the rigid body coordinate selected, the end point position of the beam can be expressed by the rigid body mode variable alone. With this simple representation of the end point position, we can derive the inverse dynamics equation easily.

By using Lagrange's equations of motion, the dynamic equation of a flexible manipulator is obtained with generalized coordinates.

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} + \frac{\partial \mathcal{F}}{\partial \dot{q}_i} = Q_i \quad i=0,1,\dots,n$$

$$[M] \ddot{q} + [D] \dot{q} + [K] q = [B] \tau$$

for $i,j = 0, 1, \dots, n$

$$q = \begin{Bmatrix} q_0 \\ \vdots \end{Bmatrix}$$

$$[M] = \begin{bmatrix} M_y & - \\ & \ddots \end{bmatrix}, \quad M_y = \begin{pmatrix} \rho A \int_0^l \phi_i(x) \phi_j(x) dx \\ + I_h \phi_i'(0) \phi_j'(0) \\ + M_e \phi_i(l) \phi_j(l) \\ + J_e \phi_i'(l) \phi_j'(l) \end{pmatrix}$$

$$[D] = c_0 \begin{bmatrix} \phi_i'(0) \phi_j'(0) & - \\ & \ddots \end{bmatrix}, \quad [B] = \begin{bmatrix} \phi_i'(0) \\ \vdots \end{bmatrix}$$

for $i,j = 1, \dots, n$

$$[K] = \begin{bmatrix} 0 & 0 & - \\ 0 & K_y & - \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad K_y = EI \int_0^l \phi_i''(x) \phi_j''(x) dx$$

The mass, stiffness, damping, and input matrices are derived in these general forms. They are valid for different definitions of rigid body modes. The damping matrix represents the joint friction modeled as the viscous friction with the coefficient c_0 .

The dynamic equation can be divided into a rigid body motion part and a flexible motion part as follows:

$$\begin{bmatrix} M_r & M_f \\ M_f^T & M_f \end{bmatrix} \begin{Bmatrix} \ddot{q}_r \\ \ddot{q}_f \end{Bmatrix} + \begin{bmatrix} D_r & D_f \\ D_f^T & D_f \end{bmatrix} \begin{Bmatrix} \dot{q}_r \\ \dot{q}_f \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_f \end{bmatrix} \begin{Bmatrix} q_r \\ q_f \end{Bmatrix} = \begin{bmatrix} B_r \\ B_f \end{bmatrix} \tau \quad (1)$$

where $q_r = q_0$; rigid body coord.,

$$q_f = \begin{Bmatrix} q_1 \\ \vdots \end{Bmatrix}; \text{ flexible mode coord.}$$

For a state space form, we obtain the following dynamic equation.

$$\dot{X} = \begin{bmatrix} 0 & I \\ M^{-1}K & M^{-1}D \end{bmatrix} X + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} \tau$$

$$Y = [C]X + [F]\tau \quad (2)$$

$$\text{where } X = \{q_r, \dot{q}_r, \dot{q}_f, q_f\}^T$$

$$= \{q_0, \dot{q}_0, \dot{q}_1, q_1, \dots\}^T$$

2.2. Inverse Dynamic Equations

From the above the direct dynamic equations, we will derive the inverse dynamic equation. Eqn. (1) can be written in two parts.

$$[M_r]\ddot{q}_r + [M_{rf}]\ddot{q}_f + [D_r]\dot{q}_r + [D_{rf}]\dot{q}_f + [B_r]\tau \quad (3)$$

$$[M_{rf}]^T\ddot{q}_r + [M_{ff}]\ddot{q}_f + [D_{rf}]^T\dot{q}_r + [D_{ff}]\dot{q}_f + [K_{rf}]q_r + [K_{ff}]q_f = [B_{rf}]\tau \quad (4)$$

From Eqn. (3), torque is expressed as

$$\tau = [B_r]^{-1} \{ [M_r]\ddot{q}_r + [M_{rf}]\ddot{q}_f + [D_r]\dot{q}_r + [D_{rf}]\dot{q}_f \} \quad (3a)$$

Substitution of above Eqn. (3a) into Eqn. (4) gives the following relations between the flexible coordinate q_f and the rigid body coordinate q_r .

$$\begin{aligned} [M_r]\ddot{q}_r + [D_r]\dot{q}_r + [K_r]q_r &= [B_{rr}]\ddot{q}_r + [B_{rf}]\ddot{q}_f \\ \text{where } [M_r] &= \{ [M_r] - [B_r][B_r]^{-1}[M_{rf}] \} \\ [D_r] &= \{ [D_r] - [B_r][B_r]^{-1}[D_{rf}] \} \\ [K_r] &= [K_r] \\ [B_{rr}] &= \{ [B_r][B_r]^{-1}[D_{rr}] - [D_{rf}]^T \} \\ [B_{rf}] &= \{ [B_r][B_r]^{-1}[M_{rf}] - [M_{rf}]^T \} \end{aligned} \quad (5)$$

From Eqn. (4), the acceleration of flexible coordinate is expressed as follows:

$$\begin{aligned} \ddot{q}_f &= -[M_{ff}]^{-1}[M_{rf}]^T\ddot{q}_r - [M_{ff}]^{-1}[D_{ff}]^T\dot{q}_r \\ &\quad - [M_{ff}]^{-1}[D_{rf}]^T\dot{q}_f - [M_{ff}]^{-1}[K_{ff}]q_f + [M_{ff}]^{-1}[B_{ff}]\tau \end{aligned} \quad (6)$$

Substitute this Eqn. (6) into Eqn. (3), then we will get the following Eqn.

$$\tau = [C_{rr}]\ddot{q}_r + [C_{rf}]\dot{q}_r + [F_{rr}]\dot{q}_r + [F_{rf}]\dot{q}_f \quad (7)$$

$$\begin{aligned} \text{where } [G] &= \{ [B_r] - [M_{rf}][M_{ff}]^{-1}[B_{rf}] \}^{-1} \\ [C_{rr}] &= [G] \{ -[M_{rf}][M_{ff}]^{-1}[K_{ff}] \} \\ [C_{rf}] &= [G] \{ [D_{rf}] - [M_{rf}][M_{ff}]^{-1}[D_{ff}]^T \} \\ [F_{rr}] &= [G] \{ [D_{rr}] - [M_{rf}][M_{ff}]^{-1}[D_{rf}]^T \} \\ [F_{rf}] &= [G] \{ [M_{rf}] - [M_{rf}][M_{ff}]^{-1}[M_{rf}]^T \} \end{aligned}$$

If Eqn. (5) and Eqn. (7) are represented in a state space form, the inverse dynamic equations will be written in this form.

$$\text{Let } X_i = (q_r, \dot{q}_r)^T, \quad q_r = (q_r, \dot{q}_r)^T$$

$$\dot{X}_i = \begin{bmatrix} 0 & I \\ M_i^{-1}K_i & M_i^{-1}D_i \end{bmatrix} X_i + \begin{bmatrix} 0 & 0 \\ M_i^{-1}B_{iu} & M_i^{-1}B_{iu} \end{bmatrix} \dot{q}_r \quad (8)$$

$$\tau = [C_{iu}, C_{ou}]X_i + [F_{iu}, F_{ou}]\dot{q}_r$$

$$\dot{X}_i = [A_i]X_i + [B_i]\dot{q}_r$$

$$\tau = [C_i]X_i + [F_i]\dot{q}_r$$

The inverse dynamic equation is obtained in this simple form. By the way, it seems at first to be impossible to integrate that equation because the inverse system matrix A_i has positive real valued poles (which came from the positive zeros of the transfer function of the direct dynamic system) as well as negative poles. However, if we relax the solution range of that equation to include noncausal solutions, we can obtain a unique stable solution by integrating this 1st order differential equation.

The inverse dynamic system equation can be divided into the causal part and the anticausal part by using a similarity transformation as follows.

[T]: Orthogonal transformation matrix

$$\begin{aligned} X_i &= [T]P_i \\ &= [T_c, T_{ac}] \{ P_c, P_{ac} \}^T \end{aligned}$$

where $X_i = (q_r, \dot{q}_r)^T$, T_c 's basis includes the eigenvectors which have negative eigenvalues and T_{ac} is made of the eigenvectors of positive eigenvalues.

$$[T]^{-1}A_i[T] = \begin{bmatrix} A_{kc} & 0 \\ 0 & A_{kac} \end{bmatrix}$$

$$\begin{bmatrix} \dot{P}_c \\ \dot{P}_{ac} \end{bmatrix} = \begin{bmatrix} A_{kc} & 0 \\ 0 & A_{kac} \end{bmatrix} \begin{bmatrix} P_c \\ P_{ac} \end{bmatrix} + \begin{bmatrix} B_{kc} \\ B_{kac} \end{bmatrix} \dot{q}_r \quad (9)$$

$$\begin{bmatrix} \tau_c \\ \tau_{ac} \end{bmatrix} = \begin{bmatrix} C_{kc} \\ C_{kac} \end{bmatrix} \begin{bmatrix} P_c \\ P_{ac} \end{bmatrix} + \begin{bmatrix} 1/2 F_{ic} \\ 1/2 F_{iac} \end{bmatrix} \dot{q}_r$$

$$\tau = \tau_c + \tau_{ac}$$

Such a coordinate change decouples the inverse system into two subsystems of Eqn. (9). The new variable P_c represents the coordinates of the causal system, and the P_{ac} represents the anticausal system.

For a given end point trajectory of Figure 2, the causal part torque is obtained by integrating the causal part inverse dynamic equations forward from the initial time of the trajectory. Then, the anticausal system equations must be integrated backward from the final time of the trajectory. The total torque, which is the output of these equations, is obtained by adding the both part torques as shown in Figure 3. As additional outputs, the reference trajectories of the output of Figure 4 were obtained from the

integration of the inverse dynamic equation.

3. CONTACT CONTROL

Impact Force Impact is a high frequency phenomena dependent mainly on the local parameters at the point of impact, especially the tip velocity and the elastic, mass and damping properties of the tip and the environment. The delay inherent in the tip response any elastic link will defeat a strategy to reduce impact force by tip force feedback control of joint actuators located at the other end of the link. Only for an extremely soft environment will the response be fast enough. Low impact force and quick task completion requires accurate tip placement before impact and low velocity upon impact. Another approach that deserves attention is to cushion the impact between arm and environment with a "shock absorber" system. This might be designed in conjunction with the improved control approach.

To study impact, the flexible model of the link must be supplemented with a model of the surface compliance and damping. The dynamic model of the arm remains essentially the same as without contact, except the contact surface and the force sensor must be incorporated as additional stiffness and damping matrices. The tip position is readily available with the choice of coordinates used in the model. The tip is allowed to contact a surface modeled by a spring and damper. The tip is free to rebound from the surface as dictated by the combined dynamics, where upon the contact force drops to zero.

Figure 6 is an overlay of the tip forces for various tip approach velocities. The initial impact results in rebound, loss of contact, and further impacts. Higher tip approach velocities mean higher impact forces but the change in velocity has minimal effect on the time between initial contact and the occurrence of the peak force which typically occurs on the first or second impact. Figure 5 shows an expanded view of a single impact for various surface stiffnesses and displays the time scale of these impacts, which is of the order of .001 seconds. To reduce these impact forces with feedback force control, the control action would have to be taken and propagate to the tip in less than 1/2 the duration of the impact. This is clearly infeasible for the light weight arm, whose natural frequency is only 2.1 Hz. Even supposedly rigid arms will not have the speed of response needed to react in the required time because of the actuator limit. This leaves the designer with two alternatives:

- 1) reduce the speed on impact, or
- 2) modify the compliance at the point of contact.

Either of these alternatives might be useful in particular situations.

A "shock absorber" could be mounted on the tip to reduce the peak force. While the peak force has been reduced, the total momentum transferred to the environment would remain about the same, which may indicate problems for other parts of the system. If enough travel of the shock absorber is permitted, the arm might now be able to respond to the initial impact. The soft suspension allows the tip to be used as a proximity sensor. Engineering this solution might limit the utility of the arm, however.

The solution discussed in this paper is to reduce the speed on impact by carefully controlling the tip position. This requires that the planned motion trajectory bring the tip very near the surface, or else the low approach velocity will delay the progress of the task. The inverse dynamics approach allows this to be done for the flexible arm. The position of the surface can be updated as it comes within range of a proximity sensor and the trajectory can

be modified en route. It is also possible to plan for a non-zero tip velocity at the end of the motion trajectory which becomes the approach velocity.

Force Tracking Since the deflection variables of the arm are modeled and controlled along with joint variables, force and tip position control are both naturally accommodated by the same model. A smooth time history of tip force is readily converted to a static displacement of the flexible modes of the links. These displacements are commanded to achieve the desired tip force. The time allowed to reach the final desired force and the force tracking gains are important parameters in obtaining successful force control.

The force trajectory used in this paper is a cubic polynomial fitted between zero and the desired final force. The force control mode is entered upon the initial contact. From the known desired force the necessary flexible state variables are computed, assuming in our case that the system is in static equilibrium. (Refined assumptions regarding the dynamics of the constrained system are, of course, possible. Referring to Eqn.2, one can see that the flexible variables q_f are found to be

$$S_{t_i} = \frac{HL}{2EI}(1-X_i) F_c$$
$$q_f = C^{-1} \{S_{t_i}\}$$

where F_c is the desired contact force, S_{t_i} is the strain at X_i , and C^{-1} is the submatrix of the output matrix C .

As can be seen from Figure 6, the approach velocity has negligible effect on the force tracking and regulation. The desired force level also has little effect. The gains used in the force tracking algorithm do affect force tracking, however, as does the force trajectory as illustrated by shortening the time for achieving the desired final force level.

The force control algorithm prescribes desired values for the contact force, the flexible states, q_f , and the joint variables. The tip contact force is fed back with a proportional control, and the flexible states and the joint variable is fed back with a PD control. Integral action could be added to eliminate the steady state error of imperfections such as friction. These imperfections are not included in the model, so the simulation shows no problems in this regard. This is not the case for the real system as observed later in the physical experiments. Increases in proportional gain are shown in Figure 7. Increased gains ultimately lead to instability.

The force trajectory determines the abruptness of the changes that the flexible variables will be tracking. Figure 8 shows the tip forces resulting from shorter trajectories leading to the same final force value. The force trajectory is initiated on the first contact. By the time the arm has returned for the second contact the force trajectory may be half completed, as in the first case in Figure 8. This leads to very high force levels on the second contact. Even for the case of zero approach velocity, with the tip in zero force contact with the environment, the sudden force command can cause the tip to leave the surface due to a nonminimum phase reverse action.[2]

4. EXPERIMENTAL RESULTS

The contact controller was applied to a single link experimental arm to verify the results. The parameters of the system are the same as for the simulated system. These parameters are shown in Table 1, which references the description

of Figure 1. Figure 9 shows schematically the control system implemented. The measurements in the vector Y include joint position and velocity and strain at the base of the link and near its mid-point. The results of position and contact force control are represented in Figure 11,12,13. The desired, simulated, and experimentally measured contact force, tip position, and link base strain are displayed. Tip position, displayed in Figure 11, is calculated based on the joint position and the strain values, since tip position measurements are not directly available in this experiment.

Table 1. Physical properties of the flexible manipulator

Link	EI: stiffness lb _f in ²	4120	H: thickness in	3/16
	Ro A: unit length mass lb _f s ² /in ²	2.54E-4 * 0.14	L: length in	47
Tip mass	Me: mass lbm	0.1	Je:rot. inertia	3.8E-5
Hub	Ih: rot. inertia lb _f s ² in	0.14		

The experiments duplicate to acceptable accuracy the simulation results with the exception of the steady state error in the contact force of Figure 12. This steady state error is apparently due to the substantial static friction in the motor. The primary imperfection of the model used in simulation and controller design is the absence of friction. The friction is compensated for in the position control phase of the task. The friction model is not able to compensate for both the dynamic friction during motion and the static friction when the arm is almost stationary. Since there is a limit on the proportional gains for a stable response, an integral action is proposed to improve steady state error.

Finally, the strain at the base, the desired, the simulated, and the experimental values, are shown in Figure 13 for all three phases, position control, approach and impact, and force trajectory control. The motion in this case is a 70 degree rotation of the 47 inch long beam, that is an arc length of about 57 in.

5. SYSTEM PERFORMANCE

While an optimization of the overall performance has not yet been attempted, the trends permitted by the planning and control approach described here are clear. The complexity resulting from the additional sensors needed, the complexity of the control, and the need for reasonably accurate models of the arm are all reasons to use conventional approaches to contact control when they will suffice.

The task time is broken into several parts:

- 1) position tracking
- 2) position regulating (tip position settles)
- 3) velocity control (move toward impact)
- 4) contact force tracking, and
- 5) contact force regulating (force reading settles)

The time for each part is shown in Figure 10. Position tracking is reduced by controlling the arm as a flexible system. Faster moves are then possible. Since the inverse dynamic approach results in zero overshoot, the position regulating part is reduced to almost zero. Velocity control time is reduced since the inverse dynamic approach avoids overshoot. Contact force tracking time

may be increased with the proposed method to bring about a reduction in the regulation or settling time.

6. EXTENSION TO THE MULTI-LINK CASE

The scale of the initial simulations and experiments justifiably lead one to question the applicability of the approach used here to multiple links and axes of motion. The most critical questions surrounding the technique are:

- 1) Can the technique be extended to tip motions in three dimensions, and by what techniques?
- 2) The single link case is well approximated by a linear model while an articulated multi-link arm has prominent dynamic nonlinearities. How limiting is the dependence on linear techniques?
- 3) The number of states of an arm grows with the number of axes. A six degree of freedom robot would have at least three axes and links to be considered for positioning a contact point. How does the computational complexity scale with the number of degrees of freedom.

The approach, even in a single degree of freedom bears promise for high performance servo applications. In addition, there is no conceptual difficulty in extending the techniques from one dimension to three dimensions. The mode shapes used to model the flexible beam should continue to be of the same boundary conditions to enable position to be found as a function of the rigid variable alone. Other issues, such as a flexible equivalent of hybrid control appear possible, but the detailed analysis has not been carried out.

The dynamics of a multi-link arm involve nonlinear terms not found in the single link case. These nonlinearities are the second order velocity terms (Coriolis and centrifugal forces), the variations in the inertia matrix (joint angle dependent), and the variations in the gravity forces (joint angle dependent). The contact control problem arises under the relatively low velocity situation near contact. The square of the small velocities will therefore be even smaller. Since the velocities are small, the position changes will also be small. A long slender arm supplements these arguments by moving small amounts in terms of joint angles for a large tip motion. Our large two degree of freedom experimental arm, for example, has been verified to show little nonlinear behavior even at maximum joint velocity.[5]

The final issue in the extension is the calculation time. As a last resort it is possible to calculate the motion trajectory off-line before hand. The current trajectory is calculated before the motion on a MicroVAX II computer, roughly a 1 MIP machine, in 1.01 seconds for 57 inch long tip trajectory (1.066 sec travel time). RISC processors and DSP chips with 20 MIP performance are commonplace and inexpensive. When progressing from one link to three links, how does the computational complexity scale? For the one link case of this paper, six states are found in the flexible model. The cost effective method of inverse dynamics reduces this to four because the nominal (rigid body) dynamics are specified by the desired tip position. The remaining fourth order system is separated into two second order system by the causal and anti-causal property of the solution. The doubling of states causes twice as many state equations, each twice as complex. Hence the computational complexity grows as the square of the degrees of freedom. The increase by a factor of 9 in computational complexity is more than offset by the possible increase by a factor of 20 to 40 in computational power readily

available. It is also possible that the algorithms themselves could be further streamlined.

7. CONCLUSIONS.

An effective scheme for contact control of a flexible link arm has been derived, analyzed and experimentally verified. It provides fast tip motion with no overshoot, impact with modest impact forces, and force tracking up to a desired constant force level. There are several avenues to improve and extend the method. The primary one discussed here is the extension of the technique to multiple degrees of freedom. It appears that this extension is feasible with current technology.

Applications of the approach to aircraft servicing, nuclear waste remediation, and manufacturing of large parts seems to be among the most suitable applications at this time, since long, light weight arms are involved and there is a need for repeated contact with the work piece.

ACKNOWLEDGEMENTS

This research was made possible in part through the support of NASA Grant No. NAG 1-623.

References

1. Asada, H. and Ma, Z., "Inverse Dynamics of Flexible Robots", *Proc. of the American Control Conference*, pp.2352-2359, 1989.
2. Cannon, R. and Schmitz, E., "Initial Experiments on End Point Control of a Flexible One-Link Robot", *International Journal of Robotics Research*, Vol.3, No.3, pp.49-54, 1984.
3. Hastings, G. and Book, W. J., "Experiments in the Optimal Control of A Flexible Manipulator", *Proc. of the American Control Conference*, pp.728-729, Boston, 1985.
4. Kwon, D.-S. and Book, W.J., "An Inverse Dynamic Method Yielding Flexible Manipulator State Trajectories", *Proc. of the American Control Conference*, Vol.1, pp.186-193, San Diego, May, 1990.
5. Lee, J.W., "Dynamic Analysis and Control of Lightweight Manipulators with Flexible Parallel link mechanisms", *Ph.D. thesis*, Georgia Institute of Technology, 1990

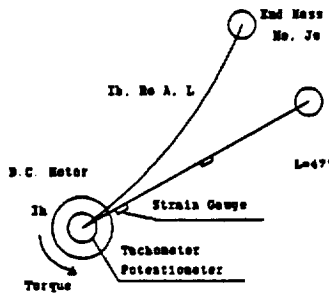


Figure 1. A single link flexible manipulator

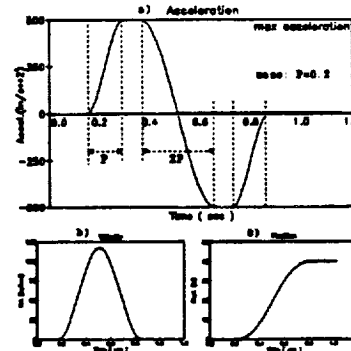


Figure 2. Desired end point trajectory
a)Acceleration, b)Velocity, c)Position

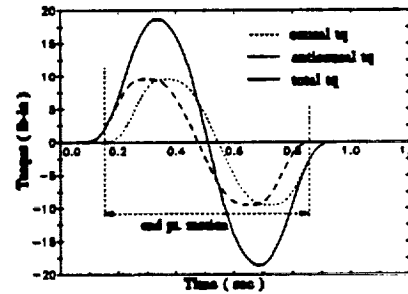


Figure 3. Calculation of torque with the inverse dynamic method

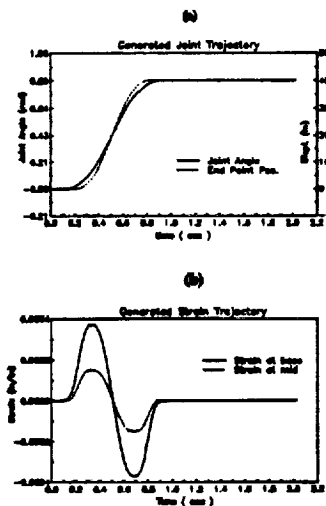


Figure 4. Trajectory generation by using the inverse dynamic method: a) Joint angle, End point position, b) Strain at base and mid-point

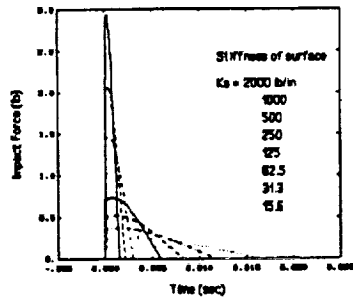


Figure 5. Impact forces for different contact surfaces

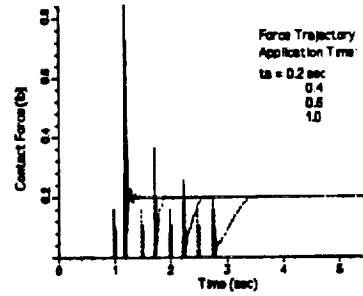


Figure 8 Contact force for desired force trajectories with various application times

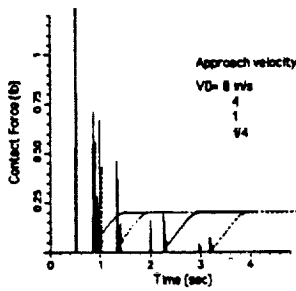


Figure 6 Contact force for various tip approach velocities

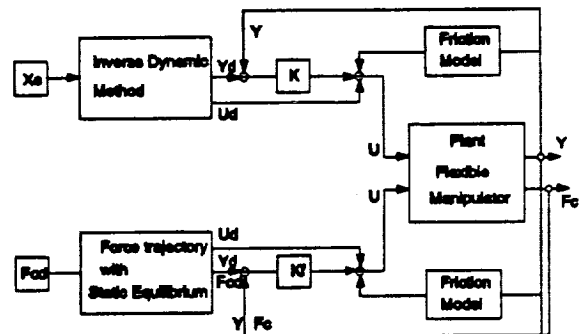


Figure 9. The control scheme of the experiment

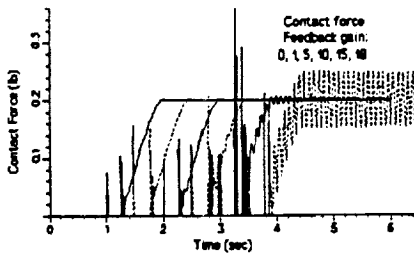


Figure 7 Contact force for different force feedback gains

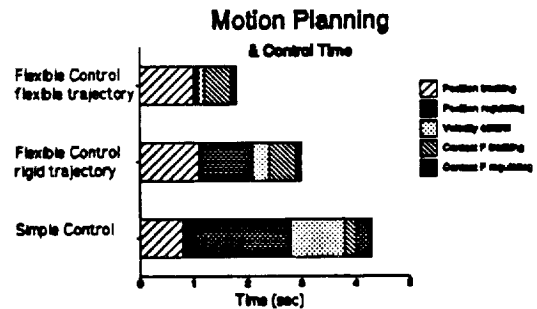


Figure 10. Comparison of task completion times

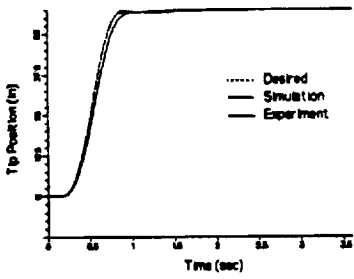


Figure 11 The experimental results of the tip position

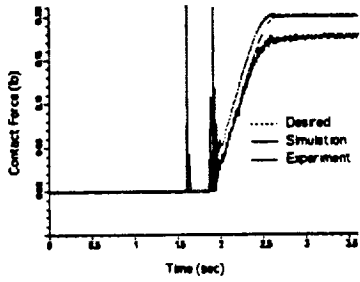


Figure 12 The experimental result of contact force

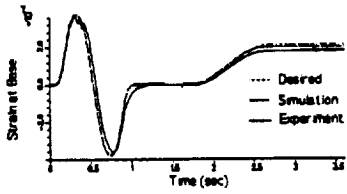


Figure 13 The experimental result of the strain at the base

**An Inverse Dynamic Tracking Control for
Bracing a Flexible Manipulator**

SUMMARY

A manipulator can improve its positioning accuracy in a manner similar to the human who braces his wrist for a more accurate finger motion. Reduction of positioning uncertainty by bracing can increase practical applications of flexible large-workspace manipulators. The goal of this research is to realize the bracing manipulator, which is flexible. In order to brace the manipulator against a surface of the environment, a tip position tracking control becomes essential to avoid large impacts on contact, and to control the force after contact. If we consider the characteristics of the flexible manipulator, the control task may be characterized as a tracking control of a nonlinear, noncolocated, nonminimum-phase system with uncertainties.

