

# Performance Evaluation of Cooperating Robot Movements Using Maximum Load Under Time-Optimal Control

Pierre Larochelle and J.M. McCarthy  
Department of Mechanical Engineering  
University of California, Irvine  
Irvine, California 92717

January 23, 1991

## Abstract

This paper evaluates the performance of a cooperating robot system using movements planned for minimum time. Minimum time movements characteristically require that a set of motors in the robot be driven at their maximum torque throughout the motion. Thus these movements are limited by the combination of motor performance, mechanical advantage of the kinematic chain, and the location of the start and goal positions. By increasing the payload until a minimum time solution is no longer feasible the performance limit of the system for the associated path is obtained. The cases of two robots working individually and together are examined in order to find the performance enhancement obtained by having two robots cooperate.

## 1 Introduction

In this paper we describe the TORUS software package which determines time-optimal paths for general robotic systems. This software is used to evaluate the performance of a pair of cooperating robots. This is done by determining the maximum load feasible for a time-optimal movement of a cooperating robot system along a given path. This movement characterizes the performance limits of the robot system on this path. Bobrow et al. 1985 and Shin and McKay 1985 decompose the problem of determining these movements into optimization along a given path followed by path optimization. McCarthy and Bobrow 1989, and Bobrow et al. 1985 generalized this strategy to determine time-optimal point to point movement for the closed chain robot systems formed by two robots holding the same workpiece.

The focus of this paper is on the maximum load that

can be carried on several different paths by a pair of planar cooperating robots. These maximum payloads are then compared to those achievable by either arm working alone. The results are that while 10lbs is the maximum payload for a single robot, this increases to more than 200lbs when the two robots work together. In addition, we have found that while strong and weak paths may be intuitively obvious for single robots, they are not as readily apparent for cooperating robots.

The motor torques, link masses and dimensions for the robot used in this analysis are an idealized version of an arm under development at Odetics, Inc. While this analysis technique is applicable to general manipulator systems (McCarthy and Bobrow, 1990), the focus here is on planar movements to highlight its use in the performance evaluation of robot systems.

## 2 System Dynamics

In this section the equations of motion are derived for the closed chain formed by two general robots and a workpiece, see Fig. 1 (McCarthy and Bobrow 1990, Huang and McClamroch 1988, and Mills and Goldenberg 1989).

Let  $q_1$  and  $q_2$  be the vectors of joint angles for the left and right robots, respectively, and let  $q_3$  define the position of the workpiece. The equations of motion of a general pair of cooperating robots are obtained by first writing Lagrange's equations for each robot arm and the workpiece separately. The holonomic loop closure constraints and Lagrange multipliers are used to combine the three systems to obtain;

$$\begin{aligned} [M_1(q_1)]\ddot{q}_1 + h_1(q_1, \dot{q}_1) &= u_1 + c_1 \\ [M_2(q_2)]\ddot{q}_2 + h_2(q_2, \dot{q}_2) &= u_2 + c_2 \\ [M_3(q_3)]\ddot{q}_3 + h_3(q_3, \dot{q}_3) &= u_3 + c_3 \end{aligned} \quad (1)$$

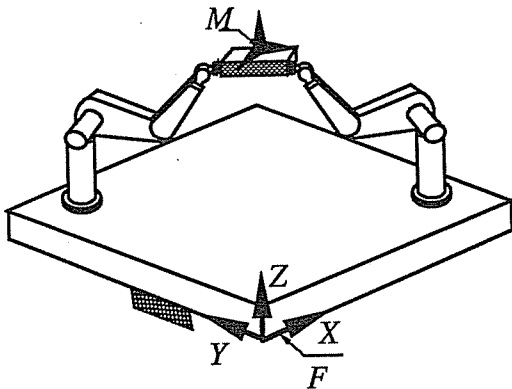


Figure 1: Two cooperating PUMA manipulators.

Where  $[M_i]$ ,  $h_i$  are the corresponding mass matrix, Coriolis and gravity terms, respectively,  $u_i$  are the generalized forces due to the actuators for each system, and  $c_i$  represent the generalized forces due to the interaction of the three systems.

