JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 18, No. 6, November–December 1995

# Experiments with Model-Simplified Computed-Torque Manipulator Controllers for Free-Flying Robots

R. Koningstein*
*CriSys, Ltd., Newmarket, Ontario L3Y 7V1, Canada*
and
R. H. Cannon Jr.†
*Stanford University, Stanford, California 94305*

**A simplified computed-torque controller is presented for two common tasks performed by a two-armed free-flying robot: manipulator inertial endpoint control and cooperative-arm object manipulation. A computed-torque controller is formulated in such a manner that the differences between fixed-base and free-flying controller formulations are explicit and well defined. An analysis of these differences shows that the terms due to robot body accelerations are computationally expensive to include in the controller model, whereas those due to robot body velocities are both computationally inexpensive and vital to proper operation of the controller. For the two robot tasks under study, numerical simulations predict a wide range of robot/payload mass distributions for which robot body accelerations can be neglected in the manipulator controller model with insignificant performance degradation in both open- and closed-loop control. The error introduced is less than that typically due to uncertainties in mass distribution (1%). The performance of two simplified computed-torque controllers are compared to their full-order counterparts using an experimental laboratory robot and found to be indistinguishable from one another. An estimation is made of the significant computational savings possible with this simplification.**

## Introduction

POSITION-DERIVATIVE (PD) joint angle control has been known to be inadequate to the task of inertial-space endpoint control of free-flying robots. In particular, with a free-flying robot, manipulator angles alone do not determine endpoint position or grasped-object orientation.

Free-flying robots possess more degrees of freedom than their fixed-base counterparts. This has typically resulted in the inclusion of robot body accelerations and velocities in controller dynamic models. Previously published experimental work in endpoint control from free-flying robots by Alexander,[1] Umetani and Yoshida,[2] and Carignan,[3] and theoretical work by Vafa and Dubowski,[4] has concentrated on formulating new controllers to model the effects of the free-flying robot body.

Koningstein et al.[5] presented a new formulation to augment the manipulator Jacobian in order to control free-flying and/or cooperating-arm systems. Full-order controllers for manipulator systems on a free-flying robot can attain precise, high-preformance control but at a computational cost that is significantly higher than that of fixed-base manipulator controllers.

Reduced-order controllers for free-flying systems can offer control capabilities at lower cost. Masutani et al.[6] investigated (in simulation) the performance of simplified endpoint controllers on free-flying manipulators using Jacobian transpose control with Umetani and Yoshida's generalized Jacobian technique.[2] The Jacobian transpose method, however, does not model the dynamics of the system, and precise, high-performance control was not attainable. Van Woerkom and Guelman[7] used a generalized inverse to solve the Jacobian equation but found that the resultant system accelerations were inconsistent with the system dynamics. They concluded that this was not a desirable method with which to implement reduced-order control. It is evident that reduced-order controllers for free-flying robots that do not have internal dynamic and kinematic models that are largely consistent with the system being controlled will not yield precise, high-performance control.

Papadopolous and Dubowski[8] suggested that "nearly any control algorithm that can be used for fixed-base manipulators can be also employed in the control of free-flying systems," based on certain similarities between the kinematic and dynamic equations. We believe that if the controller equations are expressed in such a manner that the differences between fixed-base and free-flying systems are explicit, then an analysis of these differences can yield insight into which parts contribute to precise, high-performance control and at what computational cost.

The system that will be studied in this paper is a two-armed free-flying robot that can use its arms to grasp an object. Two common control tasks for this robot are arm endpoint control, and cooperative-arm object manipulation. Computed torque (CT) manipulator controllers will be formulated in such a manner that the differences between the free-flying and fixed-base equivalent controller are explicitly visible. The differences fall into two categories, those due to robot body accelerations and those due to robot body velocities.

Whereas there are other control objectives and system configurations in which manipulator-induced robot body accelerations can play a significant role, for these two tasks, numerical simulations predict a wide range of robot/payload mass distributions under which robot body accelerations can be neglected in the controller model with insignificant performance degradation. This theoretical study predicts that the effects on the controller performance of neglecting base accelerations can be less than that due to uncertainties in mass distribution and friction effects. This prediction is experimentally confirmed for both independent arm endpoint control and cooperative-arm object manipulation.

By neglecting robot body accelerations in the model, a potential saving of an order of magnitude in computation in the manipulator controller can be realized. As a consequence, however, the Jacobian augmentation terms that could implement integrated momentum control or explicit body control are lost. This control will then need to be external to the manipulator controller. These results have also been published in a thesis.[9]

## Notation

The notation employed throughout this paper is intended to be as consistent as possible with that developed by Kane and Levinson.[10]

Received Sept. 2, 1992; revision received June 15, 1995; accepted for publication July 17, 1995. 

*Vice President, 1235 Gorham Street, Unit 1. Member AIAA.
†Charles Lee Powell Professor of Aeronautics and Astronautics, Department of Aeronautics and Astronautics, Room 250, Durand Building. Member AIAA.