First, the author introduces the staged positioning concept and provides a conceptual background of the bracing strategy. Second, in order to understand the dynamics and determine the control strategy, a single-link flexible manipulator is modeled by the assumed-modes method, and the validity of the model is verified with experimental results. Several techniques to obtain more accurate mode functions are discussed and compared. Third, a time-domain inverse dynamic method is proposed to cancel out the nonminimum-phase positive zeros, which cannot be canceled by feedback controls. The trajectory is also planned to use the maximum capacity of an actuator and to avoid exciting flexible vibrations. Fourth, the tracking controller is designed to control the end-point of a flexible manipulator without any overshoot or residual

vibrations. Its perfect tracking or asymptotic tracking performance is analyzed and discussed for zero and non-zero initial condition cases. Finally, impact phenomena are investigated, and the contact force control of flexible bracing manipulator is presented to set up a control scenario of a bracing manipulator. The proposed control scheme is implemented on a single-link flexible manipulator, and analyzed and evaluated through simulations and experiments.

Tracking Control of a Nonminimum-Phase Flexible Manipulator

Dong-Soo Kwon

Wayne J. Book

Flexible Automation Laboratory

The George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology

Atlanta, GA 30332

Abstract

Perfect tracking or asymptotic tracking of the tip position of a flexible manipulator, which is a typical nonminimum-phase system, has been very difficult because of the positive zeros of the nonminimum-phase system and the lack of desired trajectories of flexible coordinates. This paper presents a tracking control scheme combined with the inverse dynamic feedforward control. The inverse dynamic method calculates the feedforward torque that cancels poles and zeros of the nonminimum-phase system. It also generates the desired flexible coordinate values, which match equivalent to the tip position trajectory dynamically. The feedback loop achieves tracking capability with the calculated desired flexible coordinate trajectories. The control scheme has been applied to the tip position control of a single-link flexible manipulator for zero and non-zero initial condition cases. The non-zero initial conditions of the system states are divided into three components of rigid body, causal and anticausal parts. The rigid body

component is used for the desired tip trajectory planning and the other components of these are used separately for the calculation of the feedforward torque of causal and anticausal parts. Through simulation and experiment, we explore the effectiveness and limitation of this method for moving non-zero initial condition cases.

1. Introduction

More and more today, robot manipulators are being used in the manufacturing processes, such as welding, material handling, and assembly. An increasing number of applications demand the capability of larger work-space, more accurate positioning, faster motion, less power consumption, and greater payload. The lightweight, large work-space manipulator has intrigued the engineers who have to deal with large objects, such as the space station assembly, large structure welding, airplane assembly, and nuclear waste handling. However, the large lightweight structure results in inherent flexibility, which induces complex dynamics in a manipulator.

The dynamic model is infinite dimensional due to infinite flexible modes. It is nonlinear, and always has uncertainty due to the unmodeled dynamics. The tip position control with the joint torque is usually nonminimum-phase due to flexibility, and noncolocated because of different sensor/actuator location. In spite of the above characteristics of a flexible manipulator, the end-effector position should accurately track the desired path for some practical applications. Therefore, our control objective becomes the tracking control of the nonminimum-phase, noncolocated, and nonlinear system with uncertainties. This may be the most general and complicated problem in the control of robotic manipulators.

To design a proper control scheme for a flexible manipulator, we have to decide what to overcome, what to avoid, or what to compromise. Most recently proposed ideas for the control of a flexible manipulator can be classified into following categories. Each method seems to have its advantages and penalties.

a) Use many sensors and actuators

If the number of actuators and sensors are equal to the number of controlled modes, the flexible structure can be controlled easily without a spillover problem [14]. Crawley [8] showed the possibility by using a distributed array of piezoelectric devices for actuation on a flexible structure. This may be the direct approach to the multiple mode system, but the penalty is the hard ware cost.

b) Change the structure of the manipulator

Alberts, et al [1] used a thin film of viscoelastic material between the structure surface and an elastic constraining layer to obtain passive damping. Park and Asada [15] tried to change the nonminimum-phase flexible manipulator system to a minimum-phase system by relocating the torque actuation point. This approach may be limited because the modification of the structure is not always possible.

c) Use advanced feedback control schemes

Hastings and Book [9] used optimal control with the strain feedback. Cannon and Schmitz [6] demonstrated the end-position control with the tip position and the joint rate feedback. Kotnik, et al [11] presented the results of the end-point acceleration feedback. Wang and Vidyasagar [19] applied the stable factorization approach to obtain the optimal step response. Singular perturbation methods were tried by Siciliano and Book [16], Khorrami and Ozguner [10], and

others. Various adaptive techniques have been proposed by Yuh [21], Yuan and Book [20], Cetinkunt and Wu [7], and other researchers. Bartolini and Ferrar [3] tried to extend the adaptive pole assignment control to nonminimum-phase system. Even though many control schemes have been proposed to control the flexible vibration actively, most control schemes will allow bounded tracking errors because the feedback control cannot cancel the nonminimum-phase zeros that have positive real values.

d) Allow noncausality

Bayo and Paden [3,4] developed the frequency-domain inverse dynamic method to calculate the feedforward torque profile for the end-point tracking. Kwon and Book [12] presented a time-domain inverse dynamic method, which generates desired flexible coordinates trajectories as well as the torque profile. These methods cancel the nonminimum-phase zeros, but the penalty is the loss of causality.

e) Modify the input to not excite the vibration

Seering and his co-workers [17,18] showed the residual vibration can be significantly reduced by shaping input signal with a series of impulses. The input shaping technique is effective for the simple mode system, but it is very sensitive to the damping and resonant frequency accuracy of the dynamic model.

Among the above control methods, since the inverse dynamic method can truly compensate the nonminimum-phase positive zeros, and can achieve the tracking performance, the asymptotic tracking controller is designed using the time-domain inverse dynamic method in this paper, and implemented on a single-link flexible manipulator for zero and non-zero initial condition cases. This paper shows that the inverse system can be divided into causal and

anticausal parts, and each system can be integrated separately with the initial conditions of the system states that are divided into rigid body, causal and anticausal parts. The rigid body component is used for the desired tip trajectory planning and the other two components are used separately for the calculation of the feedforward torque of causal and anticausal parts.

In the following sections, first, a single-link flexible manipulator is described and modeled using the assumed mode method. Second, the time-domain inverse dynamic equation is derived from the dynamic equation. Next, a tracking controller for the nonminimum-phase system is designed combining the inverse dynamic feedforward control and the joint feedback control. The tracking convergence characteristics are discussed in the Laplace domain, and the simulation and experimental results are presented for various initial condition (I.C.) cases including zero I.C. and non-zero I.C.

2. Modeling

A single link flexible manipulator having planar motions is described as shown in Figure

1. The rotating inertia of the servo motor, the tachometer, and the clamping hub are modeled

as a single hub inertia I_h . The payload is modeled as an end mass M_e and a rotational inertia J_e . Though structural damping exists in the flexible link, it is ignored in modeling.

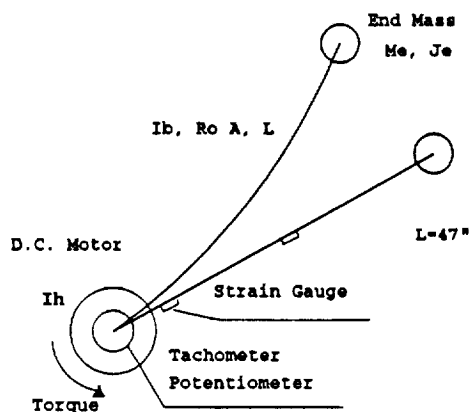


Figure 1. A single link flexible manipulator

Table 1. Physical properties of the flexible manipulator

Link	EI: stiffness (lb _f in ²)	4120	H: thickness (in)	3/16
	Ro A: unit length mass (lb _f s ² /in ²)	2.54E-4 * 0.14	L: length (in)	47
Tip mass	M _e : mass (lbm)	0.75	J _e :rot. inertia (lb _f s ² in)	3.8E-5
Hub	I _h : rot. inertia (lb _f s ² in)	0.14		

To derive equations of motion of the manipulator, we describe the position of a point on the beam with virtual rigid body motion and elastic deflection using a Bernoulli-Euler beam model. The rigid body mode coordinate that passes through the end point of the beam is selected [2,12], and the mode functions of pinned hub-inertia, pinned end-mass boundary conditions are used to describe the flexible displacement of the beam. Because of the selection of rigid body coordinate, the end point position of the beam can be expressed by the rigid body mode variable alone. With this simple representation of the end point position, we can derive the inverse dynamics equation easily.

By using Lagrange's equations of motion, the dynamic equations of a flexible manipulator are obtained with generalized coordinates. The detailed equations can be found in the reference [12].

$$[M] \ddot{q} + [D] \dot{q} + [K] q = [B] \tau$$

The dynamic equations can be divided into a rigid body motion part and a flexible motion part as follows:

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{rf}^T & M_{ff} \end{bmatrix} \begin{Bmatrix} \ddot{q}_r \\ \ddot{q}_f \end{Bmatrix} + \begin{bmatrix} D_{rr} & D_{rf} \\ D_{rf}^T & D_{ff} \end{bmatrix} \begin{Bmatrix} \dot{q}_r \\ \dot{q}_f \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ff} \end{bmatrix} \begin{Bmatrix} q_r \\ q_f \end{Bmatrix} = \begin{bmatrix} B_r \\ B_f \end{bmatrix} \tau \quad (1)$$

where $q_r = q_0$; rigid body coord.,

$$q_f = \begin{Bmatrix} q_1 \\ \vdots \\ q_n \end{Bmatrix} ; \text{flexible mode coord.}$$

For a state space form, we obtain the following dynamic equation.

$$\dot{X} = \begin{bmatrix} 0 & I \\ M^{-1}K & M^{-1}D \end{bmatrix} X + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} \tau$$

$$Y = [C]x + [F]\tau \quad (2)$$

$$\text{where } X = \{q_r, q_f, \dot{q}_r, \dot{q}_f\}^T$$

$$= \{q_0, q_1, \dots, q_n, \dot{q}_0, \dot{q}_1, \dots, \dot{q}_n\}^T$$

3. Inverse Dynamic Equations

From the above the direct dynamic equations, we will derive the inverse dynamic equation, which represents the relation between the desired acceleration of the rigid mode (equivalent to the tip acceleration) as input, and the torque as output. Eqn. (1) can be written in two parts.

$$[M_{rr}]\ddot{q}_r + [M_{rf}]\ddot{q}_f + [D_{rr}]\dot{q}_r + [D_{rf}]\dot{q}_f = [B_r]\tau \quad (3)$$

$$[M_{rf}]^T\ddot{q}_r + [M_{ff}]\ddot{q}_f + [D_{rf}]^T\dot{q}_r + [D_{ff}]\dot{q}_f + [K_{ff}]q_f = [B_f]\tau \quad (4)$$

From Eqn. (3), torque is expressed as

$$\tau = [B_r]^{-1} \{ [M_{rr}] \ddot{q}_r + [M_{rf}] \ddot{q}_f + [D_{rr}] \dot{q}_r + [D_{rf}] \dot{q}_f \} \quad (3a)$$

Substitution of Eqn. (3a) above into Eqn. (4) gives the following relations between the flexible coordinate q_f and the rigid body coordinate q_r .

$$[M_f] \ddot{q}_f + [D_f] \dot{q}_f + [K_f] q_f = [B_{i1}] \dot{q}_r + [B_{i2}] \ddot{q}_r$$

$$\text{where } [M_f] = \{ [M_{ff}] - [B_f][B_r]^{-1}[M_{rf}] \}$$

$$[D_f] = \{ [D_{ff}] - [B_f][B_r]^{-1}[D_{rf}] \} \quad (5)$$

$$[K_f] = [K_{ff}]$$

$$[B_{i1}] = \{ [B_f][B_r]^{-1}[D_{rr}] - [D_{rf}]^T \}$$

$$[B_{i2}] = \{ [B_f][B_r]^{-1}[M_{rr}] - [M_{rf}]^T \}$$

From Eqn. (4), the acceleration of the flexible coordinates is expressed as follows:

$$\begin{aligned} \ddot{q}_f = & -[M_{ff}]^{-1}[M_{rf}]^T \ddot{q}_r - [M_{ff}]^{-1}[D_{rf}]^T \dot{q}_r \\ & - [M_{ff}]^{-1}[D_{ff}] \dot{q}_f - [M_{ff}]^{-1}[K_{ff}] q_f + [M_{ff}]^{-1}[B_f] \tau \end{aligned} \quad (6)$$

Substitute Eqn. (6) into Eqn. (3), then we will get the following Eqn.

$$\tau = [C_{i1}] \dot{q}_f + [C_{i2}] \ddot{q}_f + [F_{i1}] \dot{q}_r + [F_{i2}] \ddot{q}_r \quad (7)$$

$$\begin{aligned}
\text{where } [G] &= \{[B_r] - [M_{rd}][M_{fd}]^{-1}[B_d]\}^{-1} \\
[C_{i1}] &= [G]\{-[M_{rd}][M_{fd}]^{-1}[K_{fd}]\} \\
[C_{i2}] &= [G]\{[D_{rd}] - [M_{rd}][M_{fd}]^{-1}[D_{fd}]\} \\
[F_{i1}] &= [G]\{[D_{rr}] - [M_{rd}][M_{fd}]^{-1}[D_{rd}]^T\} \\
[F_{i2}] &= [G]\{[M_{rr}] - [M_{rd}][M_{fd}]^{-1}[M_{rd}]^T\}
\end{aligned}$$

If Eqn. (5) and Eqn. (7) are represented in a state space form, the inverse dynamic equations will be written in this form.

$$\text{Let } X_i = \{q_f, \dot{q}_f\}^T, \quad q_{ir} = \{q_r, \dot{q}_r\}^T$$

$$\begin{aligned}
\dot{X}_i &= \begin{bmatrix} 0 & I \\ M_i^{-1}K_i & M_i^{-1}D_i \end{bmatrix} X_i + \begin{bmatrix} 0 & 0 \\ M_i^{-1}B_{i1} & M_i^{-1}B_{i2} \end{bmatrix} \dot{q}_{ir} \\
\tau &= [C_{i1}, C_{i2}]X_i + [F_{i1}, F_{i2}]\dot{q}_{ir}
\end{aligned} \tag{8}$$

$$\dot{X}_i = [A_i]X_i + [B_i]\dot{q}_{ir}$$

$$\tau = [C_i]X_i + [F_i]\dot{q}_{ir}$$

Even though the inverse dynamic equation is obtained in this simple form, it seems at first to be impossible to integrate that equation because the inverse system matrix A_i has positive real valued poles (which came from the positive zeros of the nonminimum-phase direct dynamic system) as well as negative poles. However, if we relax the solution range of that equation to include noncausal solutions, we can obtain a unique stable solution by integrating this 1st order differential equation.

The inverse dynamic system equation can be divided into the causal part and the

anticausal part by using a similarity transformation as follows.

[T]: Orthogonal transformation matrix

$$\begin{aligned} X_i &= [T]P_i \\ &= [T_c, T_{ac}]\{P_c, P_{ac}\}^T \end{aligned}$$

where $X_i = \{q_f, \dot{q}_f\}^T$, T_c 's basis includes the eigenvectors which have negative eigenvalues and T_{ac} is made of the eigenvectors of positive eigenvalues.

$$\begin{aligned} [T]^{-1}A_i[T] &= \begin{bmatrix} A_{ic} & 0 \\ 0 & A_{iac} \end{bmatrix} \\ \begin{Bmatrix} \dot{P}_c \\ \dot{P}_{ac} \end{Bmatrix} &= \begin{bmatrix} A_{ic} & 0 \\ 0 & A_{iac} \end{bmatrix} \begin{Bmatrix} P_c \\ P_{ac} \end{Bmatrix} + \begin{bmatrix} B_{ic} \\ B_{iac} \end{bmatrix} \dot{q}_{ir} \\ \begin{Bmatrix} \tau_c \\ \tau_{ac} \end{Bmatrix} &= \begin{bmatrix} C_{ic} \\ C_{iac} \end{bmatrix} \begin{Bmatrix} P_c \\ P_{ac} \end{Bmatrix} + \begin{bmatrix} 1/2 F_i \\ 1/2 F_i \end{bmatrix} \dot{q}_{ir} \\ \tau &= \tau_c + \tau_{ac} \end{aligned} \quad (9)$$

Such a coordinate change decouples the inverse system into two subsystems of Eqn. (9). The new variable P_c represents the coordinates of the causal system, and the P_{ac} represents the anticausal system.

For a given end-point trajectory of Figure 2, the causal part torque is obtained by integrating the causal part inverse dynamic equations forward from the initial time of the trajectory. Then, the anticausal system equations must be integrated backward from the final time of the trajectory. The total torque, which is the output of these equations, is obtained by adding the both torques as shown in Figure 3. As additional outputs, the flexible coordinate trajectories, which give the joint angle and strain trajectories, were obtained from the integration

of the inverse dynamic equation.

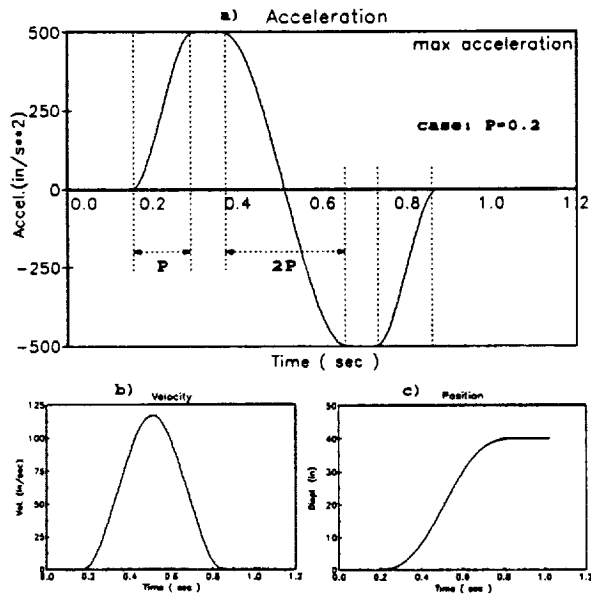


Figure 2. Desired end point trajectory
a)Acceleration, b)Velocity, c)Position

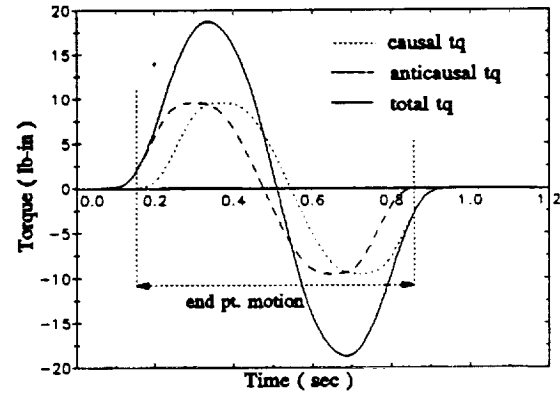


Figure 3. Calculation of torque with the inverse dynamic method

4. Analysis of the System State Space

It is very useful to analyze the relations between the dynamic system states, and causal and anticausal subsystems' states to construct additional desired trajectories of flexible coordinates. We can obtain the reference trajectory of the flexible mode coordinates of the direct dynamic system from a given rigid mode trajectory and the calculated state trajectories of the causal and anticausal systems.

As we can expect in Eqn. (2), (8), and (9), the space of the full state vector X of the direct dynamic system can be divided into three subspaces: the rigid body coordinate subspace q_{ir} , the causal part flexible coordinate subspace P_c , and the anticausal part flexible coordinate subspace P_{ac} . These subspaces are linearly independent and orthogonal to one another. The relations of these spaces are illustrated in Figure 4, and described by the following Eqn. (10)

when only two flexible modes are considered.

$$\text{where } X = \{q_r, q_{f1}, q_{f2}, \dot{q}_r, \dot{q}_{f1}, \dot{q}_{f2}\}^T$$

$$q_{ir} = \{q_r, \dot{q}_r\}^T, \quad X_i = \{q_{f1}, q_{f2}, \dot{q}_{f1}, \dot{q}_{f2}\}^T$$

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} q_{ir} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_i \quad (10)$$

$$X = H_r q_{ir} + H_f(TP_i)$$

$$X = H_r q_{ir} + H_f T_c P_c + H_f T_{ac} P_{ac}$$

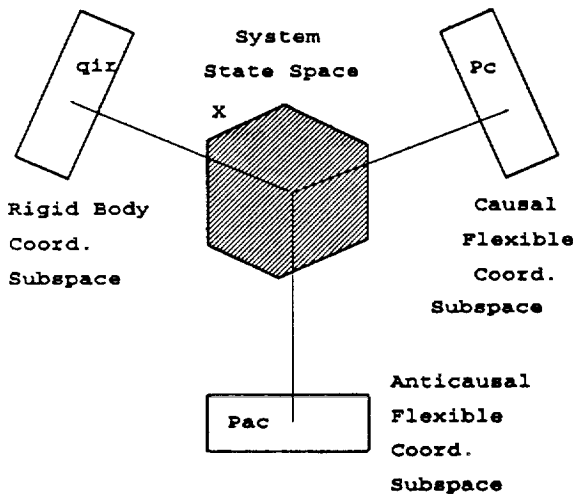


Figure 4. Dimensional analysis of state variables of flexible manipulator dynamic equations

From the given end point trajectory, the rigid body coordinate trajectory q_{ir} is obtained. The flexible coordinate trajectories of P_c and P_{ac} are then calculated from the integration of Eqn. (9). Thus, the trajectories of the complete state vector X can be obtained by using Eqn. (10). These trajectory values can be used as reference commands for feedback tracking control.

The generation of complete state trajectories is an advantage of this inverse dynamic method. We get the strain trajectories as well as the joint trajectory. We no longer have to give the flexible manipulator reference commands to follow the trajectory like a rigid manipulator.

5. Tracking Controller Design

Let's consider a linear nonminimum-phase model (the output is the tip position), and a linear minimum-phase model (the output is the joint angle) of a flexible manipulator in the form of

$$\begin{array}{ll}
 \text{. nonminimum-phase} & \text{. minimum-phase} \\
 X_e(s) = \frac{Z_e(s)}{P(s)} \tau(s), & \theta(s) = \frac{Z_\theta(s)}{P(s)} \tau(s)
 \end{array} \quad (11)$$

where X_e : Tip position
 θ : Joint angle
 τ : Joint torque input

$$P(s) = s^2 \prod_{i=1}^n (s \pm j p_i)$$

$$Z_e(s) = \prod_{i=1}^n (s \pm z_{ei})$$

$$Z_\theta(s) = \prod_{i=1}^n (s \pm j z_{\theta i})$$

The nonminimum-phase system has imaginary poles p_i and real positive and negative zeros z_i compared to the minimum-phase system, which has the same imaginary poles and imaginary zeros $z_{\theta i}$.

The control objective is to make the output $X_e(t)$ (the tip position) follow the desired time-varying trajectory $X_{ed}(t)$ applying the joint torque $\tau(t)$. In Figure 5, the input torque τ will consist of the inverse dynamic feedforward torque τ_d , the feedback torque τ_b , driven by tracking errors, and the disturbance or friction torque w .

$$\tau = \tau_d + \tau_b + w \quad (12)$$

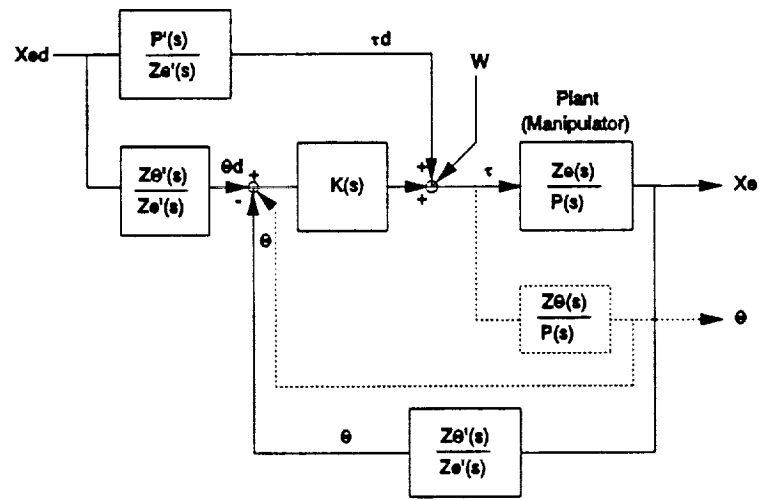


Figure 5. Tracking control of a nonminimum-phase flexible manipulator

The feedforward torque τ_d is obtained using the inverse dynamic method that is equivalent to the two-sided Laplace transform with the region of convergence along the imaginary axis to get the stable solution in the time-domain. The feedback torque τ_b is generated by the error signals between the desired joint and strain trajectories, and the measured joint angle and strains. In experiment, the joint angle is measured directly without being estimated from the end-point X_e measurement. Therefore, in Figure 5, the actual feedback loop is the dot line (...) rather than the solid line (—). Using the colocated feedback control with the joint trajectory, we could avoid the noncolocated control problem of the direct the tip position feedback. This advantage is obtained from generation of the desired flexible coordinate trajectories, which gives the joint trajectories and the strain trajectories that match the tip trajectory dynamically.

$$\begin{aligned}
\tau_d &= \frac{P'(s)}{Z'_e(s)} X_{ed}(s) \\
\tau_b &= K(s) (\theta_d(s) - \theta(s)) \\
\text{where } \theta_d(s) &= \frac{Z'_\theta(s)}{Z'_e(s)}, \\
\theta(s) &= \frac{Z_\theta(s)}{P(s)} \tau(s) \quad \text{or} \quad \frac{Z'_\theta(s)}{Z'_e(s)} X_e(s)
\end{aligned} \tag{13}$$

The ' notation of Eqn. (13) means the estimated model of the real system.