In order to relate the constraint forces  $c_i$  to the dynamics of the system, use the fact that the constraints do no work. That is to say they do not absorb any power. Hence, defining  $c^T = (c_1^T, c_2^T, c_3^T)$  and  $q^T = (q_1^T, q_2^T, q_3^T)$  we have,

$$c^T \dot{q} = 0 \quad (2)$$

for all admissible velocities  $\dot{q}$ .

For constrained systems, the  $\dot{q}_i$  are not arbitrary; they must satisfy the closure equations which guarantee that the robots hold the workpiece. Using the standard Lagrangian formulation of the dynamics, it can be shown that there is a vector  $\lambda \neq 0$  such that

$$c = [J(q)]^T \lambda \quad (3)$$

where  $[J(q)]$  is the Jacobian of the complete system, (Greenwood, 1988). This vector  $\lambda$  defines the internal forces of the closed chain system. The coupled equations of motion for the system become

$$[M(q)]\ddot{q} + h(q, \dot{q}) = u + [J(q)]^T \lambda, \quad (4)$$

where

$$[M(q)] = \begin{bmatrix} [M_1] & 0 & 0 \\ 0 & [M_2] & 0 \\ 0 & 0 & [M_3] \end{bmatrix}$$

$$h(q, \dot{q}) = \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \end{Bmatrix} \text{ and } u = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \quad (5)$$

In general the forces  $u$  are related to the vector of actuator torques  $\tau$  by a matrix transformation:

$$u = [B]\tau \quad (6)$$

### 3 Time Optimal Control

#### 3.1 Strategy

The given path is parameterized in terms of a path parameter  $s$ . This parameter  $s$  then identifies the position of the workpiece in space as it traverses the specified path. The total time to traverse the path is,

$$t_{total} = \int_{s_i}^{s_f} \frac{ds}{\dot{s}} \quad (7)$$

Our goal is time-optimal control. That is to say we seek the control that minimizes the time required for the system to traverse the given path. This control maximizes the velocity of the system along the path. To maximize  $\dot{s}$  the system must be either accelerating, or decelerating, as much as possible. The time-optimal solution is found by determining the maximum, or minimum, acceleration  $\ddot{s}$  along the path that is attainable by the robot system, (Bobrow et al., 1985 and Shin and McKay, 1985). The *time-optimal control* is found by identifying the points along the path at which the system must switch from maximum acceleration to minimum acceleration.

#### 3.2 Implementation

The *time-optimal trajectory* along a given path can now be found. Using the closure equation and the path equation, the equations of motion for the system, Eq. 4, are written in terms of the path parameter  $s$ , and its first two time derivatives, (Chu, 1990). For each  $s$  and  $\dot{s}$  Eq. 4 becomes a set of  $n + p + m + 1$  linear equations in  $2(p + m + 1)$  unknowns; where  $n$  is the number of coordinates used to describe the system,  $p$  is the number of actuators, and  $m$  is the number of constraint equations. Eq. 4 and the bounds on  $\tau$  and  $\lambda$  combine to form a linear programming problem;

$$\begin{aligned} &\text{Maximize/minimize } \dot{s} \text{ subject to:} \\ &a(s)\ddot{s} - [B]\tau - [C]^T \lambda = d(s, \dot{s}) \end{aligned} \quad (8)$$

$$\begin{aligned} \tau^{min} &\leq \tau \leq \tau^{max} \\ \lambda^{min} &\leq \lambda \leq \lambda^{max} \end{aligned}$$

The second step is to construct the *maximum velocity curve*. The maximum velocity curve represents the maximum velocity the system can have at each point on the specified path. The maximum velocity curve is found by numerically determining the maximum value of  $\dot{s}$ ,  $\max(\dot{s})$ , for each  $s$  along the path. For each value of  $s$  along the

path an arbitrary guess is assigned to  $\max(\dot{s})$ . If there exists a solution to Eq. 8 then the initial guess is increased until a solution does not exist. If there does not exist a solution for the initial guess, the guess is decreased until a valid solution is found. Note that for a solution to be valid both  $\tau$  and  $\lambda$  must be within their prescribed limits. Using this method the  $\max(\dot{s})$  for each  $s$  along the path can be approached using an algorithm such as the *bisection method*.

