January 1993

Time-Optimal Trajectories for Cooperative Multi-Manipulator Systems

Seungbin B. Moon
Purdue University School of Electrical Engineering

Shaheen Ahmad
Purdue University School of Electrical Engineering

TIME-OPTIMAL TRAJECTORIES FOR COOPERATIVE MULTI-MANIPULATOR SYSTEMS

SEUNGBIN B. MOON
SHAHEEN AHMAD

TR-EE 93-3
JANUARY 1993

SCHOOL OF ELECTRICAL ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

Time-Optimal Trajectories for Cooperative Multi-Manipulator Systems

Seungbin B. Moon and Shaheen Ahmad
Real-Time Robot Control Laboratory
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907
U.S.A.

ABSTRACT

In this paper we present two schemes for planning the time-optimal trajectory for cooperative multi-manipulator system (CMMS) carrying a common object. We assume that the desired path is given and parameterizable by an arclength variable. Both approaches take into account the dynamics of the manipulators and the dynamics of the object. The first approach employs linear programming techniques, and it allows us to obtain the time-optimal execution of the given task utilizing the maximum torque capacities of the joint motors. The second approach is a sub-time-optimal method which is computationally very efficient. In the second approach we try to divide the given load into a share for each manipulator in the CMMS in a manner in which the trajectory acceleration/deceleration is maximized, hence the trajectory execution time is minimized. This load distribution approach uses optimization schemes which degenerate to a linear search algorithm for the case of two robots manipulating a common load, and this results in significant savings on the computation time. The load distribution scheme not only enables us to reduce the computation time but also gives us the possibility of applying this method in real time planning and control of CMMS. Further, we show that under certain object trajectories the load distribution scheme yields truly time-optimal trajectories.

I. INTRODUCTION

A cooperative multi-manipulator system(CMMS) is defined as the system of multiple robots handling a common object, forming closed kinematic chains, as shown in Figure 1. In many robotic applications, CMMS is needed for handling an object due to the inherent nature of the task itself or the desire to enhance the flexibility. For example, in a flexible assembly system one may use multiple robots to assemble two or more parts into a final product. If an object is so big that it can not be handled by a single robot, the use of multi-robots may be required. For instance, in the space station, multiple robots will be used to handle objects of large inertia and size.

Research on time-optimal trajectory planning for CMMS has to consider many issues which may not be important in the single robot trajectory planning. These issues include the incorporation of the object dynamic equation in CMMS[13], the distribution of the load among the multiple robots[9,15,16,17], the kinematic constraints represented by the closed chains[14], and the internal force control[11]. Moon and Ahmad[1, 2] applied the trajectory time scaling concept developed for single robot by Hollerbach[3] to trajectory scaling for multi-robot. It was shown that linear programming could be used to find the trajectory speed-up factor in order to minimize the traversal time for a given velocity profile. Since it was assumed that the velocity profile is fixed, the scaling algorithm could not obtain the true time-optimal trajectory for CMMS, but the merit of this scaling algorithm is that it can be readily used to improve the existing control structure in a simple fashion. Time-optimal trajectories for CMMS when the desired path is known in advance has been studied independently both by Moon and Ahmad[4] and Bobrow et. al.[5], and linear programming approach was used to determine the trajectories. Both Moon and Ahmad[4] and Bobrow et. al.[5] employed the parameterization of the path concept used for single robots by Shin and McKay[6] and Bobrow et. al.[7]. Chen[18] also developed a similar algorithm based on the parameterization and linear programming, but he did not address the internal force issue.

In this paper we first summarize the general time-optimal trajectory planning method for cooperative multi-manipulator system as presented before[4]. In this approach, linear programming techniques are used to find the extreme value of the acceleration we can obtain from the given dynamic equations and force torque constraints. This approach requires considerable computation time and this fact prohibits the possibility of employing this algorithm in real time applications.