From Eqn. (11), (12), (13), the output $X_e(s)$ will be in the form of

$$X_e = \frac{Z_e}{P + K Z_\theta} W + \frac{Z_e (P' + K Z'_\theta)}{Z'_e (P + K Z_\theta)} X_{ed} \tag{14}$$

If the feedback gain of $K(s)$ becomes very large, the disturbance effect will be negligible, so that the tracking performance will be depend on the accuracy of the zeros of the model as follows.

$$X_e = \frac{Z_e Z'_\theta}{Z'_e Z_\theta} X_{ed} \tag{15}$$

From Eqn. (14), the tip position tracking error dynamics will be as follows:

$$\left(1 + \frac{K Z_\theta}{P}\right) E = \frac{Z_e}{P} W + \left\{ \left(\frac{P' Z_e}{P Z'_e} - 1 \right) + \frac{K}{P} \left(Z_e \frac{Z'_\theta}{Z'_e} - Z_\theta \right) \right\} X_{ed} \tag{16}$$

$$\text{where } E = X_{ed} - X_e$$

If the disturbance effect is negligible, and the dynamic model is good enough to cancel the system dynamics, the error dynamics will converge to zero as shown in Eqn. (17).

$$\begin{aligned}
 \left(1 + \frac{KZ_0}{P}\right) E = 0 & \quad \text{If } W = 0 \\
 P' = P & \\
 Z_e' = Z_e & \\
 Z_0' = Z_0 &
 \end{aligned}
 \tag{17}$$

Then, the tracking dynamics are determined entirely by the joint feedback loop characteristics, which is a stable collocated control loop. Therefore, the control scheme will guarantee perfect tracking if the initial tracking errors are zero, and asymptotic tracking convergence for non-zero initial tracking errors, as long as the model is accurate. Even if the model is not exact, the tracking error will be bounded because the positive poles of the inverse system give a stable torque profile using the two-sided Laplace transform.

6. Perfect Tracking with Zero Initial Conditions

The above combined control scheme of the inverse dynamic feedforward control and the feedback control was implemented for a rest-to-rest motion of the experimental manipulator. For a real time control, a Micro Vax II was used with 12 bit A/D and D/A boards. The off line calculation of a trajectory and a torque profile was also performed in the Micro Vax.

By applying the precalculated torque, compensating the joint friction, and using the feedback of the tracking error at the joint, the excellent result of Figure 5.5 was obtained. The flexible manipulator could stop without any overshoot or any residual vibration after it moved 40 inches (48.76 degrees) within less than 0.8 second. In the strain signal, there exists a rough jerk that could be eliminated by using a smoother acceleration profile. Unfortunately, since an end-point position sensor was not available for the system, the end point position couldn't be

measured directly. However, the end point tracking performance can be estimated from the joint tracking result and the strain tracking result. If the joint doesn't have any overshoot or vibration and the strain doesn't show any residual vibration, the end point can be presumed to stop without any overshoot or vibration.

The experimental results show that a simple joint feedback PD controller provides excellent tracking if it is combined with the inverse dynamic feedforward control, and if the joint trajectory are generated from the desired end-point trajectory considered flexible dynamics. Full state feedback has been shown to converge even faster, but the increased difficulty in implementation seems unwarranted.

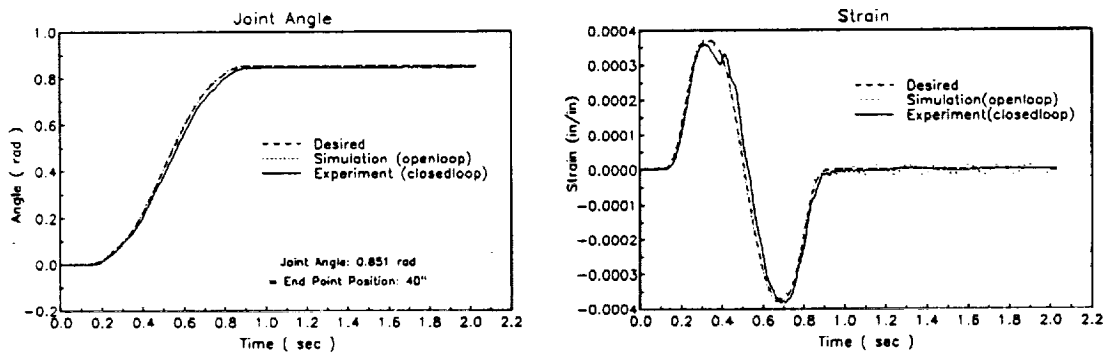


Figure 6. Experimental results of the combined control of the inverse dynamic feedforward control and the joint tracking feedforward control: a) Joint angle, b) Strain at the base

7. Tracking Control for Non-zero Initial Condition Cases

The previous section showed good tracking results for rest-to-rest motion, which gives zero initial and final flexible coordinate values. How, then, can this method can be applied to non-zero initial condition (moving) cases? In the control scenario of Book's bracing manipulator

[13], the large flexible manipulator will be commanded to move in free space using a simple feedback controller with rough information of the target surface position until it senses the target bracing surface. If it measures more accurately the distance from the tip position to the bracing surface during motion, the controller will modify or create the desired tip trajectory accommodating the moving non-zero initial conditions, and calculate the required torque and the desired trajectories of all states with respect to the new tip trajectory. Then, it will follow the trajectory and will stop just in front of the surface or approach the surface with a suitable slow speed, and will contact to the surface without a large impact. Therefore, we need to be able to do trajectory planning and tracking control for general moving initial conditions.

7.1 Analysis of Initial Conditions of the States and Trajectory Planning

As explained in Figure 4 of section 4, the system states X of a flexible manipulator's dynamics can be divided into a rigid body component q_{ir} , a causal component P_c , and an anticausal component P_{ac} . Naturally, the initial system states X_0 can be divided into q_{ir0} , P_{c0} , and P_{ac0} .

$$X_0 = H_r q_{ir0} + H_f T_c P_{c0} + H_f T_{ac} P_{ac0}$$

First of all, let's assume the initial condition (I.C.) X_0 and the desired final state X_f is known or measured exactly. The final condition X_f also will be divided into three components q_{irf} , P_{cf} , and P_{acf} . From the initial and final values of the rigid body coordinate, we are generating the desired end point trajectories including the desired acceleration, velocity, and position profiles considering the minimum traveling time and the high frequency content. The desired trajectories of Figure 7 are generated with the following example data: the end point initial acceleration $X_{ca0} = -10 \text{ in/s}^2$, the initial velocity $X_{cv0} = 60 \text{ in/s}$, the initial position X_{c0}

= 16.4 in, and the desired final position $X_{ef} = 57.4$ in, $X_{ev0} = 0$, $X_{ca0} = 0$. The final condition doesn't have to be stationary: it can be an arbitrary moving condition. Now we have planned the desired tip trajectories of a flexible manipulator. If the tracking controller is working properly, the end point will follow the trajectory and will stop without any overshoot or vibrations.

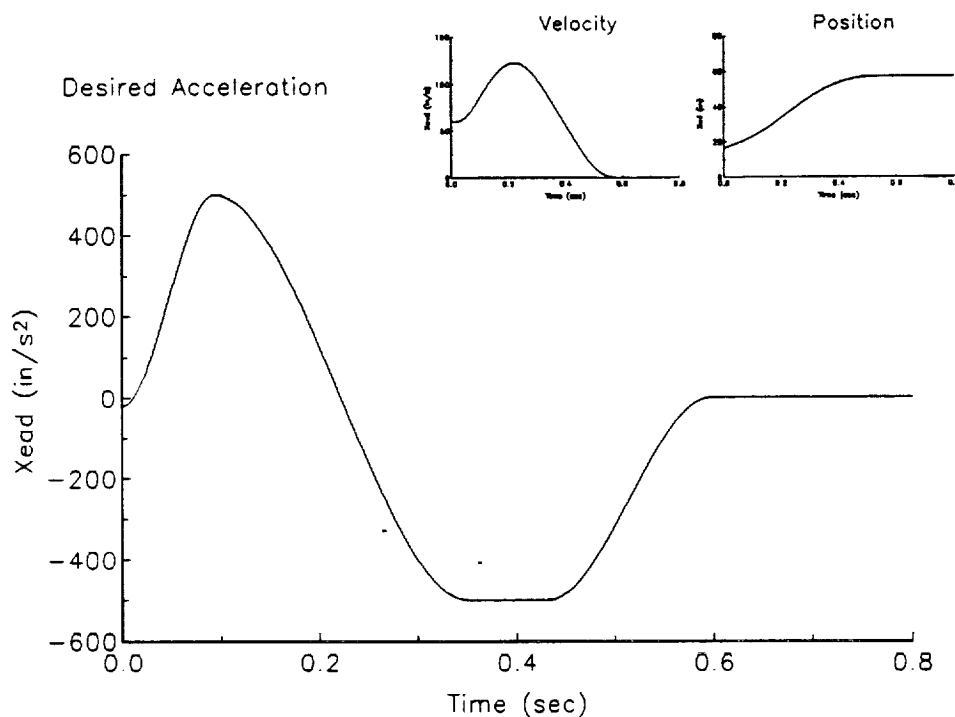


Figure 7. The desired end-point trajectories for non-zero I.C.

7.2. Inverse Dynamic Torque Calculation

Since the initial and final conditions, and the rigid body coordinate trajectory (equivalent to the end-point trajectory) are given, the torque profile can be calculated using the same inverse dynamic method for the zero I.C. case. The causal torque is calculated by integrating the causal part equation with the initial condition P_{c0} , and the anticausal torque is calculated by integrating

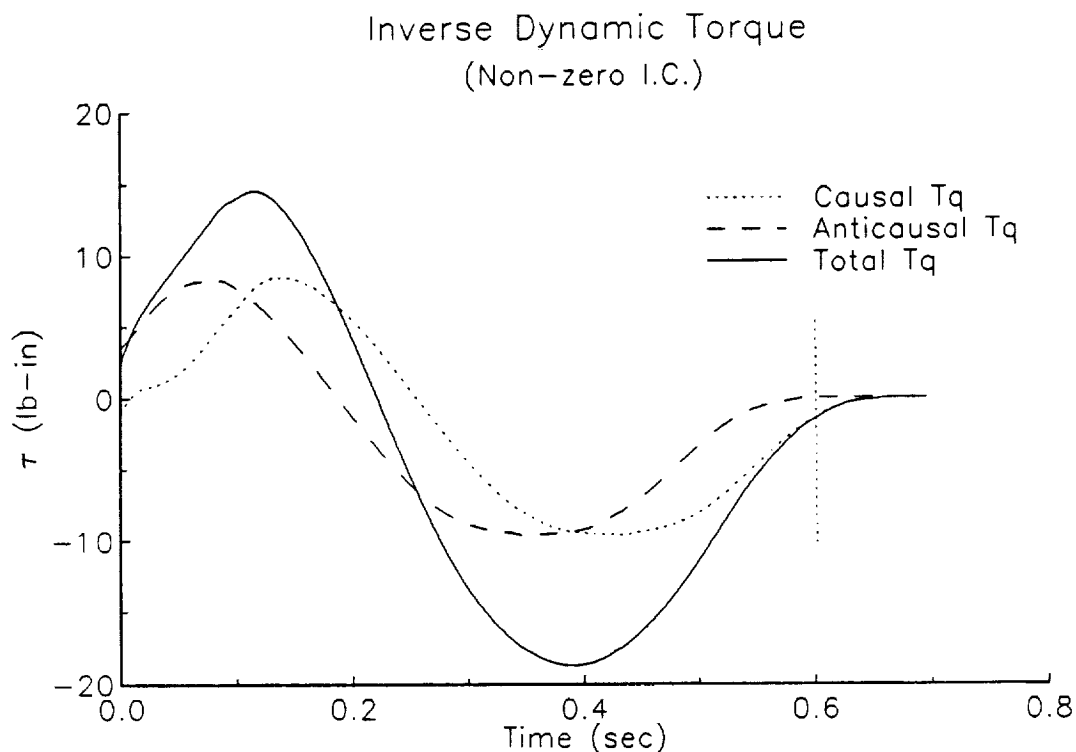


Figure 95.7 Inverse dynamic torque for non-zero I.C.

the anticausal part equation backwards with the final condition P_{acf} as shown in Figure 8. That means the anticausal part of the I.C. P_{aco} cannot be considered in the inverse dynamic torque calculation. This is a limitation of the inverse dynamic method for non-zero initial condition case with the current fixed acceleration profile of the end-point. If the acceleration has more degree of freedom with extra parameters, it can satisfy the anticausal part initial condition by manipulating the auxiliary parameters. However, this approach of changing the acceleration profile structure is not so practical if implemented. In this research, we maintain the acceleration profile of Figure 2 adjusting the acceleration and deceleration time ratio without trying to match up to the anticausal part I.C. Even though the initial conditions of anticausal

part flexible coordinates cannot be considered in feedforward inverse dynamic control, the tracking error can be made to converge to zero by the asymptotic tracking capability of the feedback loop. The following sections examine several mismatched cases of initial tracking errors.

7.3 Simulation Studies of Several Initial Tracking Errors

Case 1: Perfect knowledge of I.C. (no initial tracking errors)

In order to get zero initial tracking errors, we have to know and use the exact initial values of the rigid coordinate and causal part flexible coordinate for trajectory planning and torque calculation, and the initial values of the anticausal part flexible coordinates naturally must be the same as the calculated anticausal flexible coordinates at $t=0$, which were obtained from the backward integration. Then, perfect tracking can be expected as shown in Figure 5.8. The simulation results perfectly match the desired trajectories of the end-point, the joint angle, and strains.

Case 2: Initial errors in flexible coordinates

The assumption of the case 1 is that the initial values of the anticausal flexible coordinates naturally meet with the calculated values of the backward integration. If we relax this unrealistic assumption or allow the measurements of the strain to have errors, we will always have initial tracking errors of flexible coordinates. As the anticausal part torque near $t=0$ cannot cancel the dynamics of the system, but it is good enough to cancel the dynamics at $t=t_f$, Figure 5.9 shows the asymptotically converging results. Though the strain signals have

oscillating tracking errors initially, they converge to the desired trajectories, and produce no overshoot or residual vibrations at the final position.

Case 3: Initial errors in rigid body coordinates

In practical situations, it can be assumed generally that the measurement of the tip acceleration, velocity and position may have noise or errors. In this case, the desired end point trajectory is generated using the mismatched initial conditions. The rigid body coordinate trajectory (equivalent to the end-point trajectory) based on the mismatched I.C., will affect the causal part and the anticausal part torque calculation. Therefore, the feedforward torque cannot achieve the desired tracking characteristics, but the feedback control will compensate the tracking errors asymptotically. As shown in Figure 11, the tip trajectory converges slowly to the desired trajectory. This example was simulated with a wrong estimation of the initial tip velocity 60 in/s for the actual initial tip velocity 100 in/s. Intentionally, large initial errors were used for clear illustration of the tracking convergence.

With this case, we can expect the estimation of the initial value of the rigid body coordinate to be very significant for the inverse dynamic method, because it is used for trajectory planning and it affects the torque calculation directly.

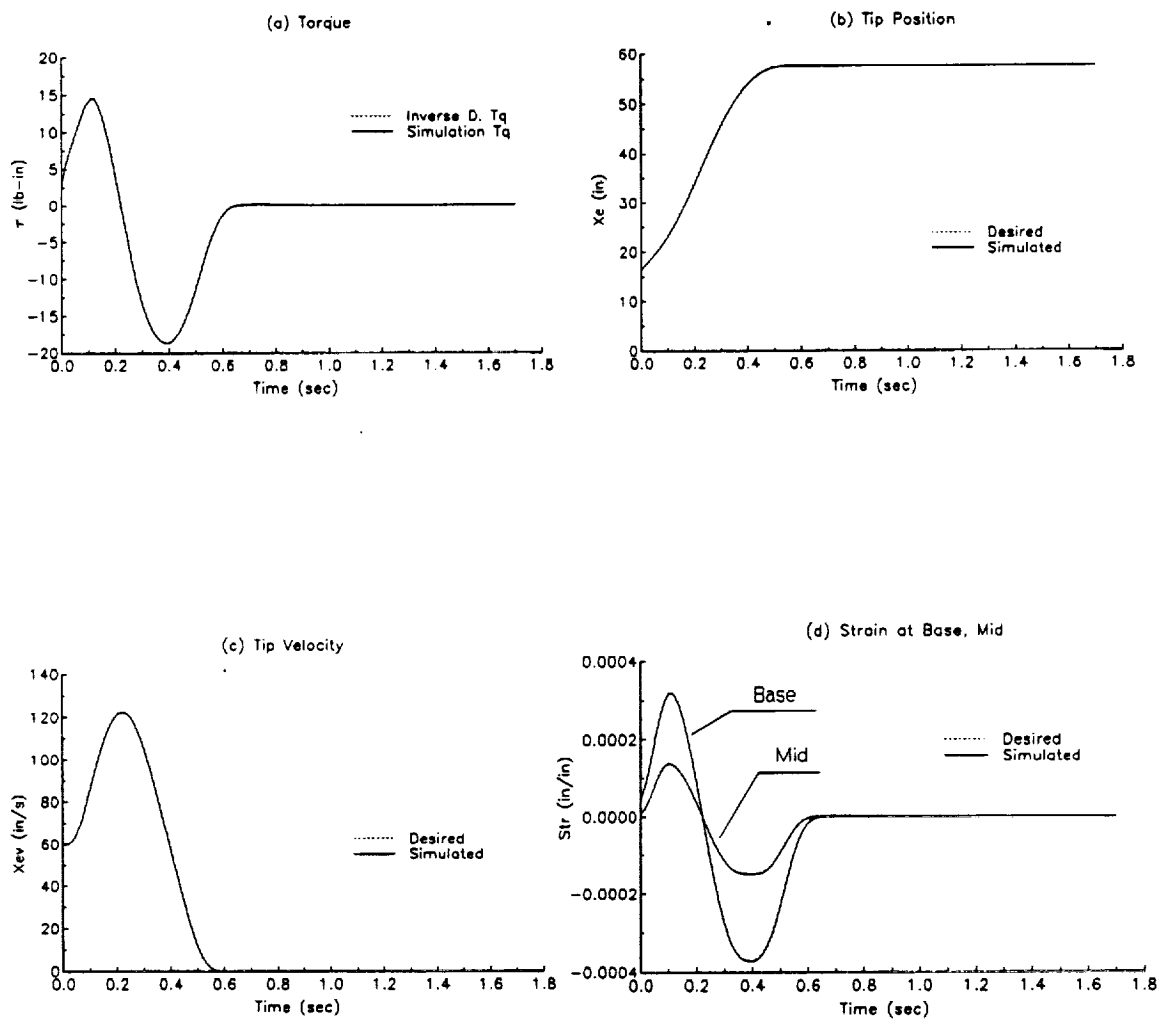


Figure 9. Case 1: Perfect tracking with zero initial tracking errors

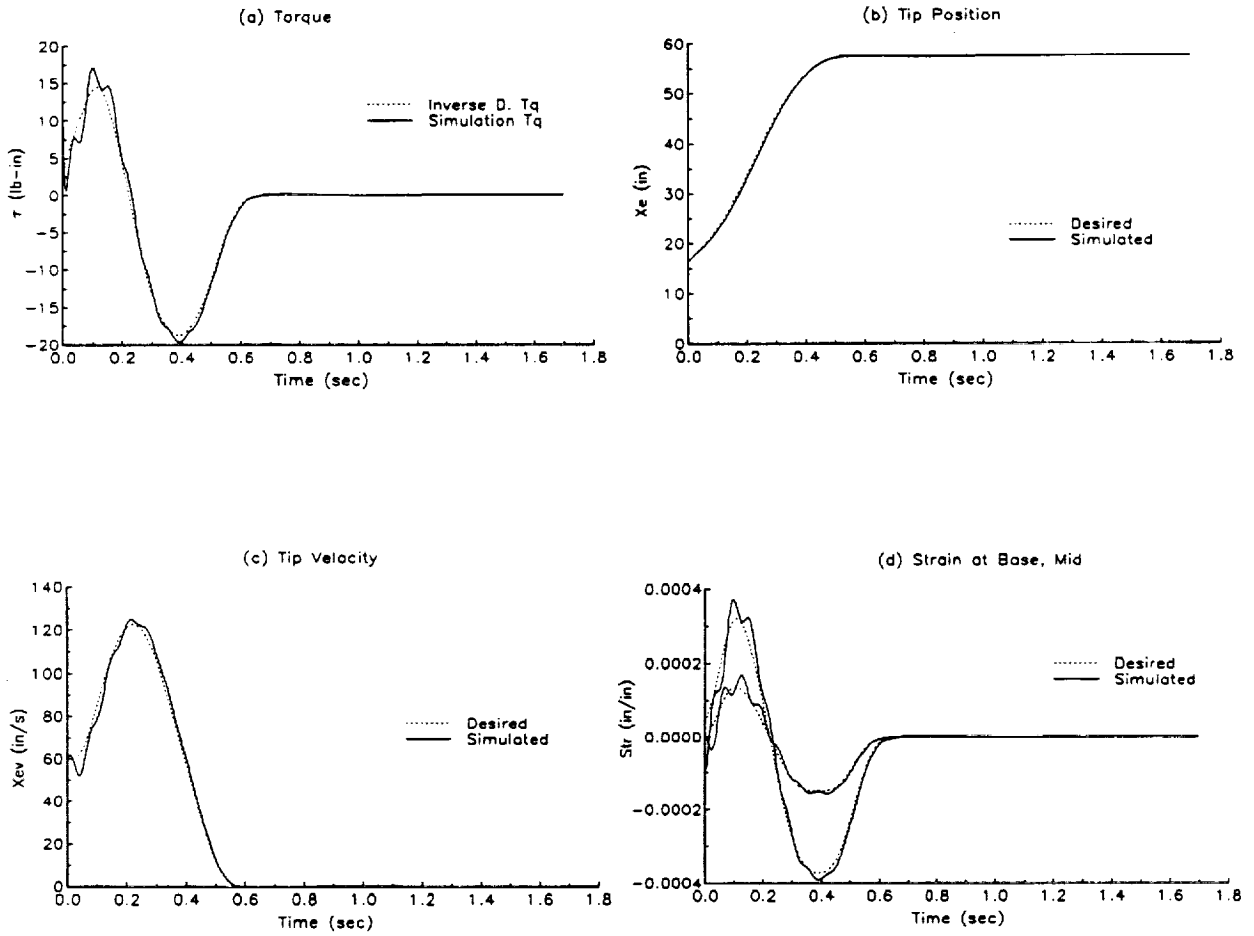


Figure 10. Case 2: Asymptotic tracking with initial errors of flexible coordinates

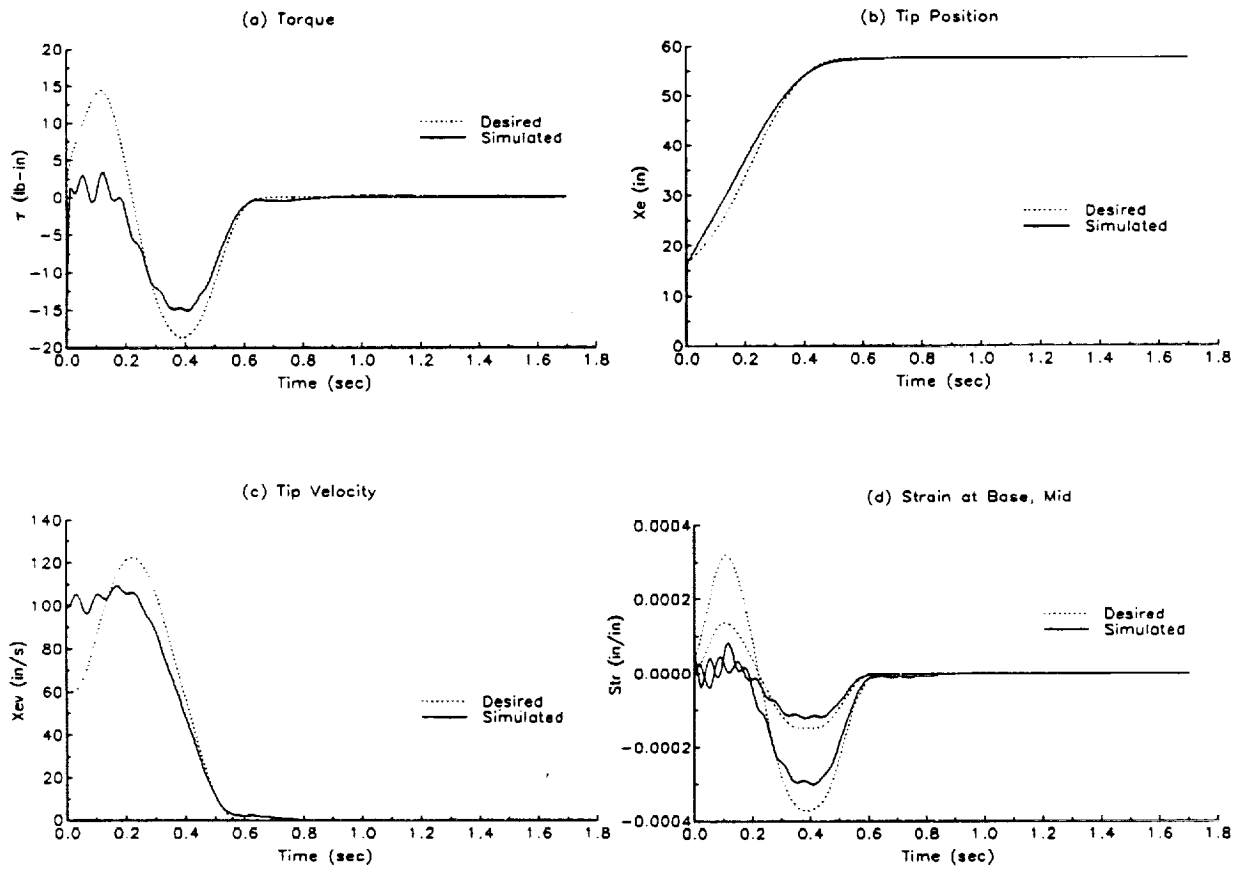


Figure 11. Case 3: Asymptotic tracking with initial errors of rigid body coordinates

7.4 Practical trajectory planning method

The previous simulation analysis is based on the assumption that any calculation of the trajectory planning and the torque and desired trajectory calculation can be performed within next sampling time just after measuring the initial conditions. In order to use this control method for practical application, we had better consider the fact that the calculation time is much longer than one sampling period of the controller. Moreover, the sudden application of the torque is not desirable because it contains high frequency enough to excite the unmodeled high system natural frequency.