Once the maximum velocity curve has been constructed the time-optimal trajectory can be found. First, using Eq. 8 solve for the minimum acceleration  $\ddot{s}$  at the end point of the path and integrate backwards in time to construct the *minimum acceleration curve*. This integration is performed until the curve intersects the maximum velocity curve or one of the axis in the  $\dot{s}$ - $s$  plane. Similarly, find the *maximum acceleration curve* by integrating forward in time from the beginning of the path. Connecting these curves in the  $\dot{s}$ - $s$  plane are *switching points*. These are the points along the path at which the system changes from maximum acceleration to maximum deceleration and vice-versa. These points were found using a modified form of the *ladder search algorithm*, (Chu, 1990).

## 4 Finding the Optimal Path

The given path is varied using a non-linear optimization routine to find the path which minimizes the time for the system to go from its specified start and end positions. That is to say the control points of the path are manipulated to find the time-optimal path that takes the robot system from the start point to the end point. This path is not a global optimal path in that it is limited by the number of control points of the B-Splines. However, as the number of control points is increased the path will approach the global time-optimal path (Chu, 1990).

## 5 Software Overview

The methodology described above has been implemented in a software package entitled, TORUS (Time Optimal Robot System), see Fig. 2. Initially, the equations of motion are coded in EOM, and the program MAP rewrites these equations in terms of the path parameter  $s$ . The non-linear optimization package used to find the control points of the time-optimal path is called ADS (Vanderplaats, 1984). The algorithm used is the *variable metric method for unconstrained minimization* (Davidon, Fletcher, and Powell, 1963). On first iteration it is assumed that the path being considered is not optimal and the time-optimal control problem along this path is solved. The control points are sent to FORBAK (FOward and BAckward integration controller) to integrate the equations of motion in order to construct

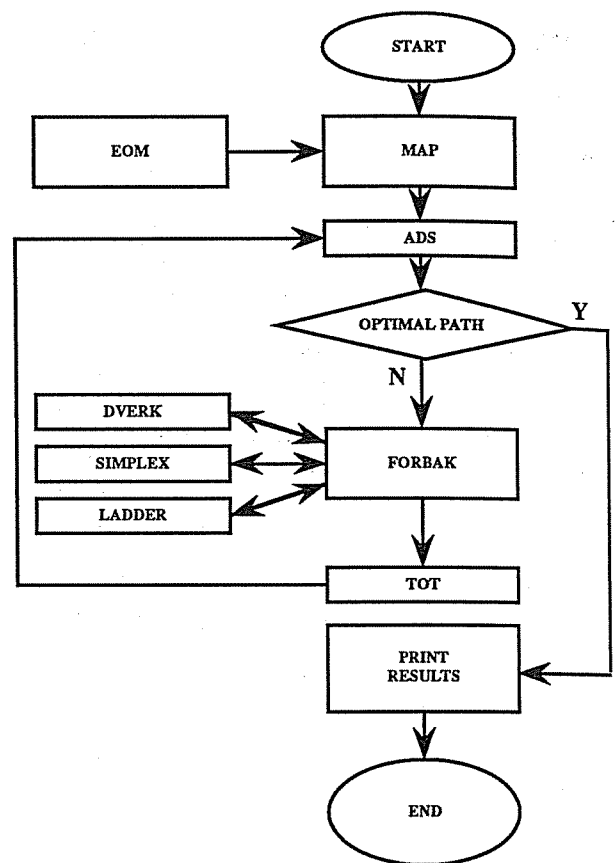


Figure 2: TORUS Flowchart

the maximum and minimum acceleration curves. To accomplish this FORBAK calls upon three subprograms: DVERK is a 5th and 6th order Runge-Kutta (IMSL) integration routine used to integrate the equations of motion. SIMPLEX is a package written to solve the linear programming problem of equation Eq. 8 using the simplex method (The, 1979). LADDER is a program written to find the switching points that connect the maximum and minimum acceleration curves using the ladder search algorithm. TOT (Time-Optimal Trajectory) is a program written to take the output of FORBAK, the maximum and minimum acceleration curves and the switching points, and construct the time-optimal trajectory for the system. From the time-optimal trajectory  $t_{total}$  is computed and its value is sent to ADS along with the control points that define the path. If ADS determines that a time-optimal path has not been found it computes the control points for the next iteration. If ADS has decided the path is the time-optimal path, that is to say the algorithm has converged to a minimum  $t_{total}$ , then the output is printed and execution ceases.