The second approach developed in this paper employs search schemes in order to find the optimal load distribution factors which generate the sub-minimum-time trajectory for the CMMS. In the dual manipulator case, we need to find a single distribution factor which divides the required load between two manipulators. If we have more than two robots, the search space naturally increases, that is, if we employ n robots handling an object, we need to develop a search algorithm over an $(n-1)$ space to find the optimal distribution factors. In the case of two robots, as presented in this paper, a systematic line search algorithm is developed, which is computationally efficient. The method described in this paper may be used in real time planning and control of CMMS since the algorithm only takes a fraction of the computation time in comparison to the first approach which yields the time optimal trajectory. The sub-optimal algorithm can be improved by employing the systematic search schemes to find the switching points in the phase plane[12].

This paper is divided into six sections. Multi-robot system model is given in Section 2. Section 3 describes the general trajectory planning problem and describes an approach which achieves the time-optimal trajectory by using the linear programming method. In Section 4, we present the sub-time-optimal trajectory planning algorithm employing the optimal load distribution scheme. Simulation results are given in Section 5. The discussions and conclusions obtained from our results are given in the final section.

II. Dynamic Models For Cooperative Multi-Manipulator System

2.1. Dynamic Equations and Constraints on the Input Torques.

The dynamic equations for the i-th robot in the cooperative multi-manipulator system as shown in Figure 1 can be expressed as follows.

$$\tau_i(t) = D_i\ddot{q}_i(t)+\dot{q}_i(t)C_i\dot{q}_i(t)+G_i+J_i^TF_i \qquad \text{for } i = 1, .., n \,. \tag{1}$$

Here we assume that the i-th manipulator has the n_i joints and $\tau_i \in R^{n_i}$ is the vector of joint torques, $q_i(t) \in R^{n_i}$ is the vector of joint position, $D_i \in R^{n_i \times n_i}$ is the manipulator inertia matrix, $C_i \in R^{n_i \times n_i \times n_i}$ is the tensor of Coriolis and centrifugal terms, $G_i \in R^{n_i}$ is the vector of gravitational terms, $J_i \in R^{6 \times n_i}$ is the manipulator Jacobian matrix, and $F_i \in R^6$ is the end-effector forces exerted by the i-th manipulator onto the common object. It is expressed in the base coordinate of each robot.

The dynamic equations of motion of the common object held by the multiple robots are given as

$$P = B\,F\,. \tag{2}$$

Here $P = \begin{bmatrix} m\ddot{p}(t) + mg \\ I\dot{\omega}(t) + \omega(t) \times I\omega(t) \end{bmatrix}$, $B = [B_1\ B_2\ \ldots B_n]$, and $F = [F_1^T F_2^T\ \ldots\ F_n^T]^T$,

and $m \in R^+$ is the mass of object, $g \in R^3$ is the gravity vector, and $I \in R^{3x3}$ is the inertia matrix of the object. Linear acceleration, $p = [\ddot{p}_x, \ddot{p}_y, \ddot{p}_z]^T$ and angular acceleration, $\dot{\omega} = [\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z]^T$, of the object are expressed in the world coordinate frame, and $B_i \in R^{6x6}$ is defined as follows.

$$B_i = \begin{bmatrix} I_3 & 0_{33} \\ S_i & I_3 \end{bmatrix}. \tag{3}$$

Here $I_3 \in R^{3x3}$ is the identity matrix which implies the orientation of the i-th manipulator reference frame is identical to the orientation of the world reference frame, and 0_{33} is the 3x3 matrix of zeroes, and the matrix $S_i \in R^{3x3}$ is defined for different types of contact.

$$S_i = \begin{cases} \begin{bmatrix} 0 & -r_{iz} & r_{iy} \\ r_{iz} & 0 & -r_{ix} \\ -r_{iy} & r_{ix} & 0 \end{bmatrix} & \text{for a soft point contact} \\ 0_{33} & \text{for a rigid contact} \end{cases}$$

where $r_i = [r_{ix}, r_{iy}, r_{iz}]^T$ is a vector from the center of mass of object to the point of contact with the i-th manipulator. The definitions of the different contact types used in this paper are given below. Notice that in equation (2), all quantities are expressed in world coordinate frame.

Definition 2.1: (Soft Point Contact)

We define a **soft point contact** as the contact in which no positional change of contact point is allowed, while the relative rotational motion between the object and the end-effector can occur.