As a practical method, when the actual moving initial conditions are measured at a certain instance, we estimate the states after noncausal time $t=t_0^*$ of Figure 12, and calculate the end point trajectory from the position at $t=t_0^*$ to the final position at $t=t_f$. With this end point trajectory, the inverse dynamic torque and the desired trajectory has been calculated. To avoid the sudden step torque, the noncausal part torque between $t=t_0$ and $t=t_0^*$ is applied as the

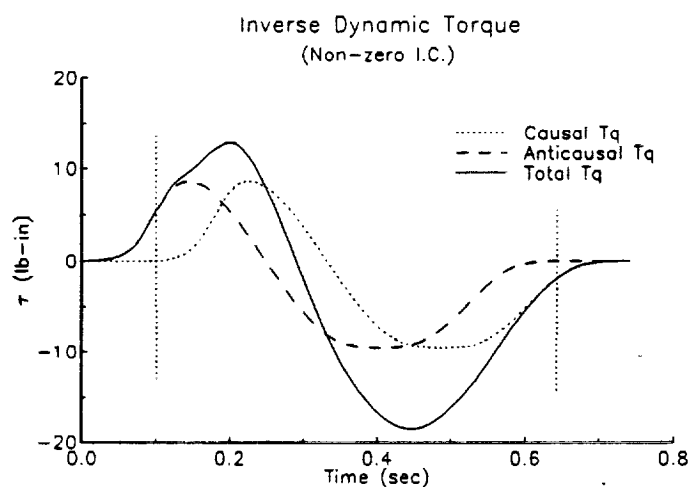


Figure 12. Inverse dynamic torque for non-zero I.C. (Practical approach)

inverse dynamic torque, assuming the initial flexible coordinate values at $t=t_0$ are all zeros.

Case 4: Initial errors in general cases (Practical approach)

Since the initial conditions, which were used for trajectory planning and the inverse dynamic calculation, are estimated from the measured initial conditions before $t=t_0^*$, and the flexible coordinate values are assumed as zero, it is natural to have initial tracking errors for all system states and an imperfect inverse dynamic torque profile.

However, Figure 13 clearly shows the asymptotically converging trend of tracking errors, and no overshoot or vibration after $t=t_f^*$.

Case 5: Experimental results of general case

The control scheme of the Case 4 has been implemented to the real experimental flexible manipulator using same conditions. As the end position sensor is not available, the end position behavior can be predicted from the joint angle and the strain measurements. Though the tracking performance in Figure 14 is not as good as the simulation result, the experimental results, which show no overshoot or residual vibrations, are good enough for contact or bracing control. Non-perfect tracking is presumed due to the joint friction because the coulomb friction force deteriorates the inverse dynamic feedforward torque profile especially at the low speed. Despite the initial tracking errors and the friction, the end point stops at the desired position without any overshoot. It indicates the nonminimum-phase system zeros are canceled by the inverse dynamic torque almost perfectly. Thus, this control method for moving initial conditions can be useful to avoid the large impact for contact or bracing.

8. Discussions

This paper presented a time-domain inverse dynamic method that calculate the torque

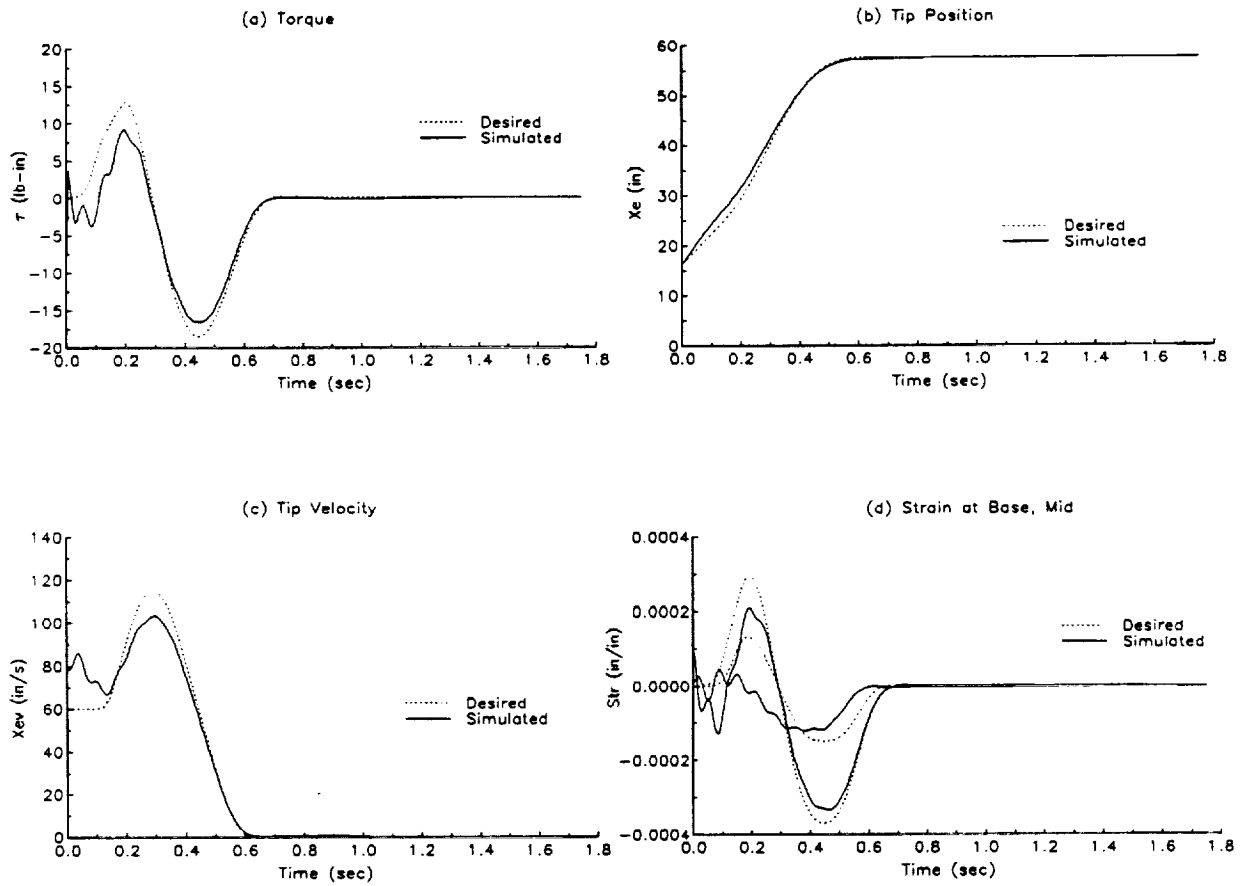


Figure 13. Case 4: Asymptotic tracking for general initial tracking errors

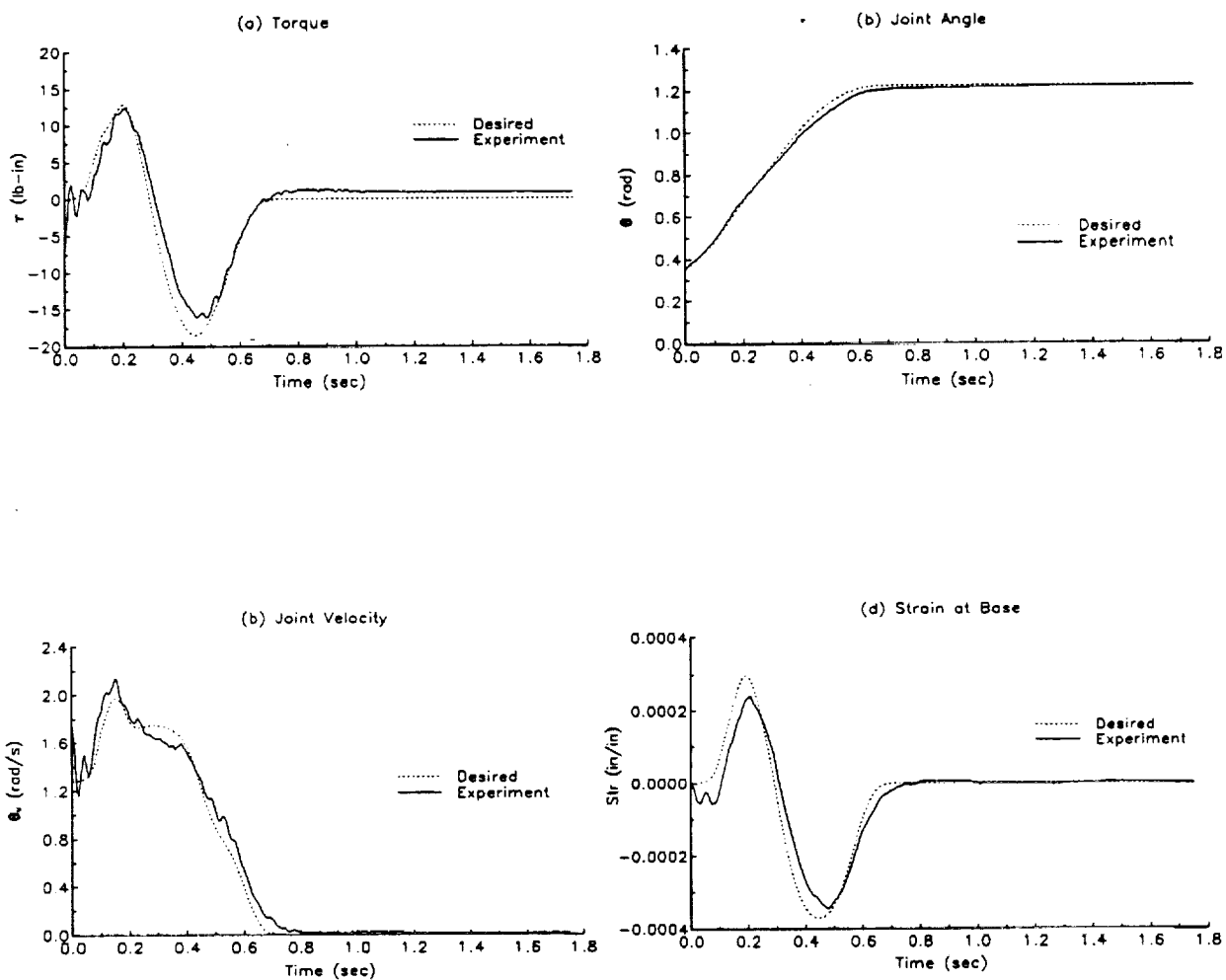


Figure 14. Case 5: Experiment on the tracking control for non-zero I.C.

profile and the desired flexible coordinates. It also gives a new interpretation of the inverse system of flexible manipulators. The inverse system has been divided into causal, and anticausal systems. And, the dynamic system state space can be divided rigid, causal, anticausal system subspaces.

The inverse dynamic torque can be interpreted in two ways, first, as a feedforward torque cancels the nonminimum phase system poles and zeros. As a result, the total system transfer function becomes a unit transfer function that makes the output exactly equal to the input. Second, we can interpret the inverse dynamic method as a input shaping technique. It functions like a noncausal notch filter that has zeros at the system resonant frequency. Therefore, the generated input torque profile doesn't have the system resonant frequency content of the input.

Many feedback control schemes require full state feedback, and assigning zero values to the desired flexible mode trajectories has been acceptable for the tracking control of flexible manipulators. Since the time-domain inverse dynamic method provides flexible mode trajectories that match the desired tip trajectory dynamically, it can be used for a trajectory generation method for many advanced feedback control schemes.

If we can measure the I.C. and accommodate the I.C. to the desired trajectories generation, we will obtain perfect tracking with no initial tracking errors eliminating the transient period like the zero I.C. case. However, it is very difficult to accommodate the anticausal part I.C. to the desired trajectory generation because of the backward integration of the anticausal system. This paper's trajectory generation method with the fixed structure acceleration profile will generally have initial tracking errors because of the anticausal part. However, the tracking controller shows the fast tracking convergence in the simulation and experiments. At the final

stop, no overshoot nor residual vibration was observed, so that this tracking method will be useful to avoid the large impact for contact or bracing.

Acknowledgement

This research has been partially supported by the National Aeronautics and Space Administration under the Grant No. NAG 1-623.

References

- [1] Alberts, T.E., Book, W.J., and Dickerson, S.L., "Experiments in Augmenting Active Control of a Flexible Structure with Passive Damping," AIAA Paper 86-0176, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 6-9, 1986.
- [2] Asada, H., and Ma, Z., "Inverse Dynamics of Flexible Robots," Proceedings of American Control Conference, 1989, pp.2352-2359.
- [3] Bartolini, G., and Ferrara, A., "Extension to Nonminimum Phase Systems of Adaptive Pole Assignment Control by Means of Robustness techniques," Proceedings of the 29th IEEE Conference on Decision and Control, December, 1990, pp.994-995.
- [4] Bayo, E., "A Finite Element Approach to Control the End-Point Motion of a Single-Link Flexible Robot," *Journal of Robotic Systems*, Vol.4, No.1, 1987, pp.63-75.
- [5] Bayo, E., and Paden, B., "On Trajectory Generation for Flexible Robots," *Journal of Robotic Systems*, Vol.4, No.2, 1987, pp.229-235.
- [6] Cannon, R.H., and Schmitz, E., "Initial Experiments on the End-point Control of a Flexible One-Link Robot," *The International Journal of Robotic Research*, Vol.3, No.3, Fall, 1984, pp.62-75.

- [7] Cetinkunt, S., and Wu, S., "Tip Position Control of a Flexible One Arm Robot with Predictive Adaptive Output Feedback Implemented with Lattice Filter Parameter Identifier," Proc. of 1990 IEEE Conference on Robotics and Automation, pp.1620-1625.
- [8] Crawley, E. F., and de Luis, J., "Experimental Verification of Distributed Piezoelectric Actuators for Use in Precision Space Structure," Structures, Structural Dynamics and Materials Conference, San Antonio, Texas, May, 1986.
- [9] Hastings, G., and Book, W.J., "Experiments in the Optimal Control of a Flexible Manipulator," Proceedings of American Control Conference, Boston, 1985, pp.728-729.
- [10] Khorrani, F., and Ozguner, U., "Perturbation Methods in Control of Flexible Link Manipulators," Proceedings of 1988 IEEE Conference on Robotics and Automation.
- [11] Kotnik, P.T., Yurkovich, S., and Ozguner U., "Acceleration feedback for Control of a Flexible Manipulator Arm," *Journal of Robotic Systems*, 5(3), 1988, pp.181-196.
- [12] Kwon, D.-S., and Book, W.J., "An Inverse Dynamic Method Yielding Flexible Manipulator State Trajectories," Proceedings of American Control Conference, San Diego, June, 1990, pp.186-193.
- [13] Kwon, D.-S., and Book, W.J., "A Framework for Analysis of a Bracing Manipulator with Staged Positioning," *Symposium on Robotics*, Proceedings of ASME Winter Annual Meeting, Chicago, IL, November, 1988, pp.27-36.
- [14] Meirovitch, L., Baruh, H., and Oz,H., "A Comparison of Control Techniques for Large Flexible Systems," *AIAA Journal of Guidance*, Vol.6, No.4, July-August, 1983, pp.302-310.
- [15] Park, J.H., and Asada, H., "Design and Control of Minimum-Phase Flexible Arms with

- Torque Transmission Mechanisms," Proceedings of 1990 IEEE Conference on Robotics and Automation, Cincinnati, Ohio, May, 1990, pp.1790-1795.
- [16] Siciliano, B., and Book, W.J., "A Singular Perturbation Approach to Control of Lightweight Flexible Manipulators," *The International Journal of Robotics Research*, Vol.7, No.4, August, 1988, pp.79-90.
- [17] Singer, N.C., and Seering, W.P., "Design and Comparison of Command Shaping Methods for Controlling Residual Vibration," Proceedings of IEEE Conference on Robotics and Automation, Scottsdale, AZ, May, 1989.
- [18] Singhose, W., Seering, W.P., and Singer, N.C., "Shaping Inputs to Reduce Vibration: A Vector Diagram Approach," Proceedings of IEEE Conference on Robotics and Automation, 1990, pp.922-927.
- [19] Wang, D., and Vidyasagar, M., "Modelling and Control of Flexible Beam Using the Stable Factorization Approach," *Robotics: Theory and Applications*, Proceedings of ASME Winter Annual Meeting, December, 1986, pp.31-37.
- [20] Yuan, B.-S., Book, W.J., and Siciliano, B., "Direct Adaptive Control of a One-Link Flexible Arm with tracking," *Journal of Robotic Systems*, 6(6), 1989, pp.663-680.
- [21] Yuh, J., "Application of Discrete-Time Model Reference Adaptive Control to a Flexible Single-Link Robot," *Journal of Robotic Systems*, 4(5), 1987, pp.621-630.

EXPERIMENTAL INVESTIGATIONS OF THE EFFECTS OF CUTTING ANGLE
ON CHATTERING OF A FLEXIBLE MANIPULATOR

J. Lew, J. Huggins, D. Magee and W. Book

Flexible Automation Laboratory
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

When a machine tool is mounted at the tip of a robotic manipulator, the manipulator becomes more flexible (the natural frequencies are lowered). Moreover, for a given flexible manipulator, its compliance will be different depending on feedback gains, configurations, and direction of interest. In this paper, the compliance of a manipulator is derived analytically, and its magnitude is represented as a compliance ellipsoid. Then, using a two link flexible manipulator with an abrasive cut-off saw, the experimental investigation shows that the chattering varies with the saw cutting angle due to the different compliance. The main work is devoted to finding a desirable cutting angle which reduces the chattering.

1. INTRODUCTION

1.1 Motivation

In real world applications, robot manipulators are mechanically very rigid by design. This rigidity is necessary for high positioning accuracy; however, it becomes difficult to perform operations when a rigid manipulator contacts a workpiece. On the other hand, flexible manipulators can provide passive compliance due to their link flexibility. With this structural compliance, certain applications such as cutting a workpiece can be performed with pure position control. Thus, the compliance can provide a simple, inexpensive solution for certain applications that otherwise could not be achieved with position control alone [1,2].

Figure 1 shows the block diagrams of the overall architecture of a cutting process with pure position control. An abrasive cut-off saw is mounted at the tip of a manipulator. Its link flexibility is represented by a spring constant (K_L). The position feedback signal is measured at each joint. Due to the flexibility of the link, it is possible to regulate the force applied to the workpiece by controlling the position of the end-effector relative to the workpiece. However, if the stiffness of the link is high, any uncertainty in the position of the workpiece, or er-

rors in the position servo of the manipulator will induce very large cutting forces. Eventually these uncontrolled large cutting forces will shorten the life of the grinding wheel and the manipulator. Also, the high stiffness (K_L) causes a high frequency oscillation or unstable chattering due to reaction forces from the workpiece. This behavior can be easily explained by a root locus with increasing K_L assuming that the position-controlled robot is a linear mass-damper-spring system. Therefore, the compliance of the manipulator becomes one of the important parameters, and more compliance is desirable in the cutting process with pure position control.

In this paper, the compliance of a manipulator is derived analytically, and its magnitude is represented as a compliance ellipsoid. It is shown that the compliance will be different depending on feedback gains, link flexibility, configurations, and direction of interest. Then, using a two link flexible manipulator with an abrasive cut-off saw, the experimental investigation shows that the magnitude of the chattering varies with the saw cutting angle due to the different compliance. The final results show a range of cutting angles with acceptable behavior for a point in the workspace with a near circular compliance ellipsoid.

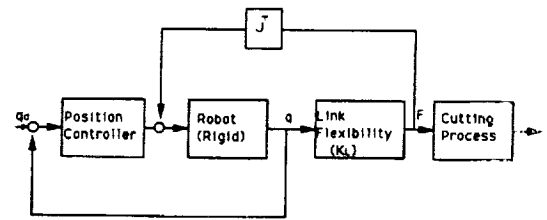


Figure 1. Block Diagram of Cutting Process with a Position-Controlled Manipulator

1.2 Chatter

In the utilization of metal-cutting machine tools, vibrations are often encountered. The contact between the tools and workpiece gives rise to excessive variations of the cutting force which endanger the life of the tool. These vibrations belong to the class of self-excited vibrations. The source of the self-exciting energy is in the cutting process. Furthermore, in many cases, the self-excited vibrations are mixed with forced vibrations excited by various sources such as continuous spinning of the tools [3]. In this paper, the self-excited and forced vibrations are referred to as chatter.

Considerable knowledge about the influence of kinematical parameters on the chatter has been assembled. As yet, however, neither a complete theoretical description and analysis has been accomplished nor reliable ways found for eliminating chatter in grinding [3,4]. By experimental trial and error, general guidelines have been established to reduce the tendency for chatter. Among these are the use of soft-grade wheels, frequent dressing of the wheel, changes in dressing techniques, reduction in material removal rate and more rigid support of the workpiece [8]. Even though many parameters influence the chattering, this paper examines mainly the relationship between the cutting angle and the compliance of the arm.

3. DYNAMIC MODEL

3.1 Cutting Process

Exact modeling of a cutting process can be very complicated [3,4,5,6]. For simplicity, this paper assumes that the cutting forces consist of the normal cutting force which is in the direction of the approach angle of the saw and the tangential cutting force which is perpendicular to the approach angle. The relationship between the normal cutting force (F_n) and tangential cutting force (F_t) can be assumed to be Coulomb friction [7], i.e.

$$\mu = \frac{F_t}{F_n} = 0.3 \sim 0.4$$

Notice that F_n is larger than F_t . Also, F_t and F_n are a function of the depth of the cut.

3.2 Flexible Manipulator

Modeling a multiple link flexible manipulator is a complicated procedure. The deflection of the arm is approximated as

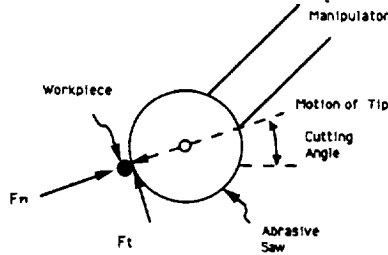


Figure 2. Definition of Cutting Angle

a finite series of separable functions which are the products of mode shape functions $\psi_j(x)$ and time dependent generalized coordinates $q_{jv}(t)$:

$$u_i(x, t) = \sum_{j=1}^m \psi_j(x) q_{jv}(t)$$

where i represents the link number and j represents the mode number. The equation of the flexible arm motion can be derived from several techniques, but the Lagrange's formulation is known for its simplicity and systematic approach [9,10]. Using Jacobians to compute the velocity of a point, the kinematic and potential energies are obtained by integrating the velocity and position of the point over the total system. These energies are used in Lagrange's equations. Therefore, the equation of the motion is

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{fr} & M_{ff} \end{bmatrix} \begin{pmatrix} \ddot{q}_r \\ \ddot{q}_f \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_L \end{bmatrix} \begin{pmatrix} q_r \\ q_f \end{pmatrix} + \begin{pmatrix} N_r + G_r \\ N_f \end{pmatrix} = \begin{bmatrix} I \\ \frac{\partial w}{\partial x} \end{bmatrix} T + \begin{bmatrix} J_r^T \\ J_f^T \end{bmatrix} F \quad (1)$$

where q_r contains the generalized rigid joint coordinates and q_f contains the generalized flexible mode coordinates. M_{ij} is the inertia matrix and K_L represents the link flexibilities. N_r and N_f include nonlinear terms such as the Coriolis and centrifugal force in each coordinate. G_r is the gravity force. T represents joint torques and F represents an end point external force when the contact with environments occurs. Finally, J_r and J_f are the partitions of the Jacobian matrix for a flexible arm.