## 6 Finding the Maximum Payload

In the cases studied the system starts and ends the path at rest. As the payload is increased, the maximum and minimum accelerations  $\ddot{s}$  that the system can attain along the path are decreased. Therefore, as the payload is increased the time-optimal trajectory falls in the  $\dot{s}$ - $s$  plane. The trajectory curve in the  $\dot{s}$ - $s$  plane drops towards the  $s$  axis.

The payload has exceeded the capacity of the system when there is some configuration ( $s^*$ ) for which there is no valid solution to Eq. 8. That is to say that at  $s^*$  the maximum, or minimum, acceleration curves yield  $\ddot{s}^* = 0$ . In which case, the system becomes stuck on the path and is no longer able to move. Physically, at  $s^*$  the motor limits or reaction force limits, have to be exceeded for the robot system to continue along the path. The maximum payload for a path is found by using the *bisection method* to incrementally increase the payload until a valid solution to Eq. 8 does not exist.

## 7 Numerical Examples

To illustrate the application of the algorithm described in this paper two identical prototype robots were used. The paths studied are made up of three connected *B-splines*. The robots are positioned such that their base joints are 0.5 (m) apart, see Fig. 3. The robots' geometry, mass, and peak motor torques are as follows:

Link	Length (m)	Mass (kg)
K	0.6136	29.5
L	0.5128	6.81
H	0.7620	varied

Motor	Max. Torque (Nm)	Min. Torque (Nm)
$\theta$	452.0	-452.0
$\phi$	226.0	-226.0
$\psi$	67.8	-67.8

### 7.1 Case 1

The first path studied is shown in Fig. 4. This path is linear in that all three path parameters,  $\{x, y, \alpha\}$ , vary linearly with respect to  $s$ . The maximum payload of each arm working alone on this path was found to be 10lbs. Note the slope of the time-optimal trajectory approaches zero at the end of the path ( $s=3$ ). In this position the payload is cantilevered at the robot's wrist and the torque limits at the wrist determine the maximum payload. Results for both the left and right robots are shown in Fig. 5 and Fig. 6, respectively. Second, the cooperation of the two robots was studied. If the reaction

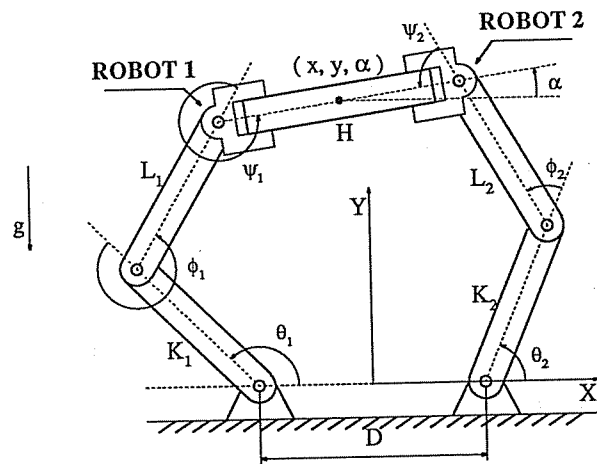


Figure 3: Two cooperating planar robots.

forces are bounded, that is to say the amount of squeezing and pulling on the object are limited to  $\pm 200$  (N) the system's maximum payload was found to be 20lbs. If the reaction forces are not bounded, the maximum payload increases to 218lbs, see Fig. 7. Finally, the path was optimized for the 218lbs payload and is shown in Fig. 8. Along this path, the time for the system to travel from the start position to the goal position decreased 18 %.

### 7.2 Case 2

In this case we examined a path along which we thought the robots would be strong. A linear path with the workpiece centered between the robots was studied, see Fig. 9. The left and right robots, working independently, were found to have a maximum payload of 6lbs. The robots working together with no bounds on the reaction forces at the workpiece were found to have a maximum payload of 20lbs.

### 7.3 Case 3

In the final case studied we analyzed what we thought would be a weak path for the robot system. A linear path with the workpiece going from left to right across the front of the robots. The path selected is shown in Fig. 10. For a robot working independently the maximum payload was found to be 10lbs. As in Case 1, the payload is limited by the robot's wrist. For the robots working in cooperation with no bounds on the reaction forces the maximum payload increased to 130lbs. This result was counter to our expectations.