Definition 2.2: (Rigid Contact)

We define a **rigid contact** as the contact in which neither positional change of contact point nor the relative rotational motion between the object and the end-effector can occur.

When the object is grasped by the gripper with a rigid contact, the linear forces applied onto the object by the end-effector do not impart angular moments onto the object, while in a soft point contact the linear forces generate angular moments onto the object. This is the reason why we have defined the two different matrices for S_i in equation (3).

We assume that the joint torques generated by motors are constrained by constants along the path, i.e., $\tau_i^- \le \tau_i(t) \le \tau_i^+$, $i = 1,.., n$ for all t. Then,

$$\tau^- \le \tau(t) \le \tau^+ \tag{4}$$

where $\tau = [\tau_1^T \; .. \; \tau_n^T]^T$, $\tau^- = [(\tau_1^-)^T \; .. \; (\tau_n^-)^T]^T$, and $\tau^+ = [(\tau_1^+)^T \; .. \; (\tau_n^+)^T]^T$.

2.2. Parameterized Dynamic Equations

Assume that the joint paths are parameterizable by the arclength function s, i.e., $q_i(t) = \tilde{q}_i(s(t))$, $i = 1,.., n$ for all t, where q_i and $\tilde{q}_i$ are different functions defined on different ranges. Since $\dot{q}_i(t) = \tilde{q}_i'(s)\dot{s}(t)$ and $\ddot{q}_i(t) = \tilde{q}_i''(s)\,\dot{s}^2(t) + \tilde{q}_i'(s)\ddot{s}(t)$, the equation of motion of the *i-th* manipulator parameterized by the arclength function is given by

$$\tau_i(t) = D_i \tilde{q}_i' \ddot{s} + \{D_i \tilde{q}_i'' + \tilde{q}_i' C_i \tilde{q}_i'\} \dot{s}^2 + G_i + J_i^T F_i \tag{5}$$

where a prime next to a variable represents d/ds and a dot over the variable represents d/dt.

Assume the path of the object is also parameterizable by the arclength function s, then we have $p(t) = \tilde{p}(s(t))$, and $\phi(t) = \tilde{\phi}(s(t))$, where p and $\tilde{p}$ are different functions defined on different ranges and ϕ and $\tilde{\phi}$ are also different functions defined on different ranges. Since $\dot{p} = \tilde{p}'(s)\dot{s}$, $\ddot{p} = \tilde{p}''(s)\dot{s}^2 + \tilde{p}'(s)\ddot{s}$, $\dot{\phi} = \tilde{\omega}(s)\dot{s}$, and $\ddot{\phi} = \tilde{\omega}'(s)\dot{s}^2 + \tilde{\omega}(s)\ddot{s}$, the parameterized object dynamics is as follows.

$$P = B\,F = H\ddot{s} + c \tag{6}$$

where $H = \begin{bmatrix} m\tilde{p}'(s) \\ I\tilde{\omega}(s) \end{bmatrix}$ and $c = \begin{bmatrix} m\tilde{p}''(s) \\ I\tilde{\omega}'(s) + \tilde{\omega}(s) \times I\tilde{\omega}(s) \end{bmatrix} \dot{s}^2 + \begin{bmatrix} mg \\ 0 \end{bmatrix}$.

III. Time-Optimal Trajectory Planning Using Linear Programming Technique

3.1. Determining The Time-Optimal Trajectory On The Phase Plane

The objective of the time-optimal trajectory planning scheme is to minimize the traversal time required to move the object from the initial point to the final point. If we assume that the desired path of the object is known and parameterizable by an arclength function s, then the traversal time t_f may be expressed as,

$$t_f = \int_{t_o}^{t_f} dt = \int_{s_o}^{s_f} \frac{dt}{ds}\, ds = \int_{s_o}^{s_f} \frac{1}{\dot{s}}\, ds \tag{7}$$

where $q_i(t_o) = \tilde{q}_i(s_o)$ and $q_i(t_f) = \tilde{q}_i(s_f)$. Notice that we assumed that $t_o = 0$ without loss of generality. From this equation we observe the fact that in order to minimize the

traversal time t_f we are required to select the parameterized velocity profile, $\dot{s}$, as high as possible over the duration of the movement[6, 7].