4. COMPLIANCE OF ARM

The equation of the dynamic motion for a flexible arm is obtained in equation (1). Since the tip of the manipulator proceeds very slowly and in a small range during cutting, the motion can be assumed to be quasi-static and linear. Therefore, the equation of the motion is simplified to the following form assuming the acceleration terms and the velocity terms are negligible.

$$\begin{bmatrix} \frac{\partial G_r}{\partial q_r} \\ 0 \end{bmatrix} \Delta q_r + \begin{bmatrix} 0 & 0 \\ 0 & K \end{bmatrix} \begin{pmatrix} \Delta q_r \\ \Delta q_f \end{pmatrix} = \begin{bmatrix} I \\ \frac{\partial w}{\partial x} \end{bmatrix} \tilde{T} + \begin{bmatrix} J_r^T \\ J_f^T \end{bmatrix} F \quad (2)$$

As shown above in the above equation (2), the gravity force also contributes to a stiffness force. If a joint angle PD controller is applied to the flexible arm, the joint torque \tilde{T} will be

$$\tilde{T} = -K_p \Delta q_r - K_v \Delta \dot{q}_r$$

where K_p and K_v are the feedback gains. This input torque can be interpreted as a spring and a damper force. Again \dot{q}_r can be neglected in quasi-static motion. If we combine all these forces for the stiffness matrix,

$$\begin{bmatrix} \frac{\partial G_r}{\partial q} + K_P & 0 \\ 0 & K_L \end{bmatrix} \begin{pmatrix} \Delta q_r \\ \Delta q_f \end{pmatrix} = \begin{bmatrix} J_r^T \\ J_f^T \end{bmatrix} F$$

This stiffness matrix is always invertible. Therefore, it can be rewritten as-

$$\begin{pmatrix} \Delta q_r \\ \Delta q_f \end{pmatrix} = \begin{bmatrix} \frac{\partial G_r}{\partial q} + K_P & 0 \\ 0 & K_L \end{bmatrix}^{-1} \begin{bmatrix} J_r^T \\ J_f^T \end{bmatrix} F \quad (3)$$

Since this matrix shows the relationships between the end-point external force and the joint coordinates, it is necessary to change the joint space to the Cartesian space. Using the Jacobian relationship which is

$$\Delta X = [J_r \quad J_f] \begin{pmatrix} \Delta q_r \\ \Delta q_f \end{pmatrix} \quad (4)$$

Substituting equation (3) into equation (4) will give

$$\Delta X = [J_r \quad J_f] \begin{bmatrix} \frac{\partial G_r}{\partial q} + K_P & 0 \\ 0 & K_L \end{bmatrix}^{-1} \begin{bmatrix} J_r^T \\ J_f^T \end{bmatrix} F \quad (5)$$

Since compliance is defined as 'displacement per input force', we may say

$$C = [J_r \quad J_f] \begin{bmatrix} \frac{\partial G_r}{\partial q} + K_P & 0 \\ 0 & K_L \end{bmatrix}^{-1} \begin{bmatrix} J_r^T \\ J_f^T \end{bmatrix} \quad (6)$$

This matrix, C , is called the compliance matrix for a flexible arm with a joint angle PD controller. This compliance matrix includes not only link flexibilities, but also feedback gains, stiffness and configurations of the arm. The link flexibilities are represented by K_L and the feedback gains by K_P . The configurations of the arm are represented by J_r and G_r . However, the compliance matrix does not incorporate the mass properties of the arm which also may influence the arm's behavior. If equation (6) is expanded, it can be rewritten as

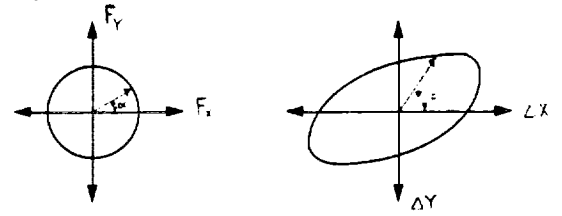
$$C = J_r \left(\frac{\partial G_r}{\partial q} + K_P \right)^{-1} J_r^T + J_f K_L^{-1} J_f^T \quad (7)$$

This form of the compliance matrix shows a major difference between a flexible arm and a rigid arm. In a rigid arm case, K_L is assumed to be very large. Therefore, we may ignore the second term of equation (7) although it is the dominant term in a flexible arm case.

Since the compliance is represented with a matrix for a multiple link manipulator, various input force directions cause different directions and sizes of displacements. This may be explained with a linear algebra concept. From equation (5), we see that the compliance is simply a linear transformation that maps the end-point force F in R^3 into a Cartesian space displacement in R^3 . The unit sphere in R^3 defined by

$$F^T F \leq 1$$

is a mapping into an ellipsoid in R^3 defined by



(a) Unit Input Force Sphere (b) Compliance Ellipsoid

Figure 3. Compliance Ellipsoid

$$\Delta X^T (C C^T)^{-1} \Delta X \leq 1$$

This ellipsoid has principal axes $\lambda_1 e_1, \lambda_2 e_2, \lambda_3 e_3$ where e_1 is a unit vector and λ_i is an eigenvalue of $(C C^T)$. We call this the 'compliance ellipsoid'. Therefore, a unit force F in the direction α induces a displacement in the direction β as shown in Figure 3.

5. A CASE STUDY

A large experimental arm designated RALF (Robotic Arm, Large and Flexible) has been constructed and is under computer control. RALF has two degrees of freedom in the vertical plane. The length of each link is about 10 feet. At the tip of RALF, an abrasive cut-off saw is mounted as shown in Figure 4. Using the compliance ellipsoid, we explore the desirable cutting angles for RALF with acceptable chattering behavior.

5.1 Analysis

Based on the actual dynamic parameters of RALF, the dynamic equation is derived in the form of equation (1). Then, actuator dynamics are assumed to be constant gains since their bandwidth is very high compared to the arm dynamics. The actuator gains are included in the feedback gains K_P . The nominal configuration during cutting is the following: the first joint angle is 106.6 degrees, and the second joint angle is 101.8 degrees. The compliance matrix is computed and its magnitude is represented in R^2 with an ellipsoid. Simulation results show that the manipulator's axis of least compliance is at an angle 30 degrees with the horizontal and the axis of greatest compliance is at an angle of 120 degrees in Figure 5(a). Therefore, the 120 degrees cutting angle is desirable to produce the least chattering due to its greater compliance. Different shapes of the compliance ellipsoid can be obtained at different configurations. For example, if the first joint is at 110 degrees and the second joint is at 50 degrees, the compliance ellipsoid can be shown as in Figure 5(b).

5.2 Experiments

To measure the chattering in plane motion, two accelerometers are mounted at the tip of RALF. One accelerometer measures the X direction vibration and the other accelerometer measures the Y direction vibration referenced to the manipulator base coordinates. Experiments use three different cutting angles 0, 40 and 90 degrees. The cutting angles are shown as in Figure 6 and the manipulator follows the given trajectory. Each

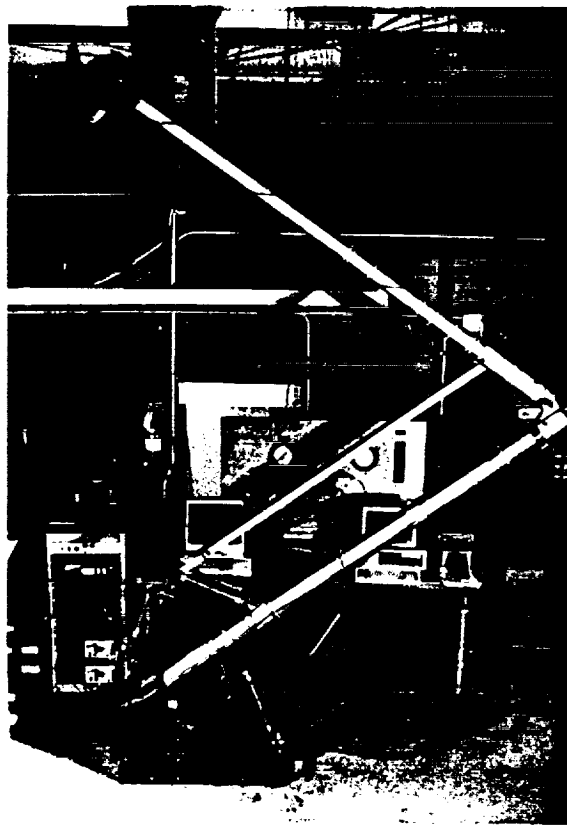


Figure 4. RALF with an Abrasive Cut-off Saw

case assumes that it has the same cutting parameters except for the cutting angle. For instance, cutting velocities are the same for each case, and the same feedback gains are used too. The workpiece is a half inch diameter steel bar and is much stiffer than the manipulator system itself.

First, the abrasive cut-off saw moves very close to the workpiece. Then, the saw is turned on without contact with the workpiece. The vibrational signal is measured by a signal analyzer, and its power spectrum is plotted in Figure 7(a). The first natural frequency is observed at 4.5 Hz compared to 5 Hz from the mathematical model. Also, another peak is observed at about 62 Hz. This frequency is believed to originate from dynamic imbalance of the saw motor turning at 3800 rpm (63.3 Hz) by the manufacturer's data.

Second, the cut-off saw followed the 0 degree desired trajectory by a joint angle PD control. The trajectory is computed based on Dickerson and Oosting's work [11]. It takes about 10 sec for the saw to go through the workpiece. The acceleration power spectra are measured for 2 sec four times during cutting and are averaged to eliminate the influence of non-periodic noise. The same procedures are used for each experiment. In Figure 7(b), first peak is measured at 9 Hz. We may interpret this shift of the first frequency due to the change in the bound-

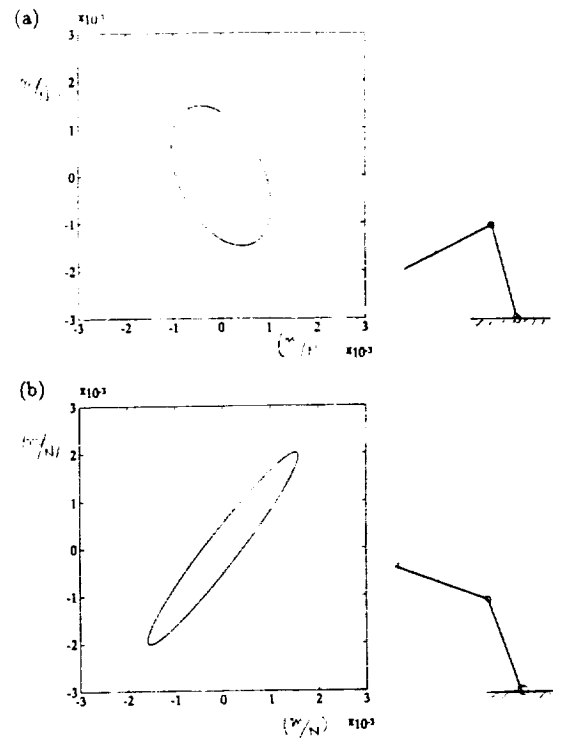


Figure 5. Compliance Ellipsoid for RALF
 (a) when $\theta_1 = 106.8$ deg and $\theta_2 = 101.8$ deg
 (b) when $\theta_1 = 110$ deg and $\theta_2 = 50$ deg

ary condition when the saw touches the workpiece. Also, we may notice that the rotation speed of the wheel is reduced due to the contact friction force.

Third, when the saw cuts the workpiece at 40 degrees, the first natural frequency (9 Hz) no longer dominates as before and is mixed with other frequency signals in Figure 7(c). However, the higher mode at 37 Hz becomes more noticeable. In other words, chattering becomes faster.

Fourth, the 90 degree cutting shows smaller magnitudes of vibration in the Y direction, and the mode at 22 Hz becomes important (See Figure 7(d)). We expect that this angle will give the least chattering based on analytical analysis. Experimental

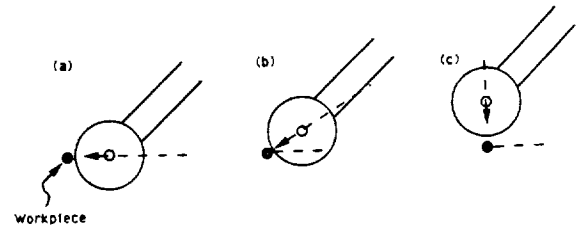


Figure 6. Various Cutting Angle
 (a) 0 deg (b) 40 deg (c) 90 deg

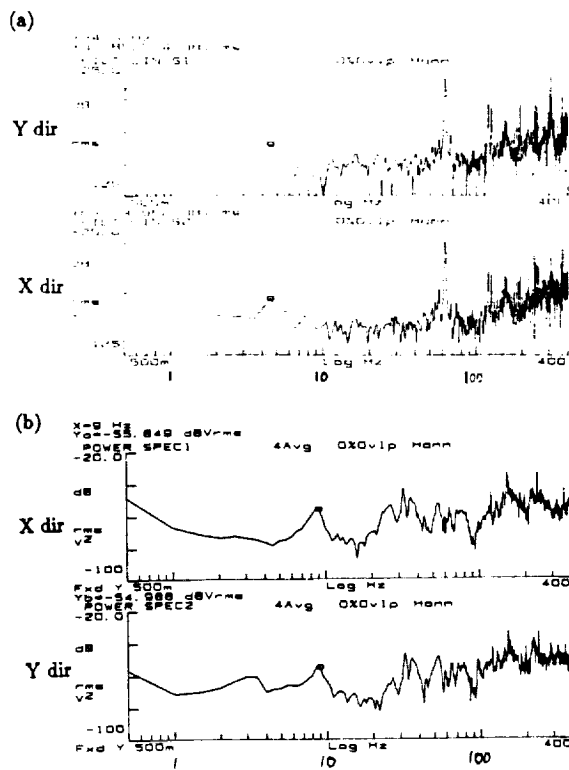


Figure 7. Power Spectrum Measurement in X-Y direction
 (a) Without Any Contact with Workpiece
 (b) 0 degrees Cutting Angle

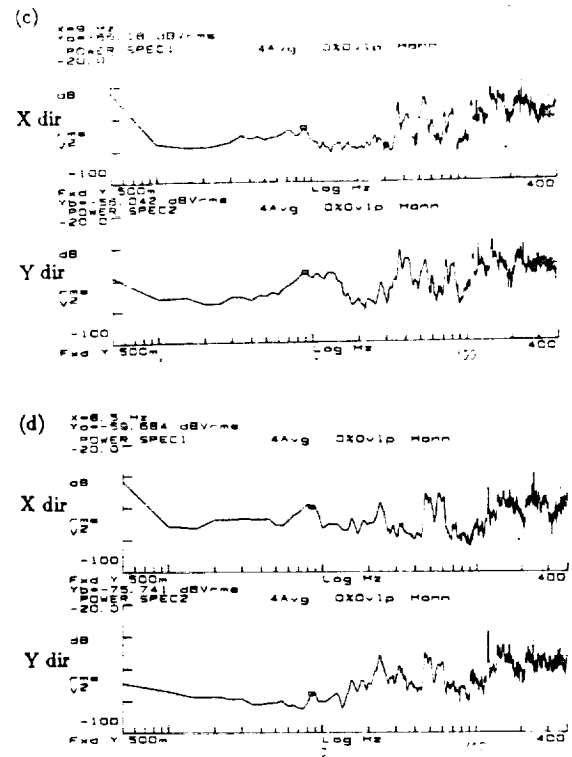


Figure 7. Power Spectrum Measurement in X-Y direction
 (c) 40 degrees Cutting Angle
 (d) 90 degrees Cutting Angle

data fails to show a distinct advantage.

Finally, various cuts have been performed by a tele-operated joystick under human control. Most of the cutting processes are successfully accomplished without any severe chattering. However, its measurement is not included in this paper due to space. Experimentally, the contact velocity is one of critical factors which initiates chattering.

6. CONCLUSIONS

Taking advantage of the passive compliance of the flexible manipulator, certain applications such as cutting a workpiece are performed with pure position control. This provides a simple, inexpensive solution for certain applications that otherwise could not be achieved with position control alone.

Both computer-controlled cutting and human-operated cutting were performed with minor chattering. However, contact velocity should remain very small to reduce chattering.

The contact with the workpiece causes a shift of the first natural frequency of a flexible arm due to the change of the boundary conditions. Different cutting angles produce different frequencies of vibrations due to the different compliances in the

direction of forcing.

Analytical studies predict 120 degrees cutting as the most desirable. However, this experimental investigation could not show distinct differences in the magnitude of chattering, although we may say that the 90 degree cutting angle generates a smaller magnitude of chattering in the Y direction. The compliance ellipsoid for our test bed is not elongated enough to make distinct differences in compliance. A different configuration could have made a more elongated ellipsoid, but further experimental tests have not been conducted due to physically limited location of the workpiece.

Acknowledgment

This work was partially supported by NASA Grant NAG 1-623. Also, we would like to thank PCB PIEZOTRONICS, INC for their generous donation of accelerometers.

REFERENCES

- [1] J. Craig, "Introduction to Robotics, Mechanics and Control", Addison-Wesley Publishing Co., 1989

- [2] M. Spong and M. Vidyasagar, "Robot Dynamics and Control", John Wiley and Sons, 1989
- [3] F. Koenigsberger and J. Tlustý, "Machine Tool Structures", Volume 1, Pergamon Press, 1970
- [4] J. Peter, R. Snoeys and A. Decneut, "The Proper Selection of Grinding Conditions in Cylindrical Plunge Grinding", Proceedings of the 16th Internal Machine Tool Design and Research Conference,
- [5] Y. Liao and L. Shiang, "Computer Simulation of Self-Excited and Forced Vibrations in the External Cylindrical Plunge Grinding Process", The Winter Annual Meeting of ASME, San Francisco, CA, Dec 10-15, 1989
- [6] A. Shunsheruddin and S. Kim, "Adaptive Control of the Cylindrical Plunge Grinding Process",
- [7] R. Hahn and R. Linsay, "Principles of Grinding..part1 Basic Relationships in Precision Grinding", Machinery, July-November 1971
- [8] S. Kalpakjian, "Manufacturing Processes for Engineering Materials", Addison- Wesley Publishing Co., 1984
- [9] J. Lee, "Dynamic Analysis and Control of Lightweight Manipulators with Flexible Parallel Link Mechanisms", Ph.D Thesis, School of Mechanical Engineering, Georgia Institute of Technology, 1990
- [10] W. Book, " Recursive Lagrangian Dynamics of Flexible Manipulators", International Journal of Robotics Research, Vol. 3 No.3, 1984
- [11] K. Oosting and S. Dickerson, "Simulation of a High-Speed Lightweight Arm", In Proceeding of the 1988 International Conference on Robotics and Automation, Vol.1, 1988

N91-30530

PLANNING AND EXECUTING
MOTIONS FOR
MULTIBODY SYSTEMS
IN FREE-FALL

Ph. D. Dissertation Proposal

Jonathan M. Cameron

January 31, 1991

CONTENTS

ABSTRACT	1
1 Introduction	2
1.1 Purpose	2
1.2 Motivation	2
2 Current Approaches (Literature Review)	5
2.1 Motion Planning and Control for Robots in free-fall	5
2.2 Path Planning for Mobile Vehicles	7
2.3 Trajectory Planning and Control for Fixed-Base Robots	8
2.4 Tabular Planning and Control	8
2.5 Symbolic Manipulation	9
2.6 Multibody Dynamics	10
3 Proposed Research	11
3.1 Overview	11
3.2 Proposed Approach	11
3.2.1 Definitions	11
3.2.2 Analyze motion possibilities	12
3.2.3 Implement simulation system	13
3.2.4 Implement symbolic construction of equations of motion	13
3.2.5 Design optimal controls to accomplish motions	15
3.2.6 Implement symbolic generation of optimal control scheme	16
3.2.7 Precompute motions between selected configurations	16
3.2.8 Adapt compression techniques to compress motion data	18
3.2.9 Design linearized motion tracking control scheme	19
3.2.10 Implement symbolic generation of linearized controller	20
3.2.11 Use simulation to verify linearized controller	20
3.2.12 Apply system to example multibody systems	20
3.3 Expected Results and Contributions	21
REFERENCES	22

(Continued)

APPENDICES

A	How the Simulation Environment Might Be Used	32
B	Sample Optimal Control Analysis	36
C	Applied Optimal Controls Example	42
	C.1 Description	42
	C.2 MACSYMA Usage Descriptions and Code	42
	C.3 Sample MACSYMA Session Output	48
D	Movement Library Size Requirements	51

Abstract

How do multibody systems move in free-fall? For instance, when a cat falls, it flips over before it reaches the ground. How does it do that? Multibody systems in free-fall move very differently than robots which are bolted to the ground. Once a robot with a fixed base stops moving, the link positions can be determined by kinematics alone. This is not true for a robot or multibody system in free-fall. The final link positions of a robot in space depend on the link trajectories during the motion as well as its kinematics. Kinematics and dynamics are tightly coupled for multibody systems in free-fall. Given these difficulties, how can we plan motions for multibody systems in free-fall?

The proposed research will center on several issues necessary to plan and execute motions for multibody systems in free-fall.

1. *What motions are possible for a multibody system in free-fall?* Mathematical techniques from nonlinear control theory will be used to study the nature of the system dynamics and its possible motions.
2. *How can we plan the link motions and joint torques necessary to move from one configuration to another?* Optimization techniques will be applied to plan motions.
3. *How can we store precomputed motion plans efficiently?* Since it is unlikely that motion plans can be computed in real time, pre-computation will be necessary. Image compression techniques are proposed to compress the precomputed motion data for storage.
4. *Once a motion is planned, how can the system execute the motion faithfully?* A linearized controller will be devised to control the system while it executes preplanned trajectories.

Symbolic manipulation techniques will be used in the research (where practical) to reduce chances for algebraic errors and to make the approach easier to apply to new multibody systems in free-fall.

The proposed research applies to a number of activities. Most obviously, it can be used to plan motions for robots in space. It can be used to plan limb motions to reorient astronauts. The research may also be useful to plan the movements of airborne divers, gymnasts, and jumpers.

1 Introduction

1.1 Purpose

The motion of ground-based robots is reasonably well understood. The underlying kinematic and dynamic analysis relies on the fact that robots are bolted to the floor—which will not move appreciably despite the motions of the robots. This is not true for robots in space. More generally, this is not true for multibody systems in free-fall. When a multibody system is not attached to the earth, its motions are considerably more complex than its ground-based counterparts. The dynamics and kinematics become inextricably coupled.

Planning motions for multibody systems in free-fall is more difficult than for fixed-base robots largely because of the interaction of the kinematics and dynamics. Planning feasible or optimal trajectories will require extensive off-line computation and cannot be done in real time. A way is needed to precompute and store the possible motions so that the possibilities can be retrieved later and used quickly for real-time planning and execution purposes.

The purpose of this research is to develop an end-to-end system that can be applied to a multibody system in free-fall to analyze its possible motions, save those motions in a database, and design a controller that can execute those motions. A goal is for the process to be highly automated and involve little human intervention. Ideally, the output of the system would be data and algorithms that could be put in ROM to control the multibody system in free-fall.

This research applies to more than just robots in space. It applies to any multibody system in free-fall. This includes astronauts in space, falling mechanisms, athletes in jumps or dives, and airborne gymnasts.

1.2 Motivation

To illustrate the complexities and potential of the types of motion that will be addressed by this research, consider the following examples.

Falling cat problem

When a cat is held upside down and dropped, it manages to turn itself rightside up before it lands. This well-known phenomenon has long intrigued children and scientists. Thomas Kane resolved the question in 1969 with a dynamic simulation that showed the motions necessary to turn the cat over during its fall [35]. His dynamic simulation involved only two bodies but duplicated the complex motion of the cat during the maneuver.

Astronaut attitude change maneuvers

Astronauts are taught a series of maneuvers that allow them to change their orientation while in free-fall. Again, it was Kane who developed these maneuvers [36, 37]. For each of the desired rotations (pitch, roll, and yaw), he developed simplified equations of motion and then analyzed them to determine what cycles of limb motions were necessary to give the desired orientation change. The result was several cycles of simple limb motions that produce orientation changes about the desired axis.

Divers, Gymnasts, and Jumpers

Some of the most complex motions of systems in free-fall occur when spring-board divers are in the air [21, 132, 133, 134, 135]. Gymnasts and athletes perform similar maneuvers while in the air during their activities. The movements of high jumpers while off the ground are also complex. The "Fosbury Flop" revolutionized high jumping by improved maneuvers while in the air [125].

Robot servicing in space

Using robots to service satellites in orbit is a goal of NASA [56, 101]. Several robot designs have been proposed which involve complex arms attached to large bodies with control-moment gyros and thrusters for maneuvering and station-keeping [8, 33, 57, 58]. These approaches to motion control have their drawbacks. Control-moment gyros are complex and expensive. Thrusters

produce plumes which may impinge on delicate equipment. If techniques could be developed to plan and execute attitude and configuration changes via limb motions, simpler, safer, and less expensive servicing systems could be designed.

2 Current Approaches (Literature Review)

Many researchers have addressed various aspects of this research. Relevant research is reviewed below.

2.1 Motion Planning and Control for Robots in free-fall

Only a few researchers have directly addressed the problem of how to plan and control motions of a robot or multibody system in free-fall.

Kane's work in astronaut maneuvering has already been mentioned. This research was reported in 1970-72 [36, 37]. He worked out the equations of motion for the human body and simplified them for the desired rotations of the body trunk. The resulting cyclic motions are interesting and useful but very different from the types of motion desired in this research. In any case, his work was specific to the human body and did not address the question of general configuration change of multibody systems in free-fall.

Longman, Lindberg, and Zedd considered a robot arm mounted to a satellite and developed special kinematics that dealt with the dynamics-kinematics interaction problem [57, 54, 58]. They assume the satellite base body contains reaction wheels to keep it from rotating when the arm moves. Their kinematics compensate for the translational movement of the base during movements of the arm. Their kinematic simplifications depend on the absence of base rotation and do not generalize to multibody systems without reaction wheels.

Vafa and Dubowsky developed the *virtual manipulator* technique for analyzing the kinematics and dynamics of robots in space [117, 118, 119, 120, 121, 80]. They devised a way to construct an imaginary manipulator with dimensions and inertial characteristics related to the actual system. The motion of the imaginary system and the real system are closely related and exactly the same for one point of the actual manipulator (such as the end effector). Once the position of this common point is determined, the necessary virtual manipulator configuration is easily computed and then the corresponding joint positions of the actual system can be determined. The dynamics of

virtual manipulators also have the advantage that the conservation of linear momentum is implicitly integrated and eliminated from the equations of motion. Unfortunately, this approach does not provide any solutions for how to move from one configuration to another. In other words, the virtual manipulator approach can be used to determine the final joint positions to accomplish some task but is not very helpful in determining the necessary joint motions to move from the starting configuration to the final configuration. They did apply this technique to manipulator motion planning by using small cyclic motions to produce small motions of the manipulator base. Although this approach is useful for planning the motion of space manipulator end-effectors, it has limited usefulness for planning large motions in which the entire final configuration is specified (such as in gymnastics).

Given an end effector position that can be reached, the virtual manipulator approach can be used to determine the necessary joint angles of the space manipulator to achieve that position.

Umetani and Yoshida studied continuous path control of manipulators. They devised an extended Jacobian for kinematic analysis of the motion of the end-effector [116]. The generalized Jacobian enforces the conservation of linear and angular momentum for the space manipulator. They use the new Jacobian to plan continuous motions of the end-effector in space and simulate them for a OMV (Orbital Maneuvering Vehicle.) This approach has similar benefits and limitations as Vafa's work because it concentrates on end-effector motion.

Nakamura and Mukherjee also addressed the problem of planning motions for space robots [71]. Their work deals specifically with the nonholonomic nature of the conservation of angular momentum. They devise kinematics which incorporate the linear and angular momentum and of the space manipulator. They then devise a controller based on a Lyapunov function. Their approach is promising but initial results were disappointing because the controller could get stuck during the motion. They solved this problem in their later work [72, 73] by designing a bi-directional control algorithm. Unfortunately, the resulting motions involve unusual cyclic motions and other peculiarities which indicate that the motion is not very general.

Sreenath and Krishnaprasad (among others) use mathematical approaches to attack the problem of controlling the motion of multibody systems in

free-fall [102, 103, 45, 131]. While these methods are difficult to understand, they do appear to offer promising techniques. Unfortunately, their research is not very useful to the proposed research because they currently consider only planar systems. In "Nonlinear Control of Multibody Systems in Shape Space," N. Sreenath indicates that extending their results to three space is "definitely non-trivial." [102, p. 1780].