The basic components of the notation of Ref. 10 are the generalized coordinates $q_r$ for expressing position and orientation and the generalized speeds $u_r$ for expressing motion. Terms are introduced for partial linear momentum $\boldsymbol{L}_r^j$ and angular momentum $\boldsymbol{H}_r^{j/S^*}$ that are similar to partial velocities $\boldsymbol{v}_r^i$ (see also Ref. 5). All velocities $\boldsymbol{v}^i$, momenta $\boldsymbol{L}$ and $\boldsymbol{H}$, and their derivatives $\boldsymbol{a}^i$, $\dot{\boldsymbol{L}}$, and $\dot{\boldsymbol{H}}$ are expressed with respect to a Newtonian (inertial) reference frame. Jacobian matrixes $\boldsymbol{J}$ and augmented Jacobian matrixes $\mathcal{J}$ are expressed as constructions of partial velocities and partial momenta.

## Two-Link Arm Endpoint Control

This first example presents an endpoint CT control system for a two-link arm in order to familiarize the reader with the notation used, prior to analysis with manipulators mounted on a free-flying robot. This manipulator system has two degrees of freedom. The control objective is the acceleration $\boldsymbol{a}^p$ of the manipulator's endpoint $p$. The Jacobian matrix for the endpoint velocity is

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{v}_1^p & \boldsymbol{v}_2^p \end{bmatrix}_{2\times 2} \tag{1}$$

where the columns are the partial velocity measure numbers of the endpoint $p$, expressed in an inertial space. The velocity of the endpoint can be expressed as

$$\boldsymbol{v}^p = \boldsymbol{J}u \tag{2}$$

The endpoint acceleration is

$$[\boldsymbol{a}^p]_{2\times 1} = \boldsymbol{J}\dot{u} + \dot{\boldsymbol{J}}u \tag{3}$$

The system accelerations are

$$\dot{u} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix}_{2\times 1} \tag{4}$$

The generalized accelerations for the manipulator are determined with the following Jacobian equation:

$$\dot{u}_{1,\ldots,2} \underset{4,3,1}{=} (\boldsymbol{J})^{-1}\left(-\dot{\boldsymbol{J}}^* u + \boldsymbol{a}_{\text{des}}^p\right) \tag{5}$$

The inverse dynamics process can use these system generalized accelerations to determine actuator torques. A feedback control law based on endpoint position (and velocity) errors can be used to provide the commanded acceleration and close the loop.

## Free-Flying Robot: Endpoint Control

A free-flying robot has more degrees of freedom than its fixed-base counterpart. By augmenting the Jacobian, it is possible to use the same technique as for fixed-base manipulators to solve for the complete set of base and manipulator accelerations and, ultimately, the joint torques and base forces via inverse dynamics.

This example presents such a control system, a two-link arm mounted on a two-dimensional free-flying robot. This system has five degrees of freedom: three of the base and two of the manipulator. In this case, full-model control objectives include the system's linear momentum rate $\dot{\boldsymbol{L}}^S$ and angular momentum rate $\dot{\boldsymbol{H}}^{S/S^*}$ about the system's mass center and the acceleration $\boldsymbol{a}^p$ of the manipulator endpoint $p$. The augmented Jacobian matrix to solve for the five-system accelerations exactly is

$$\mathcal{J} = \begin{bmatrix} \boldsymbol{L}_1^S & \boldsymbol{L}_2^S & \cdots & \boldsymbol{L}_5^S \\ \boldsymbol{H}_1^{S/S^*} & \boldsymbol{H}_2^{S/S^*} & \cdots & \boldsymbol{H}_5^{S/S^*} \\ \boldsymbol{v}_1^p & \boldsymbol{v}_2^p & \cdots & \boldsymbol{v}_5^p \end{bmatrix}_{5\times 5} \tag{6}$$

so that robot linear and angular momentum rates and endpoint acceleration are

$$\begin{bmatrix} \dot{\boldsymbol{L}}^S \\ \dot{\boldsymbol{H}}^{S/S^*} \\ \boldsymbol{a}^p \end{bmatrix}_{5\times 1} = \mathcal{J}\dot{u} + \dot{\mathcal{J}}u \tag{7}$$

The generalized accelerations for the robot body and manipulator are determined with the following Jacobian equation:

$$\dot{u}_{1,\ldots,5} \underset{7,6}{=} (\mathcal{J})^{-1}\left(-\dot{\mathcal{J}}^* u + \boldsymbol{a}_{\text{des}}^p\right) \tag{8}$$

As for the fixed-base robot, once system accelerations are solved for, inverse dynamics can be used to compute the motor torques and base forces (if any) to accomplish the control objectives. Feedback control can be achieved with an appropriate control law.

Note that if no external forces or torques are to be applied to the robot via thrusters, etc., then the momentum change objectives must be zero. This is equivalent to embedding momentum-conserving constraints in the controller.

Jacobian augmentation allows additional controls to be specified directly to the CT controller (i.e., momentum) and is readily adaptable to cooperative-arm manipulation by expressing closed kinematic chains as velocity constraints.[5]

## Model Simplification

The full-order CT controller model just developed for a two-dimensional free-flying robot will be simplified by assuming zero robot body accelerations. Including these accelerations in a CT model is computationally expensive, since a larger Jacobian matrix needs to be inverted. Constant body acceleration, such as that due to rocket (thruster) activity, can be easily included in the model and is discussed separately in the following section.