If we can obtain the maximum and minimum possible accelerations at any point in the parameterized phase plane of $(s, \dot{s})$, then we can generate the time-optimal trajectory along the path, using the schemes developed previously for single robots[6, 7]. Once $\ddot{s}$ is known, we can generate the phase plane trajectory by solving the differential equations[6, 7]. The question is how to find the maximum or minimum admissible accelerations in order to accelerate or to decelerate the robot with the bounded joint motor torques. In the single robot case, the process of finding the extreme values of the acceleration is quite straightforward as we do not have to consider the distribution of the load, the internal force constraints, or the redundant actuation as in CMMS. Unlike the single robot case, we need to employ linear programming methods in order to obtain the maximum or minimum admissible acceleration in CMMS.

3.2. Constraints On The Internal Forces

The end-effector forces, $\boldsymbol{F}_i \in \boldsymbol{R}^6$ for i = 1, .., n , applied onto the carried object can be divided into the motion forces $\boldsymbol{F}_{Mi} \in \boldsymbol{R}^6$ and the internal forces $\boldsymbol{F}_{Ii} \in \boldsymbol{R}^6$, thus $\boldsymbol{F}_i = \boldsymbol{F}_{Mi} + \boldsymbol{F}_{Ii}$. The definitions are given below.

Definition **3.1:** (Internal Forces)

The internal forces, $\boldsymbol{F}_{Ii}$, are defined as the set of end-effector generalized forces which do not contribute to the motion of the object.

Definition **3.2:** (Motion Forces)

The motion forces, $\boldsymbol{F}_{Mi}$, are defined as the set of end-effector generalized forces which contribute to the motion of the object.

Proposition **3.1:** Linear forces exerted by the end-effectors of CMMS onto the object may be resolved into three orthonormal directions of tangential, normal, and binormal directions, denoted by the vectors, e,, $\boldsymbol{e}_n$, $\boldsymbol{e}_b$, respectively. The tangential direction is the direction along the path and the normal and binormal directions are obtained from the relationships of e, $\cdot\ \boldsymbol{e}_n = 0$ and $\boldsymbol{e}_b$ = e, $\times\ \boldsymbol{e}_n$. Then, any linear forces resolved along the normal direction or the binormal direction act as the internal forces.

Proof:

Since linear forces exerted by the end-effectors generate the necessary resultant linear force to move the object, they must satisfy the object dynamic equations denoted by the

upper 3 components of the equation (2). On resolving both sides of equation (2) into orthonormal directions of e,, e_n, and e_b, we conclude that the sum of linear forces along the tangential direction is the resultant force, and the sum of linear forces along the normal direction is zero, as the required resultant force along the normal direction is zero. Likewise, the sum of linear forces along the binormal direction is zero. From the above definition of the internal forces, we have now proved that any linear forces along the normal and binormal directions act as the internal forces. Q.E.D.

Excessive internal forces may cause the carried object to be squeezed or stretched along those directions in which the internal forces are exerted. In order to prevent these undesirable effects, we need to identify the internal forces and constrain them. Note that any linear forces along the negative tangential diction also act as internal forces. This is seen from the fact that the sum of linear tangential forces must be equal to the required resultant forces. Likewise, any negative directional angular moments along the coordinate frame axes also act as the internal forces. We may constrain these internal forces as follows.

$$F_{Ii}^{-}(t) \le \Lambda_i F_i(t) \le F_{Ii}^{+}(t) \qquad i = 1, .., n . \tag{8}$$

Here $\Lambda_i = \begin{bmatrix} P_r & 0_{33} \\ 0_{33} & I_3 \end{bmatrix} \in R^{6x6}$, where P_r is a projection matrix defined as $P_r = \begin{bmatrix} e_t^T \\ e_n^T \\ e_b^T \end{bmatrix}$

and the limits $F_{Ii}^{-}(t)$ and $F_{Ii}^{+}(t)$ are prespecified. The first three components of equation (8) represent the constraints on the linear internal forces along the tangential, normal, and binormal directions, and the last three components represents the constraints on the angular internal forces(moments). The linear internal force on the tangential direction and the angular internal moments about the world coordinate frame axes can be limited by constraining the lower bounds in the respective components in equation (8). The vector notation for the internal force constrains is as follows.

$$F_I^{-}(t) \le \Lambda F \le F_I^{+}(t) \tag{9}$$

where $\Lambda = \begin{bmatrix} \Lambda_1 & 0 & \dots & 0 \\ 0 & \Lambda_2 & \dots & 0 \\ & & \vdots & \\ 0 & \dots & 0 & \Lambda_n \end{bmatrix} \in R^{6nx6n}$.