In a recent paper[67], Murray and Sastry use Chow's theory and Lie brackets to determine whether a motion is possible for a system subject to a nonholonomic constraint linear in velocities. If the motion is possible, they construct a path using sinusoidal path segments. They apply their technique to the motion of planar systems in free fall and to the nonholonomic vehicle problem. Their approach has useful insights and may be applicable to the proposed research but has not yet been applied to non-planar motion.

2.2 Path Planning for Mobile Vehicles

An area related to the current research is path planning for mobile robots. Much research has been devoted to planning land vehicle motion for indoor and outdoor vehicles. Works which considers mobile vehicles as points are not considered here since they are irrelevant to the proposed research. There are a few researchers who address the nonholonomic nature of vehicles with limited steering capabilities.

Laumond considers a nonholonomic vehicle and proves that it is possible to plan collision free paths through a cluttered area by combining sets of small cyclic motions [46]. In later work he showed that whenever it is possible to plan a jagged path, it is also possible to plan a smooth path for the same motion [47]. Barraquand and Latombe addressed similar issues and devised planning techniques based on potential field techniques [5]. They apply their approach to difficult problems such as parallel parking a vehicle with several trailers. They also applied their approach to robot arms with many degrees of freedom. Similar research is covered by Jacobs and Canny [30, 31]. This type of research has many insights to offer for nonholonomic systems. Unfortunately, the nonholonomic constraints due to limited steering angles are simpler than the nonholonomic constraints due to the conservation of angular momentum. These approaches have not been applied to multibody

systems in free-fall and it is not clear how applicable they are.

2.3 Trajectory Planning and Control for Fixed-Base Robots

An extensive amount of research has been done on planning motions for fixed-base robots. A sampling of this research is given in the references [6, 17, 34, 40, 22, 23, 78, 64, 74, 75, 91, 89, 97, 98, 100, 105, 114, 126, 127].

One of the most promising approaches is presented by Tan and Potts [106, 107, 108]. Their approach does everything. Their technique is intended for fixed-base robots but is general enough to handle multibody systems in free-fall. Their technique handles full dynamic nonlinearities, actuator limitations, joint constraints (position, velocity, and jerk), avoids obstacles, and incorporates an energy objective as well. This approach has not been adapted to multibody systems in free-fall but appears useful for this research.

Another promising approach to planning optimal motions is given by Luus in recent research on controlling chemical processes [61]. His approach is based on dynamic programming and may be useful for the problem at hand. Luus has applied his technique to problems with up to eight nonlinear ordinary differential equations and determined optimal controls with limits on input variables.

2.4 Tabular Planning and Control

Another area relevant to this research is precomputing motion data and storing for later use in planning and control. This is sometimes called a *tabular* approach since motion data are precomputed and stored in tables for later retrieval in planning or control. Very little has been done in this area.

Raibert does use tabular techniques with some success for control of the cyclic parts of motion of his one-legged hopping machine [81, 82, 83]. Tabular techniques were also proposed by Albus [1, 2]. Hollerbach criticizes tabular approaches in general (and these in particular) when he concludes that dynamics simulation codes can be made fast enough to run in real time [27]. This criticism is not relevant to the proposed research. It may be pos-

sible to simulate the motion of multibody systems in free-fall faster than real time if the torques or forces to apply at each actuator during the motion are known before hand. The proposed research is to determine these actuator inputs. This cannot be done in real-time using known techniques even on super computers.

2.5 Symbolic Manipulation

Symbolic manipulation offers researchers many opportunities to improve the quality of their work by producing results much faster than is possible by hand, reducing the chance of mathematical errors, and allowing handling of more difficult problems. Applying symbolic manipulation to robot kinematics and dynamics is not new.

Hussain and Noble used symbolic computation for forward and inverse kinematic analysis of specific robot geometries which assisted the user but still required considerable interaction[28]. Direls developed a system for manipulation of matrices with symbolic entries and used this to analyze robot kinematics [16]. Kircanski and Vukobratovic constructed a system using FORTRAN-77 to symbolically generate the forward kinematics and Jacobian of a robot but not the inverse kinematics solution [41]. Lloyd and Hayward applied MACSYMA to the same problem and derive kinematics for several common robot architectures [55]. Tunstel and Vira also use MACSYMA to construct robot kinematics symbolically as an educational and design aid [113]. They also introduce a number of rules (symbolically implemented) that simplify the results.

Many researchers have also developed dynamic equations of motion for multibody systems symbolically. Liegois and company developed PL/1 software to derive equations of motion using a Lagrangian formulation. Others have written FORTRAN programs for symbolic generation of equations of motion for multibody systems using various approaches: Newton-Euler [43, 44] and Kane's equations [18]. Other similar work has been done by various researchers [7, 11, 24, 26, 29, 49, 50, 62, 66, 76, 77, 69, 88, 90, 94, 95, 96, 109, 110, 111, 112, 122, 128, 129, 136, 137] Others have applied similar techniques to systems with flexible components [12, 59, 115]. Many of these systems generate the equations of motion encoded in a FORTRAN or C program suited

to compiling and running for simulation purposes. Symbolic manipulation has also been applied to control applications [94, 104, 107].

2.6 Multibody Dynamics

Multibody Dynamics is a huge field. Many people have developed widely varying approaches to the problem of simulating and controlling multibody systems. Several references cover Multibody dynamics in detail [4, 19, 93, 85, 130]. Others, too numerous to mention, deal with dynamics in general and are applicable to multibody dynamics. Although serious multibody dynamics research was done more than 80 years ago [20], the field is not exhausted. Recent developments include many recursive techniques for inverse dynamics with operations counts proportional to the number of elements [3, 15, 19, 25, 27, 38, 39, 53, 60, 65, 86, 87, 124, 123]. (Most of these are based on recursive Newton-Euler approaches; some are based on operation space approaches [19, 53, 86, 87].) The most efficient of these approaches is given by He and Goldenberg [25]. Their recursive technique requires $91(n - 1) - 6$ multiplications and $86(n - 1) - 10$ additions, where n is the number of bodies. The efficiency of these recursive techniques allows the computation of joint torques necessary to produce desired motions in real-time for reasonably complex systems. Forward dynamics algorithms are not quite as efficient yet [48].

3 Proposed Research

Before analyzing the proposed research in detail, an overview may be helpful to orient the reader.

3.1 Overview

The goal of the research is to develop and test a system which can precompute, save, and execute motions for multibody systems in free-fall. The basic components of the research are listed below.

1. Analyze motion possibilities
2. Implement simulation system
3. Implement symbolic construction of equations of motion
4. Design optimal controls to accomplish motions
5. Implement symbolic generation of optimal control scheme
6. Precompute motions between selected configurations
7. Adapt compression techniques to compress motion data
8. Design linearized motion tracking control scheme
9. Implement symbolic generation of linearized controller
10. Use simulation to verify linearized controller
11. Apply system to example multibody systems

3.2 Proposed Approach

3.2.1 Definitions

Several terms are used in specific ways in this proposal and are defined here. The terms appear in italics in the following definitions.

Configuration (or *pose*) refers to the shape of the body as determined by the joint positions. *Orientation* refers to the attitude of the system with respect to some global reference frame. More precisely, orientation refers to the attitude of some reference link of the body with respect to a global reference frame. If a robot is bolted to the floor, there is no reason to make the

distinction between configuration and orientation. Once the base of a robot or multibody system is free to move with respect to the global reference frame, this distinction becomes useful and important.

Typical Configurations are configurations of the multibody system that occur often during motions of the system and are useful in studying and planning its motions. For instance, a tuck is a typical configuration for divers. For more detail, see Appendix A, page 32.

Motions refer to movement from one combination of configuration and orientation to another combination of configuration and orientation. In this research, this will be accomplished strictly by joint motions.

3.2.2 Analyze motion possibilities

What motions are possible for multibody systems in free-fall? That question is central to this research. The possible motions depend on the nature of the mechanism, the initial configuration and orientation and the final configuration and orientation. For instance, a mechanism with one revolute joint (like a hinge) has a limited range of motion. It can open and close but the axis of the hinge cannot be tilted by opening and closing the hinge. This is because its motion is holonomic. A nonholonomic system has more potential motions. Consider a vehicle on the plane with a limited steering angle. The front wheel imposes a motion constraint that is nonholonomic. The vehicle has three degrees of freedom in the large but only two degrees of freedom at any instant. Yet, by careful maneuvering, any position in the plane can be reached. Multibody systems in free-fall must conserve angular momentum because they have no external torques acting on them. The conservation of angular momentum can be thought of as a nonholonomic constraint on the motion of the system in free-fall. Depending on the character of the angular momentum, a mechanism in free-fall may be able to move from any combination of configuration and orientation to any other combination of configuration and orientation: or it may not—as in the case of a hinge in free-fall. Obviously, since no external forces are used, the system center of mass will not move either case.

This research will investigate this issue further and devise tests to be applied to determine if each of the desired motions is possible. For example, Probe-

nus' theorem can be applied using Lie brackets to evaluate the nonholonomic nature of the angular momentum (whether it is integrable) [51, 5, 71, 99, 70]. This can be done symbolically[42] since the angular momentum can be generated symbolically. Research will also address the general controllability and reachability for these systems. It should be noted that it is very difficult to perform this type of research without symbolic manipulation due to the complexity of the equations of motion and angular momentum.

3.2.3 Implement simulation system

A basic part of the proposed research is a simulation environment in which the various components of the research will be implemented and tested. This simulation system will allow the user to construct robots from links and joints and then simulate kinematics and dynamics of the robots. The simulation environment will be used to verify the resulting motion libraries and control schemes. An extended description of how the simulation environment can be used is included in Appendix A.

The initial implementation of the simulation environment will handle multi-body systems composed of rigid bodies since that is the focus of this research. To be even more useful, the simulation environment should also be able to handle flexible members. The software design and implementation will make provisions for future expansion in this direction.

The software approach will be object-oriented and the code will be written in an appropriate computer language such as C++, object-oriented Pascal, or Ada. An important component of such a system is the graphical display. These considerations and the goal of source-code portability indicate that C++ and X-Windows might be a good choice.

3.2.4 Implement symbolic construction of equations of motion

A number of systems exist for studying the motion of multibody systems. These include SD/Fast, SD/Exact, Autolev, DADS, and ADAMS. Others are mentioned in the literature review. These systems simulate multibody motions, and some generate C or FORTRAN code for simulation and control purposes. Unfortunately, the output of most of these systems is not

directly suitable for further symbolic manipulation (for controls analysis, for instance.)

The proposed system will generate equations of motion in symbolic form suitable for further symbolic manipulation. (A few of the systems mentioned in the literature review do this.) The resulting symbolic form of the equations of motion will be used in three ways. First, the equations of motion will be used to generate executable code for simulation and planning purposes. Second, the equations of motion will be used to analyze system controllability. Third, the equations of motion will be used to construct a linearized controller for trajectory tracking purposes. The last two will be done symbolically and the resulting symbolic material will be converted to appropriate code as necessary.

The dynamical formulation that will be used has not been determined yet. A significant part of the research will involve comparing the various approaches and choosing the most appropriate one to implement symbolically. Approaches to be compared include Newton-Euler, Lagrange equations (with Routh's extensions), Hamilton's canonical equations, Kane's equations, and spatial algebra/screw theory approaches.

To be suitable, the chosen technique of generating equations of motion must be suitable for symbolic implementation, and suitable for efficient simulations. The symbolic implementation should also apply typical methods to improve the efficiency of the code by doing such things as computing common subexpressions only once and by precomputing time-invariant terms.

An issue to be addressed is how to adapt existing recursive approaches to multibody dynamics in free-fall. These formulations are satisfactory for symbolic manipulations for the systems under consideration. Unfortunately, the numerical implementations generally depend on the angular velocity of the base remaining zero. In free-fall, this is not true. In fact, the angular velocity and position of the base depends on the motion of all the joints (due to the conservation of angular momentum.) Techniques to handle these unknown quantities during the recursions have not been described in the literature and will be studied in this research. This may lead to recursive formulations for the angular momentum of multibody systems in free-fall. (Note that this is not a problem for the symbolic use of recursive formulations since the unknown values are carried along as symbolic values in any case.)

3.2.5 Design optimal controls to accomplish motions

Since the idea is to precompute motions, it makes sense to compute the best possible motions. Therefore, optimal controls approaches will be used to plan the motions. It should be noted that motions produced by this approach will be optimal in some sense but that the main goal is to plan motions that are feasible and avoid extensive cyclic motions.

There are several possible approaches to be considered. An optimal control scheme based on variational analysis and the maximum principle is a logical candidate for computing the multibody motions. Appendix B gives a sample of the type of analysis proposed. The analysis actually used must satisfy several requirements. It must be simple enough and predictable to implement via symbolic manipulation. It must produce the system of executable equations (for example, state and costate equations) which are reasonably efficient. The analysis in the appendix is given to illustrate the type of approach proposed.

Appendix C illustrates the application of the optimal control scheme from Appendix B to an example.

Other approaches were mentioned in the literature survey. The technique described by Tan and Potts in "A Discrete Path/Trajectory Planner for Robotic Arms" is intended for fixed-base arms but is adaptable to this current problem [107]. It involves constructing a discrete non-linear model of the robot dynamics[68] which can then be used to construct a large non-linear programming problem. The approach is very flexible since it allows constraints on joint positions, velocities, jerks, and actuator limitations. It can avoid obstacles and will minimize a user specifiable cost function over the path.

Luus has devised another technique based on dynamic programming which is also applicable. In the recent paper "Application of Dynamic Programming to High-Dimensional Non-Linear Optimal Control Problems," Luus used dynamic programming to optimize several non-linear problems subject to input limitations. In one example, he studied a complex system with eight non-linear ordinary differential equations and determined optimum control input histories.

3.2.6 Implement symbolic generation of optimal control scheme

Once the optimal control scheme is designed, it must be implemented in terms of symbolic manipulations. An example of the type implementation to accomplish this is given in Appendix C. In this example, the optimal controls scheme outlined in Appendix B is implemented. Sample results in terms of state and costate equations are given for example systems. Example code is also shown which has been generated from these state and costate equations.

3.2.7 Precompute motions between selected configurations

In order to prepare the system for movement between configurations in various orientations, the necessary motions must be precomputed. The optimal control scheme must be applied to produce the movement data necessary for each motion. This will be implemented in the simulation environment.

The motion simulations will involve an extensive amount of computation and may require assistance from fast mainframe computers. One advantage of using X-Windows in a networked environment it is quite possible for the simulation environment to generate C code for the motion simulation, move this to a remote computer (perhaps a super computer), compile the code on that computer, run it on that computer, and return the data to the simulation environment without user interaction.

One premise of this research is that the amount of data generated by the motion simulations will not be over-whelming. To validate that assumption, it is necessary determine how much data will be generated for various situations. Appendix D contains a derivation of the number of data points that must be stored as a function of the various parameters. The resulting equation is:

$$N_{DP} = 3N_O N_J N_D N_P^2 \quad (1)$$

- where
- N_{DP} = Total number of data points required
 - N_O = Number of relative rotations
 - N_J = Number of joints or internal DOF
 - N_D = Number of data points per variable
(Number of time steps + 1)
 - N_P = Number of poses (or configurations)

To give some feel for the amount of data indicated by this equation, consider a few examples. Using two configurations and moderately optimistic values for the parameters in Equation 1, the amount of storage required for several cases are given in Table 1 (see Appendix D for details). The first example

Number of Joints, N_J	N_{DATA} (KB)
3	45.8
6	91.6
14	213.8

Table 1: Amount of Data Necessary to Store Motions

uses $N_J = 3$ and corresponds to a relatively simple mechanism. The second example uses $N_J = 6$ and corresponds to a six degree-of-freedom robot. The last example uses $N_J = 14$ and corresponds (roughly) to a human [134].

Although this is a large amount of data, it is within reason. Storing this amount of data on hard disks is quite feasible. Storing this amount of data in ROM is possible but N_P cannot be very large.

It may be possible to reduce the amount of data to be stored by only storing the joint positions during each motion. Joint velocities and joint torques can be computed on the fly by using recursive inverse dynamics formulations.

3.2.8 Adapt compression techniques to compress motion data

For each starting configuration and final configuration there will be three degrees of freedom in orientation that will be simulated. This can be thought of as a vector from the center of a sphere to some point on its surface plus an angle about that vector. One of the issues to be examined is how fine to subdivide these angles. The grid points must be close enough together so that interpolation between nearby motions will produce nearly correct results. This will be discussed further in the next section on the motion tracking control system. Unfortunately, increasing the number of divisions will tremendously affect the amount of number crunching necessary and the quantity of resulting data.

Since the motion simulation will produce a tremendous amount of data, an important component of this work will be how to compress it into a motion database (or library). Consider the plot for one joint position (or control input) over a motion. This is a single simple plot. Now consider a set of these for one of the degrees of freedom in orientation. Each plot of the joint position can be treated like a scan line of an image so that the set of plots can be thought of as an image. There are three degrees of freedom so the resulting data can be thought of as a two dimensional array of images. Since the motion data can be thought of as images, one approach to compressing this data is to apply image compression techniques. The current state of the art in image compression for exact reproduction is roughly 3:1 for typical images. In this case, exact reproduction is not necessary; techniques exist which give nearly lossless single image compressions of roughly 10:1 to 40:1 [32, 10]. When a number of similar images are compressed, further compression is possible by exploiting the similarity between the images—resulting in compression ratios of up to roughly 100:1. With this type of compression, it is possible to compress an extensive set of data into a reasonable amount of space. It is expected that the set of pseudo-images will be relatively similar so that compression techniques will be effective.

The image compression techniques described typically depend on the image being composed of integer data with limited range, for instance, 0-255. The joint position and control input data will typically be floating point. An issue to be addressed is what level of quantization will allow acceptable reconstruction of the joint and control profiles.

There may be a relationship between the type of compression scheme implemented and how the system will be used. If the motions are needed often and quickly (as it might be for planning), then retrieval speed becomes an issue. The most effective image compression techniques depend on reconstructing the entire image at once. All that will be needed in this case is the equivalent of one scan line from several different images. Some compression techniques may be more efficient for retrieving one scan line at a time from an image (or set of images).

The computations for compression will be extensive. This is not necessarily a significant problem for an actual application since video compression hardware exists today which can do the necessary compression at video frame rates.

3.2.9 Design linearized motion tracking control scheme

The process of compression means the reconstructed joint position and control input profiles will not be exactly what they should be. Also, there will be uncertainties in the parameters of the actual system. Given the joint position and control input profiles necessary to accomplish some configuration and orientation change how can we persuade the system to actually complete the desired motion? Obviously some type of trajectory tracking controller will be necessary. There are a number of possibilities here. One is a time-varying linearized control system. Another approach is feedback linearization. In the research, various options will be examined.

The controllability of the time-varying linearized system is an important issue that will be addressed. In a sense, the linearized control system controls the motions in the small at any instant. It will not be fully controllable (in general) since it cannot command motions that violate the angular momentum constraint.

A reasonable approach (if the mechanism is suitable) is to reduce the order of the system used to determine the planned motion (for instance, by freezing some of the joints). Then, during the motion tracking phase, the linearized controller can use those joints to keep the system close to the desired motion.

3.2.10 Implement symbolic generation of linearized controller

Once the form of the time-varying linearized controller is designed, it should not be difficult to use symbolic manipulation to apply it to the equations of motion. In this way, two implementation problems can be addressed via symbolic manipulation. First, the system will generate C code to implement the linearized feedback control law. This code will not vary during the motion. Second, the system will generate C code to compute the time-varying data necessary for the linearized controller. This code will be run as necessary to update the data in the linearized feedback control code.

3.2.11 Use simulation to verify linearized controller

To test the motion data libraries and linearized controller, the system will use perform simulations. The system will use standard multibody simulation techniques with joint actuator inputs from the linearized controller and motion libraries. These simulations will test many phases of the research. They will also give a feel for what kind of accuracy and resolution is necessary in the motion database to give adequate control with the linearized controller.

3.2.12 Apply system to example multibody systems

To illustrate use of the system and to test it, it will be applied to several example multibody systems in free-fall. Useful examples include two body systems, typical space robots, and simplified human models. Although human motion in free-fall is a desirable application, it may be too ambitious for initial applications due to its large number of degrees of freedom.

3.3 Expected Results and Contributions

It should be noted that no single piece of this research is revolutionary. At most small extensions from the state of the art are proposed. What makes this research unique is the way the components are put together. Nobody has yet successfully addressed the end-to-end problem of how to control multibody systems in free-fall in real time. This will be the primary contribution of this research.

Other contributions will include:

- Extending recursive multibody formulations for numerical simulations of systems in free-fall.
- Embedding the multibody tree structure in the software objects created to model it. ×
- Construction of a flexible, powerful, and portable simulation environment which can be applied to real motion problems.
- Design and implementation of optimal control for configuration change. Using symbolic manipulation to construct the optimal controller.
- Using image compression techniques to compress motion data.
- Storing precomputed motions for complex systems for later use. ?
- Using symbolic manipulation to implement the time-varying linearized motion tracking controller.
- Use of symbolic manipulation for dynamics and controls in one integrated system.

References

- [1] J. S. Albus. "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)." *Journal of Dynamic Systems, Measurement, and Control*. Vol. 97, 1975. pp. 1975.
- [2] J. S. Albus. "Data Storage in the Cerebellar Model Articulation Controller (CMAC)." *Journal of Dynamic Systems, Measurement, and Control*. Vol. 97, 1975. pp. 228-233.
- [3] C.A. Balafoutis, P. Misrta, and R.V. Patel. "A Cartesian Tensor Approach for Fast Computation of Manipulator Dynamics." *Proceedings of the IEEE International Conference on Robotics and Automation*. April 1988. pp. 1318-1353.
- [4] R.S. Ball. *A Treatise on the Theory of Screws*. Cambridge University Press, London, 1900.
- [5] Jérôme Barraquand and Jean-Claude Latombe. "On Nonholonomic Mobile Robots and Optimal Maneuvering." *Proceedings of the IEEE International Symposium on Intelligent Control 1989*. Albany, NY, September, 1989. pp. 340-347.
- [6] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. "Time-Optimal Control of Robotic Manipulators Along Specified Paths." *International Journal of Robotics Research*. Vol. 4, 1985. pp. 3-17.
- [7] J.W. Burdick. "An Algorithm for Generation of Efficient Manipulator Dynamics." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1986, pp. 212-218.
- [8] Mark A. Bronez, Margaret M. Clarke, Alberta Quinn. "Requirements Development for a Free-Flying Robot—the "Robin"." *Proceedings of the IEEE International Conference of Robotics and Automation*. San Francisco, California, 1986.
- [9] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control*. John Wiley and Sons, New York, 1975.
- [10] Jonathan M. Cameron. *Survey of the State of the Art in Image Compression for Low Data-Rate Remote Driving*. EM 347-90-279. Jet Propulsion Laboratory, Pasadena, California, September 12, 1990.
- [11] G. Cesareo, F. Nicolo, and S. Nicosia. "DYMIR: A Code for Generating Dynamic Models of Robots." *Proceedings of the IEEE International Conference on Robotics*. Atlanta, Georgia, 1984. pp. 115-120.
- [12] S. Cetinkunt, and W. J. Book. "Symbolic Modeling of Flexible Manipulators." *Proceedings of the IEEE International Conference on Robotics and Automation*. Raleigh, North Carolina, April 1987. pp. 2074-2081.
- [13] C.H. Chen and A.C. Kak. "A Robot Vision System for recognizing 3-D Objects in Low-Order Polynomial Time." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 19, No. 6, Nov./Dec. 1989. p. 1535-1563.
- [14] John J. Craig. *Introduction to Robotics. Mechanics and Control*. Second Edition.

Addison-Wesley, 1986, 1989.