Free-flying robot body accelerations occur in response to manipulator activity and external forces. If the robot body's mass and inertia are very large with respect to those of the manipulator(s) and its payload(s) then they will be small compared to manipulator endpoint accelerations.

In this simplified model, a reduced Jacobian equation is used to solve for a subset of the system generalized accelerations: the manipulator generalized accelerations. The simplified CT controller model presented here neglects robot body accelerations (they are assumed to be zero) but includes the robot body angular velocity. The effect of this is to remove the rows of the augmented Jacobian equation dealing with body acceleration control (or system momentum control, depending on what the control objectives may have been[5]) and remove the columns of the augmented Jacobian matrix that depended on the body accelerations. The resulting reduced Jacobian matrix is the (original) fixed-base manipulator Jacobian, however, the nonlinear terms when solving for system accelerations are not the same. The example illustrates this.

The approximate system accelerations, assuming zero base accelerations, are

$$\dot{u}_{\text{approx}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{u}_4 \\ \dot{u}_5 \end{bmatrix}_{5\times 1} \tag{9}$$

If this simplification is applied, the reduced Jacobian matrix is

$$\boldsymbol{J} \underset{9,6,1}{=} \begin{bmatrix} \boldsymbol{v}_4^p & \boldsymbol{v}_5^p \end{bmatrix}_{2\times 2} \tag{10}$$

and the row-reduced augmented Jacobian derivative matrix $\dot{\mathcal{J}}^*$ is

$$\dot{\mathcal{J}}^* \underset{9,6}{=} \begin{bmatrix} \dot{\boldsymbol{v}}_1^p & \dot{\boldsymbol{v}}_2^p & \dot{\boldsymbol{v}}_3^p & \dot{\boldsymbol{v}}_4^p & \dot{\boldsymbol{v}}_5^p \end{bmatrix}_{2\times 5} \tag{11}$$

The generalized accelerations for the manipulator are determined with the following simplified Jacobian equation, that uses the full system state to take into account robot body angular velocity:

$$\dot{u}_{4,5} \underset{12,10,9}{=} (\boldsymbol{J})^{-1}\left(-\dot{\mathcal{J}}^* u + \boldsymbol{a}_{\text{des}}^p\right) \tag{12}$$

These approximate manipulator generalized accelerations form the basis of the approximate system generalized accelerations of Eq. (9).

The inverse dynamics process then uses these approximate system generalized accelerations with the full system state to determine

manipulator actuator torques. Residual forces and torques calculated for the robot body as a consequence of this process, although used in full-model control, are to be ignored.

## Momentum and/or Base Control

By using a simplified CT controller such as this, the ability to control momentum, such as specified in the augmented Jacobian in Eq. (7), is lost; it will need to be done by an external controller. Momentum control (for stationkeeping and maneuvering) could potentially produce large robot body accelerations $\dot{u}_{1,\ldots,3}$, but these can be treated as invariant to manipulator activity, so that the Jacobian equation can include them as constants. This is accomplished by partitioning off a piece of the augmented Jacobian $\mathcal{J}_{1,\ldots,3,4,\ldots,5}$. The manipulator accelerations, instead of 12 [Eq. (12)], are then

$$\dot{u}_{4,5} \underset{6,10}{=} (\boldsymbol{J})^{-1}\left(-\dot{\mathcal{J}}^{*}u + \boldsymbol{a}^{p}_{\text{des}} - \mathcal{J}_{4,\ldots,5,1,\ldots,3}\dot{u}_{1,\ldots,3}\right) \tag{13}$$

## Effect of Model Simplification on Open-Loop Control

A measure of the accuracy of a computed-torque controller is how closely real responses (i.e., endpoint accelerations) conform to the objectives specified in an open-loop manner. Of course, friction, actuator deficiencies, and physical and mass parameter inaccuracies detract from the ideal. In the case of manipulator endpoint acceleration control, ideally the actual endpoint acceleration is that which is desired:

$$\boldsymbol{a}^{\text{ep}} = \boldsymbol{a}^{\text{des}}$$

Whereas it is difficult to measure the real accelerations precisely, and we will not do so, the effect of the controller model simplification on open-loop accelerations will prove instructive in determining the effect on closed-loop controller performance.

For the model simplification under study, we can use our simulation tools to construct a simplified controller that controls a simulated free-flying robot and observe the results numerically, or we can use our reduced Jacobian and nonlinear matrices and work out the resulting system behavior. It turns out that the manipulator endpoint accelerations in inertial space $\boldsymbol{a}^{\text{ep}}$ are the sum of a linear function of the desired accelerations $\boldsymbol{a}^{\text{des}}$ and unmodeled nonlinear components $f_{\text{NL}}$ (which are not functions of $\boldsymbol{a}^{\text{des}}$),

$$\begin{aligned}\boldsymbol{a}^{\text{ep}} &= \begin{bmatrix}\mathbf{T}_{xx} & \mathbf{T}_{xy}\\ \mathbf{T}_{yx} & \mathbf{T}_{yy}\end{bmatrix}\boldsymbol{a}^{\text{des}} + f_{\text{NL}}\\ &= \mathbf{T}\boldsymbol{a}^{\text{des}} + f_{\text{NL}}\end{aligned} \tag{14}$$

Ideally, the matrix $\mathbf{T}$ would be unity, and the nonlinear terms $f_{\text{NL}}$ would be zero; this will typically not be true for systems whose mass properties are imperfectly known.