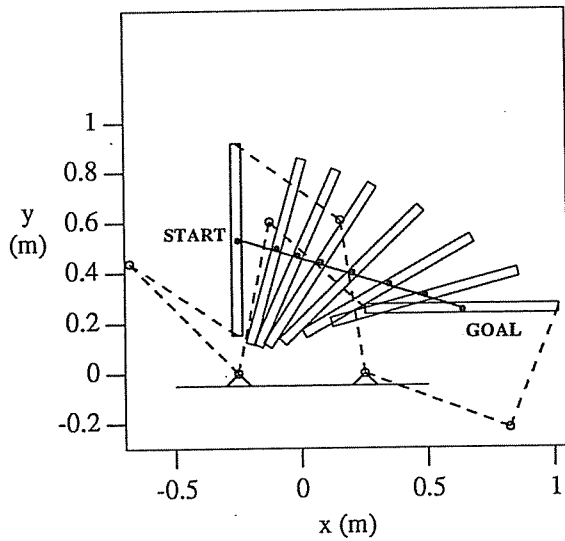


Figure 4: The linear path.

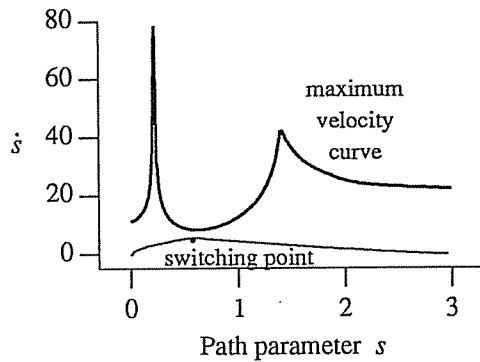


Figure 5: Time-optimal trajectory for the maximum payload along the linear path for the left robot.

Elapsed time: 4.90 sec. Payload: 10 lb.

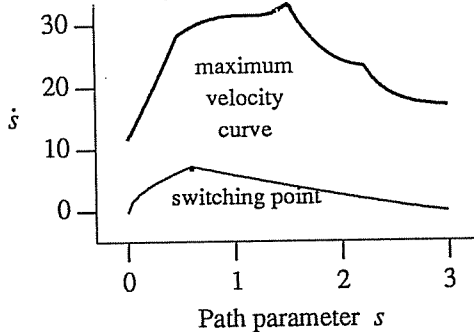


Figure 6: Time-optimal trajectory for the maximum payload along the linear path for the right robot.

Elapsed time: 2.51 sec. Payload: 10 lb.

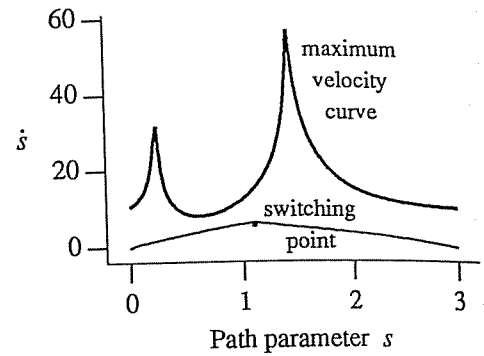


Figure 7: Time-optimal trajectory for the maximum payload of the robot system along the linear path.

Elapsed time: 1.93 sec. Payload: 218 lb.

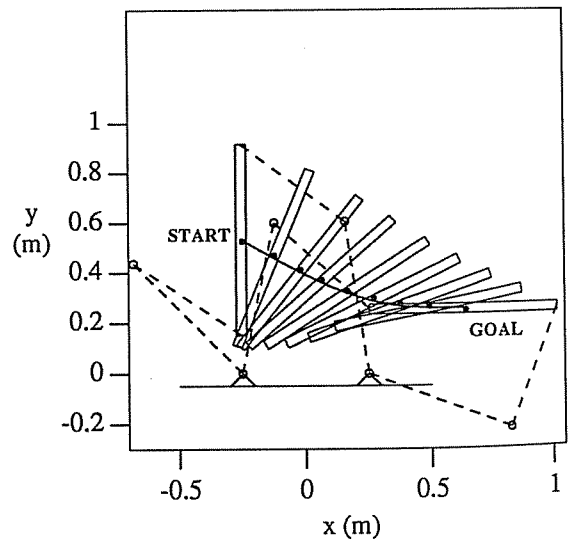


Figure 8: The optimal path.