- [15] X. Cyril. J. Angeles, and A. Misra. "Efficient Inverse Dynamics of General N-Axis Robotic Manipulators." *Proceedings of the 9th Symposium on Engineering Applications in Mechanics*. May 1988, pp. 595-602.
- [16] Morris R. Direls, "Symbolic Matrix Manipulation Package for the Kinematic Analysis of Robot Manipulators." *Computers in Mechanical Engineering*, Vol. 5, No. 2, September 1986, pp. 38-46.
- [17] S. Dubowsky, M.A. Norris, and Z. Shiller. "Time-Optimal Trajectory Planning for Robotic Manipulators with Obstacle Avoidance: a CAD Approach." *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986, pp. 1906-1912.
- [18] H. Faessler. "Computer-Assisted Generation of Dynamical Equations for Multi-body Systems." *International Journal of Robotics Research*, Vol. 5, No. 3, Fall 1986, pp. 129-141.
- [19] Roy Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, Boston, 1987.
- [20] O. Fisher. *Einführung in die mechanik lebender mechanismen (Introduction to the Mechanics of Living Organisms)*. Leipzig, Germany, 1906.
- [21] Cliff Frohlich, "Do Springboard Divers Violate Angular Momentum Conservation?" *American Journal of Physics*, Vol. 47, No. 7, July 1979.
- [22] H.P. Gearing et al., "Time-Optimal Motions of Robots in Assembly Tasks." *IEEE Transactions on Automatic Control*, Vol. 31, 1986, pp. 512-518.
- [23] E.G. Gilbert and D.W. Johnson, "Distance Functions and Their Applications to Robot Path Planning in the Presence of Obstacles." *IEEE Journal of Robotics and Automation*, Vol. 1, 1985, pp. 21-30.
- [24] Arthur L. Hale and Leonard Meirovitch, "Derivation of the Equations of Motion for Complex Structures by Symbolic Manipulation." *Computers and Structures*, Vol. 9, December 1978, pp. 639-649.
- [25] Xiaogeng He and A.A. Goldenberg, "An Algorithm for Efficient Computation of Dynamics of Robotic Manipulators." *Journal of Robotic Systems*, Vol. 7, No. 5, 1990, pp. 689-702.
- [26] Wolfgang Hirschberg and Dieter Schramm, "Application of NEWEUL in Robotic Dynamics." *Journal of Symbolic Computation*, Vol. 7, 1989, pp. 199-204.
- [27] John M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 11, November 1980, pp. 730-736.
- [28] M.A. Hussain and B. Noble, "Application of Symbolic Computation to the Analysis of Mechanical Systems, Including Robot Arms." *NATO ASI Series on Computer*

- Aided Analysis and Optimization of Mechanical System Dynamics*. Iowa City, Iowa, 1984. pp. 283-304.
- [29] A. Izaguirre and R. Paul. "Automatic Generation of Dynamic Equations of the Robot Manipulators Using a LISP Program." *Proceedings of the IEEE International Conference on Robotics and Automation*. San Francisco, California, 1986. pp. 220-226.
 - [30] Paul Jacobs and John Canny. "Planning Smooth Paths for Mobile Robots." *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*. May 1989. pp. 2-7.
 - [31] Paul Jacobs and John Canny. "Robust Motion Planning for Mobile Robots." *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*. 1990. pp. 2-7.
 - [32] A. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
 - [33] Lyle M. Jenkins. "Telerobotic Work System—Space Robotics Application." *Proceedings of the IEEE International Conference of Robotics and Automation*. 1986.
 - [34] M.E. Kahn and B. Roth. "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chain." *Journal of Dynamic Systems, Measurement, and Control (Transactions ASME)*. Vol. 93, 1971, pp. 164-172.
 - [35] T.R. Kane and M.P. Scher. "A Dynamical Explanation of the Falling Cat Phenomenon." *International Journal of Solids and Structures*. Vol. 5, 1969. pp. 663-670.
 - [36] T.R. Kane and M.P. Scher. "Human Self-Rotation by Means of Limb Motions." *Journal of Biomechanics*. Vol. 3, 1970. pp. 39-49.
 - [37] T.R. Kane, M.R. Headrick, and J.D. Yatteau. "Experimental Investigation of an Astronaut Maneuvering Scheme." *Journal of Biomechanics*. Vol. 5, 1972. pp. 313-320.
 - [38] K. Kazerounian and K.C. Gupta. "Manipulator Dynamics Using the Extended Zero Reference Position Description." *IEEE Transactions of Robotics and Automation*. Vol. RA-2, 1986. pp. 221-224.
 - [39] W. Khalil, J.F. Kleinfinger, and M. Gautier. "Reducing the Computational Burden for the Dynamical Models for Robots." *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986. pp. 525-531.
 - [40] Byung Kook Kim and Kang G. Shin. "Minimum-Time Path Planning for Robot Arms and Their Dynamics." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. SMC-15, No. 2, March/April 1985. pp. 213-223.
 - [41] M. Kircański and M. Vukobratović. "Computer-Aided Generation of Manipulator Kinematic Models in Symbolic Form." *15th International Symposium on Industrial Robots*. 1985. pp. 1043-1049.
 - [42] D.M. Klimov, V.M. Rudenko, and V.F. Zhuravlev. "Application of Lie Group and

- Computer Algebra to Nonlinear Mechanics." *Proceedings of the European Conference of Computer Algebra. EUROCAL '87*. Leipzig, Germany, June 1978. Lecture Notes in Computer Science No. 378. Springer-Verlag, 1987
- [43] E.J. Kreuzer. "Dynamic Analysis of Mechanisms Using Symbolic Equation Manipulation." *Proceedings of the Fifth World Congress on Theory of Machines and Mechanisms (ASME)*. 1979. pp. 599-602.
 - [44] Edwin J. Kreuzer and Werner O. Schiehlen. "Generation of Symbolic Equations of Motion for Complex Spacecraft using Formalism NEWEUL." *Advances in the Astronautical Sciences*. Vol. 54. Part 1. 1983. pp. 21-36.
 - [45] P.S. Krishnaprasad. "Geometric Phases and Optimal Reconfiguration for Multi-body Systems." *Proceedings of the 1990 American Control Conference*. 1990. pp. 2440-2444.
 - [46] Jean-Paul Laumond. "Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints." *Proceedings of the IEEE International Conference on Intelligent Autonomous Systems*. December 1987. Amsterdam. pp. 346-354.
 - [47] Jean-Paul Laumond. "Finding Collision-Free Smooth Trajectories for a Non-Holonomic Mobile Robot." *10th International Joint Conference on Artificial Intelligence*. Milano, Italy. 1987. pp. 1120-1123.
 - [48] C.S.G. Lee and P.R. Chang. "Efficient Parallel Algorithms for Robot Forward Dynamics Computations." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 18. No. 2. March/April 1988.
 - [49] M.C. Leu and N. Hemati. "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators." *Journal of Dynamic Systems, Measurement and Control. Transactions ASME*. Vol. 108. No. 3. September 1986. pp. 172-179.
 - [50] D. A. Levinson. "Equations of Motion for Multiple Rigid-Body Systems via Symbolic Manipulation." *Journal of Spacecraft and Rockets*. Vol. 14. No. 8. August 1977. pp. 479-87.
 - [51] Z. Li and J.F. Canny. *Robot Motion Planning with Non-Holonomic Constraints*. Memo UCB/ERL M89/13. Electronics Research Laboratory. University of California. Berkeley, California. 1989.
 - [52] A. Liegeois, W. Khalil, J. Dumas, and M. Renaud. "Mathematical and Computer Models of Interconnected Mechanical Systems." *Proceedings of the Second International CISM-IFTOMM Symposium*. Warsaw. September 1976. pp. 5-17.
 - [53] Kathryn W. Lilly and David E. Orin. "Efficient $O(N)$ Computation of the Operational Space Inertia Matrix." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1990. pp. 1014-1019.
 - [54] R.E. Lindberg, R.W. Longman, M.F. Zedd. "Kinematics and Reaction Moment Compensation for a Space-Borne Elbow Manipulator." *Proceedings. 24th AIAA Aerospace Sciences Meeting*. Reno, Nevada. 1986.
 - [55] John Lloyd and Vincent Hayward. "Kinematics of Common Industrial Robots."

Robotics J. 1988, pp. 169-191.

- [56] B.A. Logan, Jr., "Space Station Remote Manipulator Requirements Definition." *23rd AIAA Aerospace Sciences Meeting*, Reno, Nevada, January, 1985.
- [57] Richard W. Longman, Robert E. Lindberg, Michael F. Zedd, "Satellite Mounted Robot Manipulators -- New Kinematics and Reaction Moment Compensation." *Proceedings, AIAA Guidance, Navigation, and Control Conference*, New York, NY, 1985.
- [58] Richard W. Longman, Robert E. Lindberg, and Michael F. Zedd, "Satellite Mounted Robot Manipulators -- New Kinematics and Reaction Moment Compensation." *International Journal of Robotics Research*, Vol. 6, No. 3, Fall 1987, pp. 87-103.
- [59] P. Lucibello, F. Nicolò and R. Pimpinelli, "Automatic Symbolic Modeling of Robots with a Deformable Link." *IFAC Theory of Robots*, 1986, pp. 131-135.
- [60] J.Y.S. Luh, M.W. Walker, and R.P.C. Paul, "On-Line Computational Scheme for Mechanical Manipulators." *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, 1980, pp. 69-76.
- [61] Rein Luus, "Application of Dynamic Programming to High-Dimensional Non-Linear Optimal Control Problems." *International Journal of Control*, Vol. 52, No. 1, 1990, pp. 239-250.
- [62] G. A. Macala, "SYMBOD: A Computer Program for the Automatic Generation of Symbolic Equations of Motion for Systems of Hinge-Connected Rigid Bodies." *AIAA Paper No. 83-0013*, Presented at the 21st AIAA Aerospace Sciences Meeting, Reno, Nevada, January 1983.
- [63] K. Magnus, *Dynamics of Multibody Systems*, Springer-Verlag, Berlin, 1978.
- [64] R.V. Mayorga and A.K.C. Wong, "A Global Approach for the Optimal Path Generation of Redundant Robot Manipulators." *Journal of Robotic Systems*, Vol. 7, No. 1, 1990, pp. 107-128.
- [65] B.C. McInnis and C.K.F. Liu, "Kinematics and Dynamics of Robotics: A Tutorial Based on Classical Concepts of Vectorial Mechanics." *IEEE Transactions of Robotics and Automation*, Vol. RA-2, 1986, pp. 181-187.
- [66] J.J. Murray and C.P. Neumann, "ARM: An Algebraic Robot Dynamic Modeling Program." *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, 1984, pp. 103-114.
- [67] Richard M. Murray and S. Shankar Sastry, "Steering Nonholonomic Systems Using Sinusoids." *Proceedings of the 20th Conference on Decision and Control*, Honolulu, Hawaii, December 1990, pp. 2097-2101.
- [68] Charles P. Neuman and Vassilios D. Tourassis, "Discrete Dynamic Robot Models." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 2, March/April 1985, pp. 193-204.

- [69] P.E. Nielan and T.R. Kane. "Symbolic Generation of Simulation/Control Routines for Multibody Systems," *Dynamics of Multibody Systems*, IUTAM/IFTOMM Symposium, Udine, Italy, 1986. pp. 153-164.
- [70] H. Nijmeijer and A.J. Van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, New York, 1990.
- [71] Yoshihiko Nakamura and Ranjan Mukherjee. "Nonholonomic Path Planning of Space Robots." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1989. pp. 1050-1055.
- [72] Yoshihiko Nakamura and Ranjan Mukherjee. "Nonholonomic Path Planning of Space Robots via Bi-directional Approach." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1990. pp. 1764-1769.
- [73] Yoshihiko Nakamura and Ranjan Mukherjee. "Bi-directional Approach for Nonholonomic Path Planning of Space Robots." *Proceedings, 5th International Symposium of Robotics Research*. August 28-31, 1989. Tokyo, Japan. pp. 101-112.
- [74] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. "Task-Priority Based Redundance Control of Robot Manipulators." *International Journal of Robotics Research*, Vol. 6, No. 2. Summer 1987. pp. 3-15.
- [75] Yoshihiko Nakamura and Hideo Hanafusa. "Optimal Redundancy Control of Robot Manipulators." *International Journal of Robotics Research*. Spring 1987. Vol. 6. No. 1. pp. 32-42.
- [76] José Lopes de Siqueira Neto, Antonio Eduardo Costa Pereira, and João Bosco da Mota Alves. "Symbolic Computation Applied to Robot Dynamic Modeling." *Proceedings of the 16th International Symposium on Industrial Robotics*. 1986. pp. 389-400.
- [77] C.P. Neumann and J. Murray. "Symbolically Efficient Formulations for Computational Robot Dynamics." *Journal of Robotic Systems*, Vol. 4, No. 4. 1987. pp. 503-526.
- [78] M. Niv and D.M. Auslander. "Optimal Control of a Robot with Obstacles." *Proceedings of the American Control Conference*. 1984. pp. 280-287.
- [79] J. M. Ortega. *Matrix Theory*. Plenum Press, New York, 1987.
- [80] E. Papadopoulos and S. Dubowsky. "On the Dynamic Singularities in the Control of Free-Floating Space Manipulators." *Dynamics and Control of Multibody/Robotic Systems with Space Applications*. Ed. S.M. Joshi, L. Silverberg, and T.E. Alberts. Presented at the Winter Annual Meeting of the American Society of Mechanical Engineers, San Francisco, California, December, 1989. pp. 45-52.
- [81] Marc H. Raibert. "Analytical Equations vs. Table Look-Up for Manipulation: A Unifying Concept." *Proceedings of the IEEE Conference on Decision and Control*. New Orleans, December 1977. pp. 576-579.
- [82] Marc H. Raibert, et al., *Dynamically Stable Legged Motion*. Technical Report CMU-RI-TR-83-1. Robotics Institute, Carnegie-Mellon University, January 1983.

- [83] Marc H. Raibert, Francis C. Wimberly. "Tabular Control of Balance in a Dynamic Legged System." *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*. 1983.
- [84] V.T. Ranjan. "Minimum Time Trajectory Planning." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1985. pp. 759-764.
- [85] R.E. Roberson and R. Schwertassek. *Dynamics of Multibody Systems*. Springer Verlag, Berlin. 1988.
- [86] G. Rodriguez, K. Kreutz, and A. Jain. "A Spatial Operator Algebra for Manipulator Modeling and Control." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1989. pp. 1374-1379.
- [87] G. Rodriguez and K. Kreutz. *Recursive Mass Matrix Factorization and Inversion: An Operator Approach to Open- and Closed-Chain Multibody Dynamics*. Jet Propulsion Laboratory Publication. 88-11. March 1988.
- [88] D.E. Rosenthal and M.A. Sherman. "High Performance Multibody Simulations via Symbolic Equation Manipulation and Kane's Method." *Journal of the Astronautical Sciences*. Vol. 34. No. 3 July-September. 1986. pp. 223-239.
- [89] G. Sahar and J.M. Hollerbach. "Planning of Minimum-Time Trajectories for Robot Arms." *International Journal of Robotics Research*. Vol. 5. 1986. pp. 90-100.
- [90] David B. Schaechter and David A. Levinson. "Interactive Computerized Symbolic Dynamics for the Dynamicist." *The Journal of the Astronautical Sciences*. Vol. 36. No. 4. October-December 1988. pp. 365-388.
- [91] V. Scheinman and B. Roth. "On the Optimal Selection and Placement of Manipulators." *Proceedings, 5th International CISM-IFTOMM Symposium*. 1984. pp. 39-46.
- [92] W.O. Schiehlen and E.J. Kreuzer. "Symbolic Computerized Derivations of Equations of Motion." *Proceedings of the IUTAM Symposium on Dynamics of Multibody Systems*. Munich. 1977. Springer-Verlag. 1978. pp. 290-305.
- [93] Ahmed A. Shabana. *Dynamics of Multibody Systems*. John Wiley and Sons, New York. 1989.
- [94] Michael Sherman. "Control System Verification Using Real Time Execution of Symbolically-Generated Multibody Equations of Motion." *Proceedings of the American Control Conference*. 1988. pp. 595-601.
- [95] M. A. Sherman. "The Practical Application of Symbolic Manipulation to Multibody Dynamics." *SDIO/NASA Workshop on Multibody Simulations*. Jet Propulsion Laboratory, Pasadena, California. September. 1987.
- [96] M. A. Sherman. "SD/EXACT and SD/FAST Symbolic Multibody Codes." *SDIO/NASA Workshop on Multibody Simulations*. Jet Propulsion Laboratory, Pasadena, California, September. 1987.
- [97] K.G. Shin and N.D. McKay. "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints." *IEEE Transactions on Automatic Control*. Vol. 30.

No. 6, June 1985, pp. 531-541.

- [98] K.G. Shin and N.D. McKay. "Selection of Near-Minimum Time Geometric Paths for Robotic Manipulators." *Proceedings of the American Control Conference*, 1985, pp. 346-355.
- [99] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [100] E.D. Sontag and H.J. Sussman. "Time-Optimal Control of Manipulator." *Proceedings of the IEEE International Conference on Robotics and Automation*, 1985, pp. 1643-1652.
- [101] *Spacecraft Servicing Demonstration Plan*. National Aeronautics and Space Administration, Marshall Space Flight Center, MCR 84-1866, NAS8-35496, July, 1984.
- [102] N. Sreenath. "Nonlinear Control of Multibody Systems in Shape Space." *IEEE International Conference on Robotics and Automation*, 1990, pp. 1776-1781.
- [103] N. Sreenath. "Phase Space Analysis of Multibody Systems Using Lie-Transforms Approach." *Proceedings of the 1990 American Control Conference*, 1990, pp. 2446-2447.
- [104] David Stoutemyer. *Computer Algebraic Manipulation for the Calculus of Variations, the Maximum Principle, and Automatic Control*. ALOHA Project Technical Report, University of Hawaii, 1979.
- [105] Suk-Hwan Suh and Kang. G. Shin. "Robot Path Planning with Distance-Safety Criterion." *Proceedings of the 20th IEEE Conference on Decision and Control*, 1987, pp. 634-641.
- [106] H.H. Tan and R.B. Potts. "Minimum Time Trajectory Planner for the Discrete Dynamic Robot Model with Dynamic Constraints." *IEEE Journal of Robotics and Automation*, Vol. 4, No. 2, 1988, pp. 174-185.
- [107] H.H. Tan and R.B. Potts. "A Discrete Path/Trajectory Planner for Robotic Arms." *Journal of the Australian Mathematical Society, Series B*, Vol. 31, 1989, pp. 1-28.
- [108] H.H. Tan and R.B. Potts. "A Discrete Trajectory Planner for Robotic Arms with Six Degrees of Freedom." *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 5, October 1989, pp. 681-690.
- [109] R.W. Toogood. *Symbolic Generation of Robot Dynamics Equations, Part I: The DYNIM/CLEAR System*. ACMIR TR-87-05, University of Alberta, 1987.
- [110] R.W. Toogood. *Symbolic Generation of Robot Dynamics Equations, Part II: Case Studies Using the DYNIM/CLEAR System*. ACMIR TR-87-06, University of Alberta, 1987.
- [111] R.W. Toogood. *Robot Direct Dynamics Algorithms Using Symbolic Generation*. ACMIR TR 88-??, University of Alberta.
- [112] R.W. Toogood. "Efficient Robot Inverse and Direct Dynamics Algorithms Using Micro-Computer Based Symbolic Generation." *Proceedings of the IEEE Conference*

- on Robotics and Automation*. 1989, pp. 1827-1832.
- [113] E. Tunstel and N. Vira. "Mechanization of Manipulator Kinematic Equations via MACSYMA." *Proceedings of the IEEE International Computers in Engineering Conference and Exhibit (ASME)*. 1989, pp. 649-655.
 - [114] L.I. Tyves and S.V. Markevich. "Robot-Motion Path-Planning Algorithm with Dynamic Properties of Actuator Models." *Soviet Machine Science (Trans. Mashinovedenie)*. No. 4, 1987, pp. 27-34.
 - [115] Anthony P. Tzes, Stephan Yurkovich, and F. Dieter Langer. "A Symbolic Manipulation Package for Modeling of Rigid or Flexible Manipulators." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1988, pp.1526-1531.
 - [116] Yoji Umetani and Kazuya Yoshida. "Continuous Path Control of Space Manipulators Mounted on OMV." *Acta Astronautica*. Vol. 15, No. 12, pp. 981-986, 1987.
 - [117] Ziaeddin Vafa. *The Kinematics, Dynamics and Control of Space Manipulators: The Virtual Manipulator Concept*. Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, November 1987.
 - [118] Z. Vafa and S. Dubowsky. "Kinematic and Dynamic Models of Manipulators for Use in Space: The concept of the Virtual Manipulator." *Proceedings of the 7th World Congress on Theory of Machines and Mechanisms*. 1987, Vol. 2, pp. 1233-1236.
 - [119] Z. Vafa and S. Dubowsky, "On the Dynamics of Manipulators in Space Using the Virtual Manipulator Approach." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1987, pp. 579-585.
 - [120] Z. Vafa. "Space Manipulator Motions with No Satellite Attitude Disturbances." *Proceedings of the IEEE International Conference on Robotics and Automation*. 1990, pp. 1770-1775
 - [121] Z. Vafa and S. Dubowsky. "The Kinematics and Dynamics of Space Manipulators: the Virtual Manipulator Approach," *International Journal of Robotics Research*. Vol. 9, No. 4, August, 1990.
 - [122] L. Vecchio, S. Nicosia, F. Nicolò, and D. Lentini. "Automation Generation of Dynamical Models of Manipulators." *Proceedings of the 10th International Symposium on Industrial Robots*. Milan, Italy, March 1980, pp. 293-301.
 - [123] L.T. Wang and B. Ravani. "Recursive Computations of Kinematic and Dynamics Equations for Mechanical Manipulators." *IEEE Journal of Robotics and Automation*. Vol. RA-1. 1985, pp. 124-131.
 - [124] M.W. Walker and D. Orin. "Efficient Dynamic Computer Simulation of Robotic Mechanisms." *ASME Journal of Dynamic Systems, Measurement and Control*. Vol. 104, 1982, pp. 205-211.
 - [125] Robert S. Welch. "The Fosbury Flop is Still a Big Hit." *Sports Illustrated*, Vol. 69, No. 11, Sept. 12, 1988, pp. 12-15.
 - [126] John Wen and Alan A. Desrochers, "An Algorithm for Obtaining Bang-Bang Con-

- trol Laws." *Journal of Dynamic Systems, Measurement, and Control*, June 1987, Vol. 109, pp. 171-175.
- [127] John Wen, "Existence of the Time Optimal Control for Robotic Manipulators." *Proceedings of the American Control Conference*, 1986, pp. 109-113.
- [128] J. Wittenburg and U. Wolz, "MESA VERDE: A Symbolic Program for Nonlinear Articulated Rigid Body Dynamics." *ASME Paper No. 85-DET-151*, 1985.
- [129] J. Wittenburg and U. Wolz, "A Computer Program for the Alpha-Numerical Generation of Robot Dynamics Equations." *Proceedings of the 5th CISM-IFTOMM Symposium on the Theory and Practice of Robots and Manipulators*, Udine, Italy, June, 1984, MIT Press, 1985.
- [130] J. Wittenburg, *Dynamics of Systems of Rigid Bodies*, B.G. Teubner, Stuttgart, 1977.
- [131] Rui Yang and P.S. Krishnaprasad, "On the Dynamics of Floating Four-Bar Linkages." *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, Florida, December 1989, pp. 1632-1637.
- [132] M. R. Yeadon, "The Simulation of Aerial Movement—I. The Determination of Orientation Angles from Film Data." *Journal of Biomechanics*, Vol. 23, No. 1, 1990, pp. 59-66.
- [133] M. R. Yeadon, "The Simulation of Aerial Movement—II. A Mathematical Inertia Model of the Human Body." *Journal of Biomechanics*, Vol. 23, No. 1, 1990, pp. 67-74.
- [134] M. R. Yeadon, "The Simulation of Aerial Movement—III. The Determination of the Angular Momentum of the Human Body." *Journal of Biomechanics*, Vol. 23, No. 1, 1990, pp. 75-83.
- [135] M. R. Yeadon, "The Simulation of Aerial Movement—IV. A Computer Simulation Model." *Journal of Biomechanics*, Vol. 23, No. 1, 1990, pp. 85-89.
- [136] S. Yin and J. Yuh, "An Efficient Algorithm for Automatic Generation of Manipulator Dynamic Equations." *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 1812-1817.
- [137] S. Yin and J. Yuh, "A User-Friendly PC Program for Symbolic Robot Dynamic Equations: ARDEG." *Proceedings of the International Computers in Engineering Conference and Exposition*, 1989, pp. 643-648.

Appendix A. How the Simulation Environment Might Be Used

The following description explains how the simulation environment might be used in a step-by-step manner. This sequence described here is not the only way the system can be used, but does illustrate the basic components of the simulation system.

1. **User constructs system in simulation environment.** The starting point of analyzing the motion of the multibody system is for the user to model the multibody system to be analyzed in the simulation environment. This could be done by direct manipulation (on the computer screen) or by reading an appropriate data file. To specify the multibody mechanism by direct manipulation, the user will select the links from a catalog of link shapes, specify (or modify) the link's geometric and inertia properties, indicate where any joints would be located, and what other links are attached to each joint. The system would then create software objects to model the links and joints. Note that this would automatically establish the connectivity (or tree structure) of the system. The software for each object would know how to construct the relevant transformations (symbolically and numerically) to determine the robot kinematics. Similarly, the object's code would also know how to add their components (symbolically and numerically) to recursive dynamic formulations.
2. **User chooses typical configurations** At this point, the user will choose *typical configurations or poses* for the multibody system. The typical poses will be selected to put the multibody system in various useful or desirable configurations. For example, if the system is a human, a typical configuration might be a straight body with arms extended. In [21], Cliff Frohlich gives nine different human body configurations that are commonly used by divers and trampolinists. Whether the body is upside down or rightside up is not important in specifying the configuration. The typical poses will probably include only configuration information (such as joint positions) and will not include joint velocities. Including initial and final velocities in the typical con-

figurations will increase computation and storage requirements by an unreasonable degree. The system will be able to plan individual motions with initial and final velocities but these will not be included as part of the precomputation part of the research. This may limit the usefulness of the precomputation and storage aspect of the research to human motion since large initial and final velocities are often part of athletic motions.

The typical configurations may also be selected to simplify the system dynamics. For instance, the pose might put the wrists into a neutral position. During the planned motion, these joints might not be used to reduce the dimensionality of the problem.

3. **System automatically constructs equations of motion.** After the description of the system is entered, the system will generate the equations of motion in symbolic form. The simulation system will be able to deal with the dynamics of the system in three ways. First, it will be able to simulate the multibody system dynamics directly using standard recursive approaches. Second, it will be able to generate the equations of motion in a symbolic form. Third, it will be able to simulate the multibody dynamics by using software code generated from the symbolic representations of the equations of motion.
4. **System automatically generates optimal controls.** Once the equations of motion are generated in symbolic form, the optimal control law for reconfiguration will be derived symbolically. This is why it is important to generate the equations of motion in symbolic form. This will also include generation of code to verify whether the motion is possible (from the analysis of nonlinear controllability and reachability analysis).
5. **System simulates optimal motions.** Once the equations of motion have been generated and the optimal control scheme has been constructed, these will be used to simulate optimal motions for orientation changes between the typical configurations. During each simulation the basic data of control inputs and states during the motion will be saved. The goal is to simulate the motions from any configuration to any other configuration in any other orientation where such motions

are possible. Hopefully, the number of typical configurations will be small. (If the number of typical configurations is large, the amount of computation and resulting data may be excessive.)

6. **System compresses the resulting simulation data into library of motion data.** The preceding step will generate a large amount of data. The resulting motion data will be compressed using image compression techniques to construct a database of motion data (or motion library).
7. **System designs a linearized controller to execute the maneuvers in the motion library.** After the system constructs the motion library, the library can be used to plan and execute motions. However, the data compression techniques will introduce some errors into the reconstructed motion profiles due to quantization and other effects. This, along with imperfect modeling, indicates a motion tracking controller will be necessary. At this point, the simulation environment will use the symbolic version of the equations of motion to symbolically generate the necessary linearized controller to allow the system to execute a retrieved motion profile. This linearized controller will be converted from symbolic form to usable software code for simulation purposes.
8. **User uses system to simulate various motions.** Once the previous steps have been completed, the simulation environment can be used to simulate motions and test the motion tracking controller. This could be the goal of the entire system. Consider how such a system could be used:
 - The user could simulate possible motions just to see what they look like and what types of control inputs are necessary.
 - The system could be used to verify the linearized controller by doing simulations with precomputed motion data and a perturbed system. The control inputs could go into a true dynamic simulation to verify the results.
 - A diving coach could use the system to construct a new dive sequence and show it to divers in a movie form. This would involve using several intermediate poses and splicing together the

necessary motions. (It might also involve generating specific new motions or new motion libraries.) The system would take the sets of joint position profiles and construct a longer sequence to show how the motion would look. Individual joint motions could be isolated to show the diver what to do and when. It is even possible that this system could be used to discover maneuvers that have never been thought of before.