For the case of manipulators attached to free-flying robots, components of $f_{\text{NL}}$ are zero if the robot body's angular velocity is accounted for, since the nonlinear terms are functions of only the state (joint angles, angular rates) and are independent of the system accelerations $\dot{u}_{1,\ldots,n}$. For this study, we will include the robot body's angular velocity in the controller model since it is computationally inexpensive to do so (discussed later), and it ensures proper nonlinear compensation in the controller such that Eq. (14) becomes

$$\boldsymbol{a}^{\text{ep}} = \mathbf{T}\boldsymbol{a}^{\text{des}} \tag{15}$$

The elements of the matrix $\mathbf{T}$ are functions of the mass distribution in the robot and payload. An endpoint controller or a cooperative-arm object controller is supposed to linearize and decouple the dynamical system. Therefore, if the simplified controller is to be useful, then it should have little effect on this linearization (i.e., the values of $\mathbf{T}$ should vary little with mass distribution) and there should be little cross coupling. In other words, the matrix $\mathbf{T}$ must be close to the identity matrix for a wide range of mass distributions.

In the following simulation studies, an automated CT control and dynamics software package[9] is used to evaluate the terms of $\mathbf{T}$ over wide variations of robot body mass and robot payload mass. Data plots will be presented of the members of the transformation matrix $\mathbf{T}$ that vary with mass distribution and configuration of the robot. Plots are presented for the manipulator(s) with 45-deg shoulder angle and 90-deg elbow angle and reference robot parameters as in Table 1.

**Table 1 Robot mass characteristics**

| | |
|---|---|
| Base mass, kg | 50.0 |
| Base inertia, kg-m$^2$ | 3.2 |
| Left arm mounting on base, mm | (148,148) |
| Left upper arm center mass, mm | (59, 2) |
| Left upper arm mass, kg | 1.92 |
| Left upper arm inertia, kg-m$^2$ | 0.02 |
| Left lower arm mounting on LUA, mm | (305, 0) |
| Left lower arm center of mass, mm | (200, 0) |
| Left lower arm mass, kg | 0.34 |
| Left lower arm inertia, kg-m$^2$ | 0.012 |
| Left endpoint mounting on LLA, mm | (295, 0) |
| Right arm mounting on base, mm | (148, −148) |
| Right upper arm center of mass, mm | (59, −2) |
| Right upper arm mass, kg | 1.92 |
| Right upper arm inertia, kg-m$^2$ | 0.02 |
| Right lower arm mounting on RUA, mm | (305, 0) |
| Right lower arm center of mass, mm | (200, 0) |
| Right lower arm mass, kg | 0.34 |
| Right lower arm inertia, kg-m$^2$ | 0.012 |
| Right endpoint mounting on RLA, mm | (295, 0) |

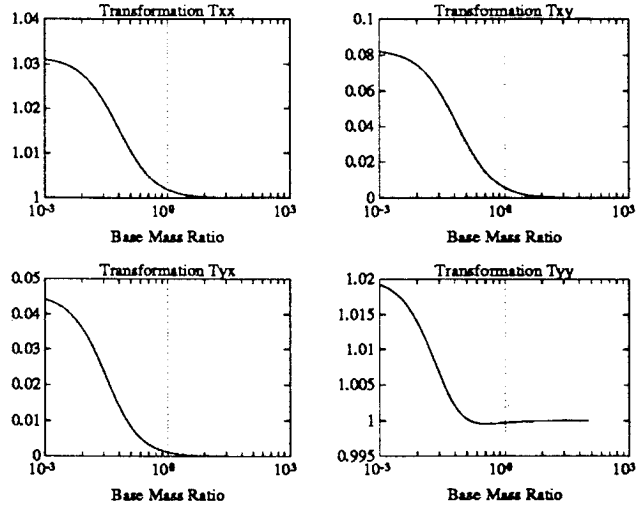


**Fig. 1 Matrix T as a function of robot body mass.**

### Effects of Variations in Robot Body Mass

In this simulation study, robot body (base) mass parameters are varied with respect to those of the experimental robot by a factor that varies between $10^{-3}$ and $10^{3}$, representing a variation from a robot body that is extremely massive and with a large moment of inertia, to that where the robot body has practically no mass or moment of inertia.

A numerical simulation that predicts elements of the transformation matrix $\mathbf{T}$ over a wide range of base mass (and inertia) values is shown in Fig. 1. The matrix $\mathbf{T}$ deviates from unity by no more than 3% with very little cross coupling for a very wide range of base masses, making this simplified controller an excellent candidate. The case of nominal mass distribution of the experimental robot, indicated by the vertical lines, yields a matrix $\mathbf{T}$ of

$$\mathbf{T} = \begin{bmatrix}1.004 & 0.008\\ 0.003 & 1.000\end{bmatrix} \tag{16}$$

### Effects of Variations in Payload Mass

In this simulation study, payload mass is varied with respect to that of the nominal payload by a factor that varies between $10^{-3}$ and $10^{3}$, representing a variation from an extremely massive payload to that where the payload has practically no mass at all.