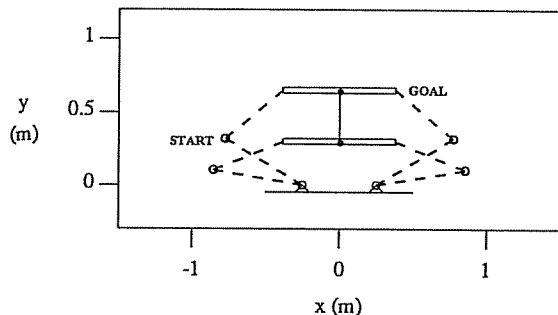


Figure 9

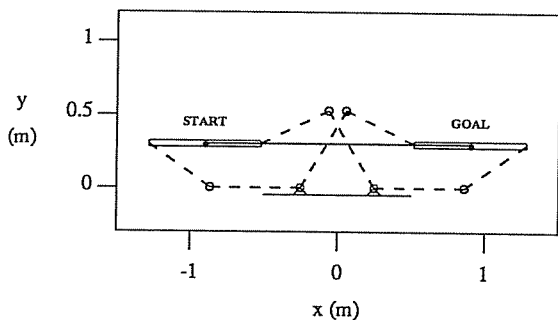


Figure 10

## 8 Conclusion

In this paper we evaluated the performance of a cooperating robot system using movements planned for minimum time. We studied the maximum load that can be carried by the robots working individually and in unison along three different paths. In Case 1, the pick and place path, we found that the maximum payload for the independent robots was 10lbs, while the maximum payload for the cooperating robots was 218lbs. In addition, we optimized the path for the 218lbs payload and achieved an 18 % decrease in time to travel from the start position to the goal position. In Case 2, the vertical path, we found the independent robots able to carry 6lbs while the cooperating system was able to carry 20lbs. In Case 3, the horizontal path, we found the individual robot's maximum payload to be 10lbs, however, the maximum payload for the cooperating system was found to be 130lbs. One insight gained from this analysis is that strong and weak paths may be intuitively obvious for single robots, however, for cooperating robots they seem counter intuitive. An important result is that while 10lbs is the maximum payload for a single robot, this can increase to more than 200lbs when the robots work together.

## 9 Acknowledgements

The support of the National Science Foundation, grant MSM-8720580, is gratefully acknowledged. In addition, this research was supported in part by the University of California, Irvine through an allocation of computer time on the Convex-C240 supercomputer.

## References

- [1] Bobrow, J., Dubowski, S. and Gibson, J. Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research*, Fall 1985.
- [2] Shin, K., and McKay, N. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, June 1985
- [3] McCarthy, J.M. and Bobrow, J. Time-optimal movement of cooperating robots. *NASA Conference on Space Telerobotics*, Jan. 1989.
- [4] Bobrow, J., and McCarthy, J.M. The number of saturated actuators and constraint forces during time-optimal movement of a general robotic system. *1990 ASME Mechanisms Conference*, 1990.
- [5] McCarthy, J.M., Bobrow, J., and Chu, V. Time-optimal control of two robots holding the same workpiece. *1990 Conference on Decision and Control*, 1990.
- [6] Huang, H., and McClamroch, N. Time-optimal control for a robotic contour following problem. *IEEE Journal of Robotics and Automation*, 1988.
- [7] Mills, J., and Goldenberg, A. Force and position control of manipulators during constrained motion tasks. *IEEE Journal of Robotics and Automation*, 1989.
- [8] Vanderplaats, G. ADS - A FORTRAN program for automated design synthesis. *Naval Post Graduate School*, 1984.
- [9] Davidon, W. Variable metric method for minimization. *Argonne National Lab., ANL-5990*, 1959.
- [10] Fletcher, R., and Powell, M. A rapidly-convergent method for minimization. *Computer Journal*, Vol 6, No. 2, 1963.
- [11] Thie, P. *An Introduction to Linear Programming and Game Theory*, Wiley and Sons, New York, 1979.
- [12] Greenwood, D. *Principles of Dynamics*, Prentice Hall, New Jersey, 1988.
- [13] Chu, V. *Master's Thesis*, Univ. of Calif., Irvine, 1990.