- Similar techniques could be used with astronauts to train them to do combined configuration and orientation maneuvers.
- The system could be used to construct a motion library and tracking controller for an orbital servicing robot. This might involve generating new motion libraries tailored to specific tasks. The resulting data and code could be put into ROM for use on orbit.

One of the goals of this work is to reduce the amount of user interaction necessary. Ideally, steps 1 and 2 would be the only steps the user would have to supervise. In reality, some interaction will probably be necessary during some of the other steps as well.

Appendix B. Sample Optimal Control Analysis

Suppose that we have a nonlinear system such as a multibody robot in free-fall and we would like to choose a set of control inputs to move the system from one configuration to another. Start by putting the equations of motion of the system into the following form (this can be done symbolically from the equations of motion):

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (2)$$

where \mathbf{x} is a vector describing the state and velocities of the multibody system and \mathbf{u} is a vector of available control inputs (such as control torques at each joint). This form was chosen since the equations of motion for multibody systems can generally be put into this form.

The initial configuration, $\mathbf{x}(t_0) = \mathbf{x}_0$, is known and the goal is to use the control inputs to move the system into the final configuration, $\mathbf{x}(t_f) = \mathbf{x}_f$. (where t_f is unspecified). The requirement that the system achieve the desired terminal state can be formulated in the terminal constraint:

$$\Psi[\mathbf{x}(t_f), t_f] = \mathbf{x} - \mathbf{x}_f = 0 \quad (3)$$

The terminal constraint is adjoined to the terminal cost (which is zero so far) by using a vector of constant multipliers, $\boldsymbol{\nu}$:

$$\Phi[\mathbf{x}(t_f), t_f] = \boldsymbol{\nu}^T \Psi = \boldsymbol{\nu}^T (\mathbf{x} - \mathbf{x}_f) \quad (4)$$

The motion should minimize some combination of time required for the motion and control effort of the actuators during the motion. A suitable cost function is:

$$J = \Phi[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} L dt \quad (5)$$

$$\text{where: } L = a + \frac{1}{2} \mathbf{u}^T B \mathbf{u} \quad (6)$$

$$0 \leq a < 1 \quad (7)$$

$$B_{ij} = \begin{cases} b_i(1-a) > 0 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (8)$$

The constant a is the ratio between the conflicting goals of minimizing the time required for the motion and minimizing the control effort of the actuators during the motion. If $a = 0$, then time is of no concern. If $a \approx 1$ then control effort levels are of little concern. Note that B is positive definite so B^{-1} exists. Also, since B is diagonal, $B^{-1} = B^{-T}$.

Following the typical optimal controls approach, the previous constrained problem can be converted to an unconstrained optimization. This is done by constructing a modified cost functional which enforces the equations of motion by adjoining the equations of motion with lagrange multipliers, λ :

$$\bar{J} = \Phi[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} [L + \lambda^T(-\dot{\mathbf{x}} + \mathbf{f} + \mathbf{g}\mathbf{u})] dt \quad (9)$$

To simplify the problem, the Hamiltonian of the system at some instant in time is introduced:

$$H = L + \lambda^T(\mathbf{f} + \mathbf{g}\mathbf{u}) \quad (10)$$

Which means the modified cost functional is:

$$\bar{J} = \Phi[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} [H - \lambda^T \dot{\mathbf{x}}] dt \quad (11)$$

Taking the total variation of \bar{J} and integrating by parts results in:

$$\delta \bar{J} = \left[\left(\frac{\partial \Phi}{\partial \mathbf{x}} - \lambda^T \right) \delta \mathbf{x} \right]_{t=t_f} + [\lambda^T \delta \mathbf{x}]_{t=t_0} + \int_{t_0}^{t_f} \left[\left(\frac{\partial H}{\partial \mathbf{x}} + \dot{\lambda}^T \right) \delta \mathbf{x} + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} \right] dt \quad (12)$$

Note that the variation with respect to λ has been omitted since it leads back to the equations of motion. To force the variation of the modified cost functional, $\delta \bar{J}$, to vanish, we choose:

$$\dot{\lambda}^T = -\frac{\partial H}{\partial \mathbf{x}} \quad (13)$$

$$\lambda^T(t_f) = \left[\frac{\partial \Phi}{\partial \mathbf{x}} \right]_{t=t_f} \quad (14)$$

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \quad (15)$$

$$\delta \mathbf{x}|_{t=t_0} = 0 \quad (16)$$

These choices guarantee that the resulting λ and u produce a stationary value of \bar{J} . For \bar{J} to be minimized, the second gradient must be positive definite:

$$\begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} > 0 \quad (17)$$

where the subscripts represent partial derivatives. This can be implemented symbolically. If this is satisfied, then the resulting controls will be optimal. See [9, p.50], for a detailed derivation of this requirement. Note that $H_{uu} = B$ which was chosen to be positive definite.

Applying these results to the problem at hand gives:

$$\frac{\partial H}{\partial \mathbf{x}} = -\dot{\lambda}^T \implies \dot{\lambda} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \lambda \quad (18)$$

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \implies \mathbf{u}^T B + \lambda^T \mathbf{g} = 0 \quad (19)$$

Equation (18) gives the differential equations for the costate vector, λ . Solving Equation (19) for u gives the optimal control law:

$$\mathbf{u} = -B^{-1} \mathbf{g}^T \lambda \quad (20)$$

Since the terminal time, t_f , is not specified, it is a free parameter. Treating t_f as a free parameter produces a modified total variation of the cost functional, $\delta \bar{J}$. The previous analysis and choices force all the terms to vanish except for the variation due to possible changes in the final time:

$$\delta \bar{J} = \left[\frac{\partial \Phi}{\partial t} + H \right]_{t=t_f} \delta t_f \quad (21)$$

See [9, p.72], for a detailed derivation of this requirement. Since Φ is autonomous, $\frac{\partial \Phi}{\partial t} = 0$ and therefore $H_f = 0$.

The Hamiltonian, H , is constant as can be seen by taking its derivative with respect to time:

$$\frac{d}{dt} H = \frac{\partial H}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial H}{\partial \lambda} \dot{\lambda} + \frac{\partial H}{\partial \mathbf{u}} \dot{\mathbf{u}} + \frac{\partial H}{\partial t} \quad (22)$$

$$= \frac{\partial H}{\partial \mathbf{x}} \mathbf{f} + \mathbf{f}^T \left(-\frac{\partial H}{\partial \mathbf{x}} \right)^T + 0\dot{\mathbf{u}} + \frac{\partial H}{\partial t} \quad (23)$$

$$= \frac{\partial H}{\partial t} \quad (24)$$

The Hamiltonian is autonomous so $\frac{d}{dt}H = \frac{\partial H}{\partial t} = 0$. This means that H is constant on the optimal trajectory. Its constant value must be the same as its final value, $H_f = 0$. Therefore, $H = 0$ on the optimal trajectory.

The final value of the costate vector, $\boldsymbol{\lambda}$, is determined from the terminal cost:

$$\boldsymbol{\lambda}(t_f) = \boldsymbol{\lambda}_f = \left[\frac{\partial \Phi}{\partial \mathbf{x}} \right]_{t=t_f}^T \quad (25)$$

$$= \left[\frac{\partial}{\partial \mathbf{x}} (\boldsymbol{\nu}^T \boldsymbol{\Psi}) \right]_{t=t_f}^T \quad (26)$$

$$= \left[\frac{\partial}{\partial \mathbf{x}} (\boldsymbol{\nu}^T (\mathbf{x} - \mathbf{x}_f)) \right]_{t=t_f}^T \quad (27)$$

$$= \boldsymbol{\nu} \quad (28)$$

This indicates that the final value of the costates are unknown constants. In order to determine their values, consider the Hamiltonian at the final time. Substitute $\mathbf{u} = -B^{-1}\mathbf{g}^T$, $\boldsymbol{\lambda} = \boldsymbol{\nu}$, $\mathbf{f}_f = \mathbf{f}(\mathbf{x}(t_f))$, and $\mathbf{g}_f = \mathbf{g}(\mathbf{x}(t_f))$ into H :

$$H = \left[a + \frac{1}{2} \mathbf{u}^T B \mathbf{u} + \boldsymbol{\lambda}^T (\mathbf{f} + \mathbf{g} \mathbf{u}) \right]_{t=t_f} \quad (29)$$

$$= a + \frac{1}{2} (-B^{-1} \mathbf{g}_f^T \boldsymbol{\nu})^T B (-B^{-1} \mathbf{g}_f^T \boldsymbol{\nu}) + \boldsymbol{\nu}^T (\mathbf{f}_f + \mathbf{g}_f (-B^{-1} \mathbf{g}_f^T \boldsymbol{\nu})) \quad (30)$$

$$= a + \frac{1}{2} \boldsymbol{\nu}^T \mathbf{g}_f B^{-T} B B^{-1} \mathbf{g}_f^T + \boldsymbol{\nu}^T \mathbf{f}_f - \boldsymbol{\nu}^T \mathbf{g}_f B^{-1} \mathbf{g}_f^T \boldsymbol{\nu} \quad (31)$$

$$= a + \boldsymbol{\nu}^T \mathbf{f}_f - \frac{1}{2} \boldsymbol{\nu}^T (\mathbf{g}_f B^{-1} \mathbf{g}_f^T) \boldsymbol{\nu} \quad (32)$$

At the final time, H is:

$$H = a + \boldsymbol{\nu}^T \mathbf{f}_f - \frac{1}{2} \boldsymbol{\nu}^T (\mathbf{g}_f B^{-1} \mathbf{g}_f^T) \boldsymbol{\nu} = 0 \quad (33)$$

However, since $H = 0$ during the entire motion, a similar statement is true at the initial time:

$$H = a + \boldsymbol{\mu}^T \mathbf{f}_0 - \frac{1}{2} \boldsymbol{\mu}^T (\mathbf{g}_0 B^{-1} \mathbf{g}_0^T) \boldsymbol{\mu} = 0 \quad (34)$$

where $\boldsymbol{\mu} = \boldsymbol{\lambda}(t_0)$, $\mathbf{f}_0 = \mathbf{f}(\mathbf{x}(t_0))$, and $\mathbf{g}_0 = \mathbf{g}(\mathbf{x}(t_0))$. Note that $\boldsymbol{\mu}$ is not known but that \mathbf{f}_0 and \mathbf{g}_0 are both known. (Note that if this is satisfied at the initial time, the optimal control guarantees that the similar requirement will be satisfied at the final time.)

This is a quadratic form in $\boldsymbol{\mu}$. It describes a multi-dimensional surface. Mathematically, this equation can have either no solutions, a unique solution, or many solutions.

A Newton-Raphson style scheme can be used to find a solution for $\boldsymbol{\mu}$ if one exists. Consider the change in H to a small change in μ_i near a solution:

$$H_0(\mu_1, \mu_2, \dots, \mu_i + \Delta\mu_i, \mu_{i+1}, \dots, \mu_n) = H_0(\boldsymbol{\mu}) + \frac{\partial H_0}{\partial \mu_i} \Delta\mu_i + O(\mu_i^2) \quad (35)$$

If $\boldsymbol{\mu}$ is near a solution, then $H_0(\mu_1, \mu_2, \dots, \mu_i + \Delta\mu_i, \mu_{i+1}, \dots, \mu_n) \approx 0$ and the $O(\mu_i^2)$ terms are nearly zero so the resulting equation is:

$$0 \approx H_0(\boldsymbol{\mu}) + \frac{\partial H_0}{\partial \mu_i} \Delta\mu_i \quad (36)$$

This equation can be solved for $\Delta\mu_i$:

$$\Delta\mu_i = -\frac{H_0(\boldsymbol{\mu})}{\left(\frac{\partial H_0}{\partial \mu_i}\right)} \quad (37)$$

Then μ_i can be improved:

$$(\mu_i)_{new} = (\mu_i)_{old} + \Delta\mu_i \quad (38)$$

This is a scalar equation for one μ_i . The values of $\frac{\partial H_0}{\partial \mu_i}$ can be determined as follows:

$$\frac{\partial H_0(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \mathbf{f}_0^T - \boldsymbol{\mu}^T (\mathbf{g}_0 B^{-1} \mathbf{g}_0^T) \quad (39)$$

For computational efficiency, the n update equations of the form of Equation (38) can be applied in parallel.

Some experimentation may be required to choose the weighting values a and b_i to produce reasonable trajectories with reasonable joint actuator levels.

In summary, the system has a set of first-order ordinary differential equations for state and costate. It is a two point boundary problem where initial and final values exist for the state. The initial values of the costate are unknown but candidates can be found using the Newton-Raphson type approach just described. This problem can be attacked by a shooting technique. Errors at the end can be used to improve the guess of the initial costate values. The only modification needed to this is that the initial costate values must be further modified so that they satisfy the quadratic form.

Appendix C. Applied Optimal Controls Example

Appendix C.1. Description

The purpose of this example is to demonstrate the type of symbolic manipulation techniques that can be applied to generate optimal controls laws. Obviously, this requires that the equations of motion are available in symbolic form.

Appendix C.2. MACSYMA Usage Descriptions and Code

The following description of the function OPTCONT is from the file OPTCONT.USAGE:

The function OPCONT will take an array of first order ordinary differential equations with a cost functional and it will derive and return the optimal control, the costate equations, the Hamiltonian, the Hessian of the system. To use this function, the dynamic system must be put in the form of a list of first order differential equations of the form:

$$\frac{dx}{dt} = f(x,u)$$

USAGE: OPTCONT(odes,L,x_name,u_name);

INPUTS:

odes : list of first order ordinary differential equations, eg,
[dx1/dt = f1(x,u), dx2/dt = f2(x,u), ...]
describing the dynamics of the system.

L : the cost functional of the system (in terms of x and u);
[scalar function of x,u]

x_name : list of names of the states used in odes and L.
Must be in the same order and in one-to-one correspondence
with the variables in the left hand sides in the odes.

u_name : list of names of the control inputs used in f and L.

OUTPUTS:

A list composed of:

costate : List of the costate first-order ode equations in terms
of the original state variable names and new multiplier
variables, Li.

costate_names : a List of the newly introduced costate names

u_opt : List of Optimal controls

H : The Hamiltonian of the system, $H = L + LT*f$ [scalar
function]

H_hess : The hessian of H,

$$H_hess = \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix}$$

This can be used to determine if the generated control
inputs are optimal. [(n+m) x (n+m) MATRIX of scalar
functions]

NOTE: The list costate contains variables of the form L1,L2,L3...
This function uses KILL on all of these it uses, so existing
variables with names of this form will be destroyed.

By: Jonathan M. Cameron

The MACSYMA code for the function OPTCONT follows:

```
OPTCONT(odes,L,x_name,u_name) :=  
BLOCK([ n : Length(odes), /* Number of states */  
        m : Length(u_name), /* Number of controls */  
        Lambda, U_Eqns, Hx, Hu, i, j,  
        costate, u_opt, H, H_hess],
```



```

/* Construct list of lambdas */
Lambda : [],
for i:1 thru n do
    Lambda : append(Lambda, [concat('L,i)]),
Apply('KILL,Lambda),

/* Form the Hamiltonian */
H : L,
for i:1 thru n do
    H : H + Lambda[i]*RHS(Odes[i]),

/* Construct costate equations */
Hx : [],
for i:1 thru n do
    Hx : append(Hx, [diff(H,x_name[i])]),
costate : [],
for i:1 thru n do (
    Depends(Lambda[i],T),
    costate : append(costate, [diff(lambda[i],T) = -Hx[i]]),
),

/* Solve for the optimal controls */
Hu : [],
for i:1 thru m do
    Hu : append(Hu, [diff(H,u_name[i])]),
U_Eqns : [],
for i:1 thru m do
    U_Eqns : append(U_Eqns, [0 = Hu[i]]),
u_opt : Solve(U_Eqns,u_name),

/* Generate the Hessian */
H_hess : ZEROMATRIX(n+m,n+m),
/* Do the Hxx block */
for i:1 thru n do
    for j:1 thru n do
        H_hess[i,j] : diff(Hx[j], x_name[i]),
/* Do the Hxu block */
for i:1 thru n do
    for j:1 thru m do
        H_hess[i,j+n] : diff(Hu[j], x_name[i]),
/* Do the Hux block */
for i:1 thru m do
    for j:1 thru n do
        H_hess[i+n,j] : diff(Hx[j], u_name[i]),
/* H_hess = [Hxx Hxu] */
/*           [Hux Huu] */

```

```

/* Do the H_uu block */
for i:1 thru m do
  for j:1 thru m do
    H_hess[i+n,j+n] : diff(Hu[j], u_name[i]),

/* Return the results */
[costate, lambda, u_opt, H, H_hess]
)$

```

The following description of the function SIMPCONT is from the file SIMPCONT.USAGE:

SIMPCONT is a function which will take lists of state and costate first order ordinary differential equations, the optimal control, and other outputs of OPTCONT and will substitute in the optimal control in the state ODEs and then solve as many of the ODEs as possible. The resulting system is ready to simplify via boundary conditions.

USAGE:

SIMPCONT(state_odes,costate_odes,u_opt,state_names,costate_names);

INPUTS:

state_odes : List of first order ordinary differential equations
 [dx1/dt = f1(x,u), dx2/dt = f2(x,u), ...] describing
 the dynamics of the system.

costate_odes : List of first order ordinary differential equations
 derived for optimal control by OPTCONT

u_opt : list of optimal controls derived by OPTCONT

state_names : list of the names of the states

costate_names : list of the names of the costates (generated by
 OPTCONT)

OUTPUTS:

A list composed of:

`new_system` : List of the new simplified set of state and costate equations

`new_names` : List of the names of the states or costates in `new_system`

`constants` : List of the new constants generated by solutions of ODEs

By: Jonathan M. Cameron

The MACSYMA code for the function `SIMPCONT` follows:

```
SIMPCONT(state,costate,u_opt,state_name,costate_name) :=
BLOCK([ n : Length(state),
        m : Length(u_opt),
        new_system : [],
        new_name : [],
        const_num : 0,
        constants : [],
        i, ii, soln],

/* Substitute the optimal controls into the state equations */
for i:1 thru m do
  for ii:1 thru n do
    state[ii] : LHS(state[ii]) = subst(u_opt, RHS(state[ii])),

/* check each of the costate ODEs and try to solve them */
for i:1 thru n do (
  soln : ode2(costate[i],costate_name[i],t),
  if soln # 'FALSE then (
    soln : subst(concat('C,const_num),%C,soln),
    constants : append([concat('C,const_num)], constants),
    const_num : const_num + 1,

/* Do substitutions with soln to eliminate the costate */
for ii:1 thru n do (
  state[ii] : LHS(state[ii]) = subst(soln,RHS(state[ii])),
  costate[ii] : LHS(costate[ii]) = subst(soln,RHS(costate[ii]))
),
for ii:1 thru length(new_system) do
```

```

        new_system[ii] :
            LHS(new_system[ii]) = subst(soln,RHS(new_system[ii]))
    )
else (
    new_system : append(new_system, [costate[i]]),
    new_name : append(new_name, [costate_name[i]])
)
),

/* check each of the state ODEs and try to solve them */
for i:1 thru n do (
    soln : ode2(state[i],state_name[i],t),
    if soln # 'FALSE' then (
        soln : subst(concat('C,const_num),%C,soln),
        constants : append([concat('C,const_num)], constants),
        const_num : const_num + 1,

        /* Do substitutions with soln to eliminate the state */
        for ii:1 thru n do (
            state[ii] : LHS(state[ii]) = subst(soln,RHS(state[ii])),
            costate[ii] : LHS(costate[ii]) = subst(soln,RHS(costate[ii]))
        ),
        for ii:1 thru length(new_system) do
            new_system[ii] :
                LHS(new_system[ii]) = subst(soln,RHS(new_system[ii])),
            /* Add this solution to the system */
            new_system : append(new_system, [soln]),
            new_name : append(new_name, [state_name[i]])
        )
    else (
        new_system : append(new_system, [state[i]]),
        new_name : append(new_name, [state_name[i]])
    )
),

/* Return the results */
Declare(constants, constant),
[new_system, new_name, constants]
)$

```

Appendix C.3. Sample MACSYMA Session Output

The following output is from a MACSYMA session using the functions OPTCONT and SIMPCONT on a simple problem.

```
(C3) load(optcont);
```

```
Batching the file USERD1:[CAMERON.PROP]optcont.mac;60
Batchload done.
```

```
(D3)          USERD1:[CAMERON.PROP]optcont.mac;60
```

```
(C4) load(simpcont);
```

```
Batching the file USERD1:[CAMERON.PROP]simpcont.mac;33
Batchload done.
```

```
(D4)          USERD1:[CAMERON.PROP]simpcont.mac;33
```

```
(C5) kill(x,v,u);
```

```
(D5)          DONE
```

```
(C6) depends([x,v,u],t);
```

```
(D6)          [X(T), V(T), U(T)]
```

```
(C7) state_eqns : [diff(x,t)=v,diff(v,t)=u];
```

```
(D7)          
$$\left[ \frac{dX}{dT} = v, \frac{dV}{dT} = U \right]$$

```

```
(C8) state_names : ['x,'v];
```

```
(D8)          [X, V]
```

```

(C9) control_names : ['u'];

(D9) [U]

(C10) L : 0.5*u^2;
(D10) 0.5 U2

(C11) /* Find the optimal control and costate equations */
results : optcont(state_eqns,L,state_names,control_names)$

(C12) costate_eqns : results[1];
(D12) 
$$\begin{bmatrix} \frac{dL1}{dT} \\ \frac{dL2}{dT} \end{bmatrix} = \begin{bmatrix} 0 \\ -L1 \end{bmatrix}$$


(C13) costate_names : results[2];
(D13) [L1, L2]

(C14) opt_control : results[3];
(D14) [U = - L2]

(C15) H : results[4];
(D15) 
$$L1 V + 0.5 U^2 + L2 U$$


(C16) Hessian : results[5];
(D16) 
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$


```

```
(C17) /* Solve as many of the ODEs as possible */
results :
  simpcont(state_eqns,costate_eqns,opt_control,state_names,costate_names)$
DUBO:[MACSYMA_412.ODE]ode2.fas;1 being loaded.
```

```
(C18) system : results[1];
```

$$(D18) \quad [X = T \left(\frac{CO^2}{2} - C1 T + C3 \right) + C2, V = \frac{CO T}{2} - C1 T + C3]$$

```
(C19) names : results[2];
```

```
(D19) [X, V]
```

```
(C20) constants : results[3];
```

```
(D20) [C3, C2, C1, C0]
```

```
(D21) DONE
```

The next step depends on the problem to be solved. In this case, it is not hard to apply initial and final state values to resolve the resulting constants. In this example, the costate and state differential equations were solved completely. This will not happen with the type of systems to be considered in this research. In general, some differential equations will be produced. In any case, it is not difficult to take the results of SIMPCONT and use MACSYMA to convert it to C or FORTRAN code for simulation purposes. Symbolic manipulation systems such as MACSYMA and Mathematica have powerful capabilities to generate program code. A function to take the results of SIMPCONT and generate code could also perform various optimizations such as computing common terms only once.

Appendix D. Movement Library Size Requirements

In order to validate the premise that the amount of data that will be saved is not too excessive, an estimate is presented in this appendix.

The motions will move the system from one combination of orientation and configuration to another combination of orientation and configuration. Suppose the goal is to move from one typical configuration to another typical configuration. Since the system is in free-fall, the final configuration has three degrees of attitude freedom with respect to the starting configuration. Think of this as the points on a unit sphere and another degree of freedom about a line from the center of the unit sphere to the points on the surface of the sphere. In order to tessellate the unit sphere, a procedure based on constructing geodesics from icosahedrons can be used [13]. The degree of the tessellation is Q . To tessellate the unit sphere so that there is an angle of approximately α between vertices, and:

$$Q = \text{int}[\arctan(2)/\alpha] \quad (40)$$

$$\begin{aligned} \text{where } Q &= \text{degree of tessellation} & (41) \\ \alpha &= \text{approximate angle between vertices} \end{aligned}$$

Therefore, the total number relative orientations that must be considered for moving from one configuration to another is:

$$N_O = \text{Number of relative orientations} \quad (42)$$

$$= \frac{2\pi}{\alpha} [10Q^2 + 2] \quad (43)$$

For each orientation, the joint positions, velocities, torques must be saved over the motion. So the number of data points for one motion is:

$$N_M = \text{Number of data points per motion} \quad (44)$$

$$= 3N_J N_D \quad (45)$$

$$\begin{aligned} \text{where } N_J &= \text{Number of joints} & (46) \\ N_D &= \text{Number of data points per variable} \end{aligned}$$

Multiplying N_O and N_M gives the number of data points necessary to store the motions from one configuration to another.

$$N_{MP} = \text{Number of data points to move from} \quad (47)$$

$$\text{one configuration to another} \quad (48)$$

$$= N_O N_M \quad (49)$$

$$= 3 N_O N_J N_D \quad (50)$$

Now assume there are N_P typical configurations. The total number data points to for motions from any configuration to any other is:

$$N_{DP} = \text{Total number of data points} \quad (51)$$

$$= N_{MP} \left[2 \binom{N_P}{2} + N_P \right] \quad (52)$$

$$= 3 N_O N_J N_D [N_P(N_P - 1) + N_P] \quad (53)$$

$$= 3 N_O N_J N_D N_P^2 \quad (54)$$

where the term $\binom{N_P}{2}$ in Equation (52) gives the number of pairs of configurations. The factor of 2 is necessary because the motions for each pair of configurations could be considered in either direction. The next term, N_P , accounts for motions from one configuration back to the same configuration in a different orientation.

Given a data compression ratio of C_R to 1 and assuming that the floating point value for each data point can be quantized to 8 bits (or a byte), the amount of data (in KB or kilobytes) is:

$$N_{DATA} = \text{Total amount of data in K Bytes} \quad (55)$$

$$= \frac{N_{DP}}{1024 C_R} \text{ KB} \quad (56)$$

To give some feel for the amount of data indicated by these equations, consider a few examples. For 10° orientation resolution, $Q \approx 6$ and $N_O = 13,032$. For each example $N_D = 10$, $N_P = 2$, and $C_R = 100$. The results are given in Table 1 on page 17.