The elements of the transformation matrix $\mathbf{T}$ over a wide range of payload mass values is shown in Fig. 2. The nominal mass

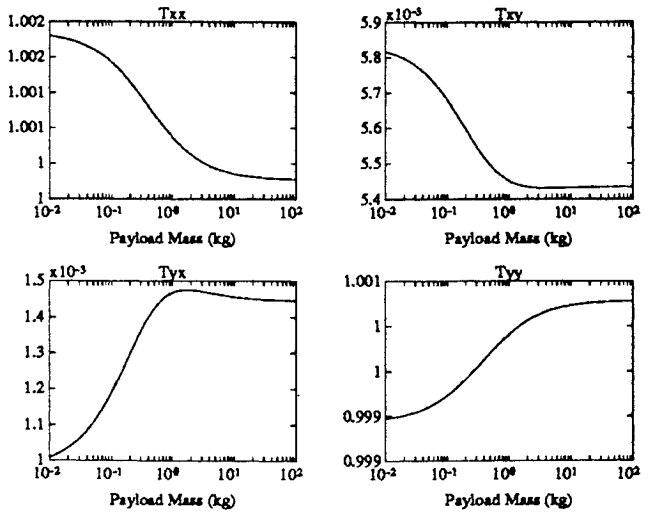


Fig. 2 Matrix T as a function of payload mass.

distribution of the experimental robot with payload is at the $10^0$ mark. The matrix **T** deviates from unity by less than a percent with very little cross coupling, making it an excellent candidate. Interestingly enough, as the mass of the payload increases, the errors due to neglecting robot body accelerations decrease; this is because the manipulator dynamics become less significant than the interaction of two effectively point masses—the base and the payload.

These numerical predictions are very similar when the robot is used to perform cooperative-arm object manipulation. For example, in the case where a payload of 6.9 kg is cooperatively manipulated, a typical matrix **T** is

$$\mathbf{T} = \begin{bmatrix} 1.0005 & 0.0054 \\ 0.0015 & 1.0007 \end{bmatrix} \tag{17}$$

## Effect of Simplification on Closed-Loop Control

The effects on closed-loop control will now be examined, given that the simplification's effect on open-loop control has been characterized. Whereas an ideal CT controller linearizes and decouples the system, the effect of the terms of the matrix **T** are to reintroduce cross coupling and change the loop gain of the system, as shown in Fig. 3.

The effect of on the gain by diagonal terms $K_g$ is to change our PD controller's closed-loop bandwidth $\omega$ and damping $\zeta$ by small amounts,

$$\omega_{\text{new}} = \sqrt{K_g} \times \omega \tag{18}$$

$$\zeta_{\text{new}} = \sqrt{K_g} \times \zeta \tag{19}$$

Endpoint acceleration cross coupling due to nonzero nondiagonal terms $K_x$ would be

$$e = K_x \times a_{\text{cross}} \tag{20}$$

Based on the results obtained for the matrix **T** in the preceding simulations, in all cases we can expect only small changes to the bandwidth and a small amount of cross coupling. In the following experiments the performance of full-order vs simplified closed-loop controllers will be examined.

### Experimental Free-Flying Robot

Endpoint control experiments are performed on a laboratory robot (Fig. 4) that has two arms with which to manipulate payloads. The robot's body is air-bearing supported over a flat, level table to achieve low-drag and zero-$g$ characteristics in two dimensions.

The robot body's mass and inertia are considerably larger than that of the manipulator arm segments, much as NASA's proposed orbital maneuvering vehicle's would be. The mass distribution of the experimental robot is as described in Table 1.

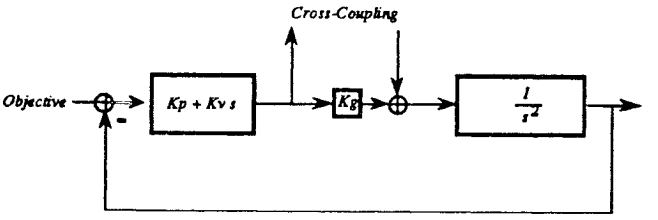


Fig. 3 Compromised closed-loop control system.

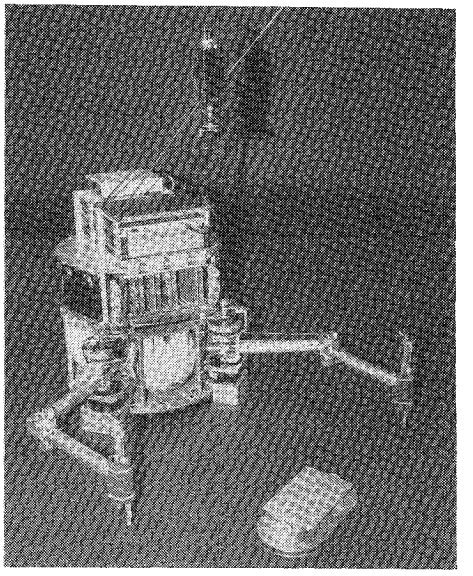

Fig. 4 Experimental laboratory robot.

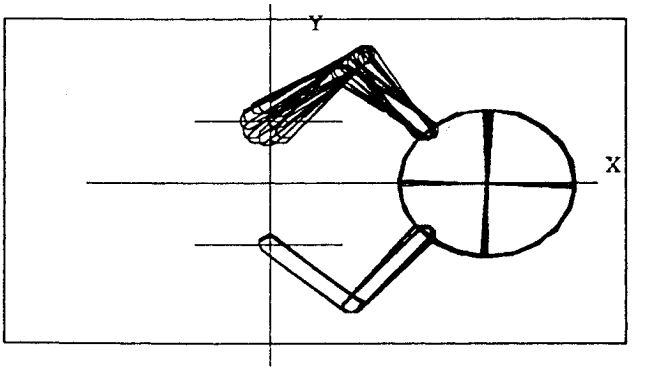


Fig. 5 One arm endpoint is moved; the other is held stationary.

The manipulator arms are actuated via limited-angle torque motors. Arm joint angles are sensed using rotational variable differential transformers. Robot body, arm endpoint, and grasped-object positions are sensed using an offboard charge-coupled device (CCD) camera. Offboard software converts sensed infrared light-emitting diode (LED) positions into point positions and into object positions and orientations as appropriate. This endpoint position (vision) sensing system has the ability to resolve to 4 mm in one axis and 10 mm in the other.

Onboard power and computers minimize the connection to the outside world that is limited to a pair of fiber-optic cables. Data collection is relayed from the robot's computer to the laboratory's computer for subsequent analysis.

### Manipulator Endpoint Control Experiment

In the first experiment, the two manipulator endpoints are position controlled as shown by a simulation in Fig. 5. One endpoint is controlled to an inertially fixed location, the other to follow a two-dimensional circular trajectory in inertial space. The gains of the PD endpoint feedback controller are $K_p = 100$ and $K_v = 22$. Control is performed using a 60-Hz sampling rate.

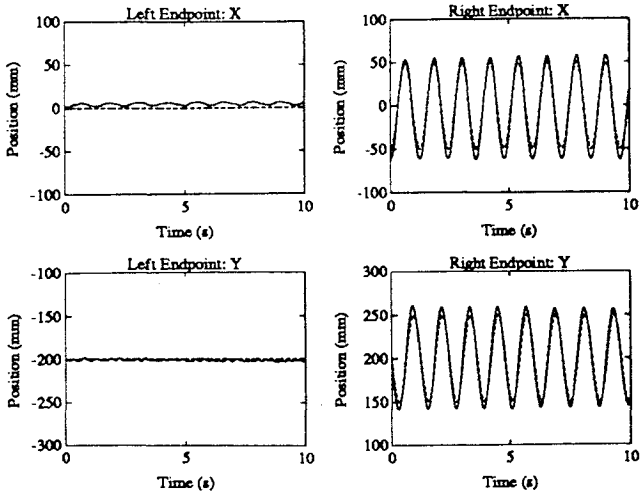


Fig. 6 Simplified CT two-arm endpoint controller performance.

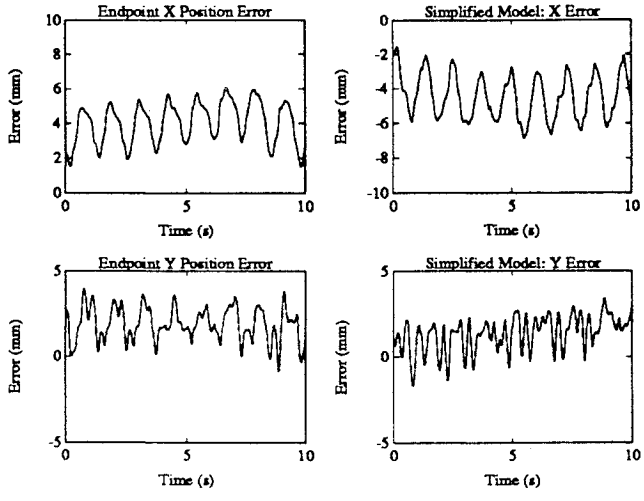


Fig. 7 Endpoint position control error comparison.

The cross coupling expected from the simplified controller model, as shown in Eq. (16) is approximately 0.008, whereas the loop gain is expected to change by 1.004, an amount equivalent to an error in the mass and inertia estimates of the robot of roughly the same magnitude: 0.4%. Based on these small deviations from the ideal, we expect to see no difference when using the simplified controller instead of the full-model controller.

Figure 6 shows experimental data obtained for the simplified CT endpoint controller. Dotted lines show commanded values, whereas solid lines show actual data. The simplified controller performance was almost identical to that of the full-order controller, as the side-by-side comparison of right endpoint position errors presented in Fig. 7 show. For both full-order and simplified controllers, errors are about 5 mm peak to peak and are due to spring forces in the joints due to wiring and tubing and due to sensor limitations. Note that due to the free-flying nature of the robot, it is difficult to exactly replicate experimental runs.

### Cooperative-Arm Object Control

In the second experiment, the position of an object grasped by both manipulator arms is controlled as shown in a simulation in Fig. 8. The object's orientation is controlled to a constant angle, and its centerpoint is controlled to follow a circular trajectory in two-dimensional inertial space. PD control gains were the same as those used for single-arm control.

The cross coupling expected from the simplified controller model as shown in Eq. (17) is approximately 0.5%, whereas the loop gain is expected to change by 0.7%. Once again, these errors are small compared to friction effects and mass parameter uncertainties.

Figure 9 shows the experimental data obtained for the simplified

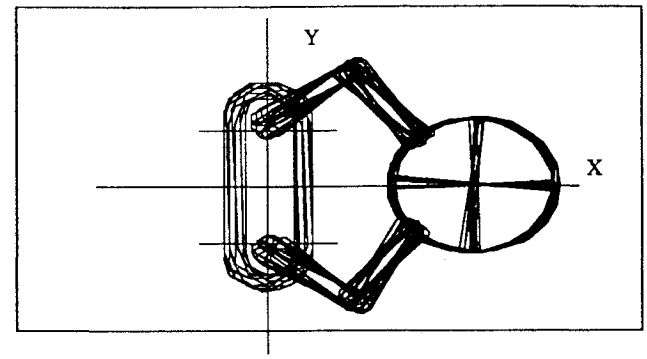


Fig. 8 Object is moved in a circle.

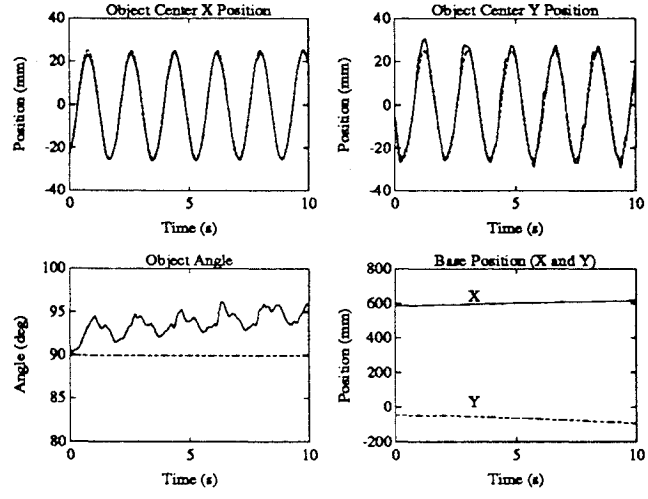


Fig. 9 Object controller performance using the simplified CT controller.

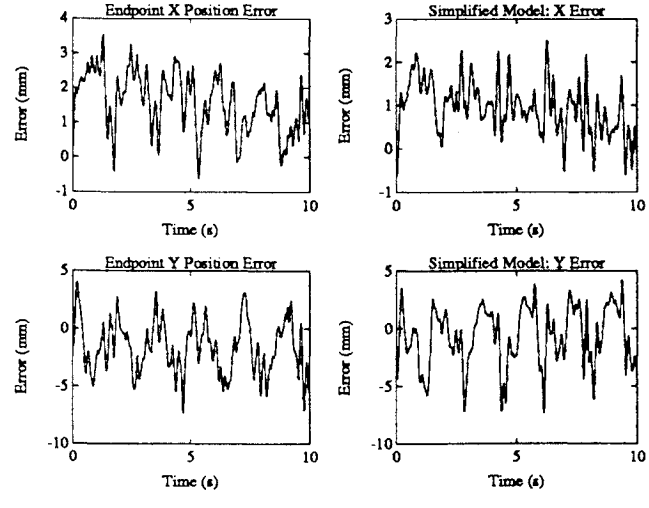


Fig. 10 Object position control error comparison.

object controller. Dotted lines show commanded values, whereas solid lines show actual data, except for the base $(x, y)$ position plot that shows real robot base position only. This controller performed well, given sensor limitations. The object's orientation could be better controlled with the use of a better angular-rate sensor in the robot body; the existing one is somewhat noisy. Furthermore, the object's orientation and center position are inferred from the grasp points and so mechanical play causes slight jumps in these plotted (estimated) positions.

The simplified object controller performance was almost identical to that of the full-order object controller, as the side-by-side comparison of object position error in Fig. 10 shows. In both cases the errors visible are about 4 mm peak to peak for $x$ and 10 mm

peak to peak for $y$, which is an artifact of the vision sensor that has different resolutions in the $x$ and $y$ directions. Errors visible are due to sensor limitations and spring forces in the joints due to wiring and tubing.

**Computational Savings**

Since the solution of the Jacobian matrix equation is order $(n^3)$ operations (effectively, solving $Ax = b$, for $x$), a reduction of $n$ by 6 (for three-dimensional systems) and by 3 (for two-dimensional systems) can produce substantial computational savings.

For example, using the techniques presented here, a three-dimensional robot with a single seven-degree-of-freedom manipulator arm $n = 6 + 7$ would require 1794 operations to solve the full Jacobian matrix equation,[9] and 322 operations to solve the simplified Jacobian equation. The more manipulator degrees of freedom, the greater the savings in computation.

Consider the case of a two-armed free-flying robot that has six degrees of freedom in its base and two arms each with seven degrees of freedom: $n = 6 + 7 + 7 = 20$.

If the generalized speeds are defined so that they do not mix joint angle rates of the two manipulators (a choice of $u_i \triangleq \dot{q}_i$ guarantees this), it is possible to simplify this system into two decoupled, individual manipulator Jacobian equations. A three-dimension robot with two seven-degree-of-freedom manipulator arms ($n = 20$) would require 6132 operations to solve the full Jacobian equation, but only 644 operations [$2 \times (n = 7)$] after the manipulator Jacobian matrix is decomposed.

The cost of including centripetal acceleration terms due to robot body angular velocity in our two-dimensional recursive kinematics and inverse dynamics algorithm[9] is 47 total additional operations. The forward kinematics and inverse dynamics for a two-dimensional free-flying robot with two two-link arms can be evaluated in 265 operations; therefore, including the robot body's angular speed does not add significantly to the cost of doing CT control.

## Conclusions

Both full-order and simplified computed torque controllers for manipulator endpoint control and cooperative-arm object manipulation free-flying robot are developed and tested on an experimental robot. The simplified controller offers manipulator performance indistinguishable from the full-order controller but at a substantial savings in computational cost.

The CT controllers are formulated such that the differences between the free-flying robot system and the fixed-base manipulator equivalent are easily examined. It is found that the terms due to the free-flying robot body velocity and acceleration manifest themselves in very different ways. Including base angular velocity in the CT controller model compensates for the nonlinear inertial forces due to base angular velocity and costs little to include. It should not be neglected. Robot body accelerations, on the other hand, are computationally expensive to include in the CT controller model.

A study was made of two common control tasks for a two-armed free-flying robot. A dynamics and control simulation program was used to predict the effect of neglecting the robot body accelerations on the response of open-loop control over a wide range of robot/payload mass distributions. The effect on open-loop control was found to be small, and the effect on closed-loop control was estimated to be on the same order as an uncertainty of mass distribution of 0.7% for the robot under study.

Experiments were performed with a laboratory free-flying robot. Manipulator performance for the full-order controllers was compared to their simplified counterparts. In these tests, the simplification had no noticeable degradation in the controller's performance. In fact, any mismodeling due to neglecting base accelerations was found to be significantly less than that due to friction effects, sensor limitations, and uncertainty in the mass distribution.

The savings in computations offered by neglecting robot body accelerations in this class of manipulator control system—together with the negligible performance degradation that this paper has shown—make it a practical technique for assuring high-performance endpoint control at a more reasonable computational cost.

## Acknowledgments

The work of the first author was supported under NASA Research Grant NCC 2-333. The authors wish to acknowledge the help of the staff and students of the Aerospace Robotics Laboratory at Stanford, particularly Marc Ullman, who was instrumental in the design, construction, and operation of the robot testbed.The authors also wish to acknowledge the constructive comments from the anonymous reviewers that led to improved presentation of the paper.

## References

[1]Alexander, H. L., "Experiments in Control of Satellite Manipulators," Ph.D. Thesis, Dept. of Electrical Engineering, Stanford Univ., Stanford, CA, Dec. 1987.

[2]Umetani, Y., and Yoshida, K., "Continuous Path Control of Space Manipulators Mounted on OMV," *ACTA Astronautica*, Vol. 15, No. 12, 1987, pp. 981–986.

[3]Carignan, C. R., "Control Strategies for Manipulating Payloads in Weightlessness with a Free-Flying Robot," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, Sept. 1987.

[4]Dubowski, S., and Vafa, Z., "A Virtual Manipulator Model for Space Robotic Systems," *Proceedings of the Workshop on Space Telerobotics* (Pasadena, CA), 1987, pp. 342–344.

[5]Koningstein, R., Ullman, M., and Cannon, R. H., Jr., "Computed Torque Control of a Free-Flying Cooperating-Arm Robot," *Proceedings of the NASA Conference on Space Telerobotics* (Pasadena, CA), 1989, pp. 235–243.

[6]Masutani, Y., Miyazaki, F., and Arimoto, S., "Modeling and Sensory Feedback Control for Space Manipulators," *Proceedings of the NASA Conference on Space Telerobotics* (Pasadena, CA), 1989, pp. 287–296.

[7]Van Woerkom, P. T. L. M., and Guelman, M., "Dynamics Modelling, Simulation, and Control of a Spacecraft/Manipulator System," National Aerospace Lab., TR 1, Amsterdam, The Netherlands, Aug. 1987.

[8]Papadopolous, E., and Dubowski, S., "On the Nature of Control Algorithms for Space Manipulators," *Proceedings of the IEEE Robotics and Automation* (Cincinnati, OH), 1990, pp. 1102–1108.

[9]Koningstein, R., "Experiments in Cooperative-Arm Object Manipulation with a Two-Armed Free-Flying Robot," Ph.D. Thesis, Dept. of Aeronautics and Astronuatics, Stanford Univ., Stanford, CA, Oct. 1990.

[10]Kane, T. R., and Levinson, D. A., *Dynamics: Theory and Application*, Series in Mechanical Engineering, McGraw–Hill, New York, 1985, pp. 1–50.