3.3. Linear Programming Problem(LPP)

We are ready to state an algorithm to find the extreme values of the acceleration at the given point in the parameterized phase plane. First, let $x = [\ \tau^T, F^T, s\ ]^T \in R^{12n+1}$.

Then we can formulate a linear programming problem to find the extreme value of the acceleration, $\ddot{s}$, as follows.

Find $\boldsymbol{x}$ which minimizes (or maximizes)

$$s = [\,0, 0, \ldots, 0, 1\,]^T \boldsymbol{x} \tag{10}$$

subject to

$$\boldsymbol{\Gamma}_1 \boldsymbol{x} = \boldsymbol{b}_1 \tag{11}$$

$$\boldsymbol{b}_2^- \le \boldsymbol{\Gamma}_2 \boldsymbol{x} \le \boldsymbol{b}_2^+ \tag{12}$$

$$\boldsymbol{x}^- \le \boldsymbol{x} \le \boldsymbol{x}^+ \tag{13}$$

where,

$$\boldsymbol{\Gamma}_1 = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{J} & -\boldsymbol{E} \\ \boldsymbol{0} & \boldsymbol{B} & -\boldsymbol{H} \end{bmatrix}, \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1^T & 0 & \cdots & 0 \\ \boldsymbol{0} & \boldsymbol{J}_2^T & \cdots & \boldsymbol{0} \\ & & \vdots & \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \mathrm{J}_n^T \end{bmatrix}, \boldsymbol{E}(s) = \begin{bmatrix} \boldsymbol{D}_1 \tilde{\boldsymbol{q}}_1' \\ \vdots \\ \boldsymbol{D}_n \tilde{\boldsymbol{q}}_n' \end{bmatrix},$$

$$\boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{b}_{11} \\ \boldsymbol{b}_{12} \end{bmatrix}, \boldsymbol{b}_{11} = \begin{bmatrix} \{\boldsymbol{D}_1 \tilde{\boldsymbol{q}}_1'' + \tilde{\boldsymbol{q}}_1' \boldsymbol{C}_1 \tilde{\boldsymbol{q}}_1'\} \dot{s}^2 + \boldsymbol{G}_1 \\ \vdots \\ \{\boldsymbol{D}_n \tilde{\boldsymbol{q}}_n'' + \tilde{\boldsymbol{q}}_n' \boldsymbol{C}_n \tilde{\boldsymbol{q}}_n'\} \dot{s}^2 + \boldsymbol{G}_n \end{bmatrix}, \boldsymbol{b}_{12} = \boldsymbol{c},$$

$$\boldsymbol{\Gamma}_2 = [\,\boldsymbol{0} \;\; \boldsymbol{\Lambda} \;\; \boldsymbol{0}\,], \quad \boldsymbol{b}_2^+ = \boldsymbol{F}^+_1, \quad \boldsymbol{b}_2^- = \boldsymbol{F}^-_1,$$

$$\boldsymbol{x}^- = [\,(\boldsymbol{\tau}^-)^T, (\boldsymbol{F}^-)^T, -\infty\,]^T, \quad \boldsymbol{x}^+ = [\,(\boldsymbol{\tau}^+)^T, (\boldsymbol{F}^+)^T, \infty\,]^T.$$

At any point on the phase plane $(s, \dot{s})$, we can find from the LPP the minimum and maximum possible acceleration, $\ddot{s}_{min}$ and $\ddot{s}_{max}$, as long as there exists a feasible solution space which satisfies the given constraints, since all the coefficients are functions of s and $\dot{s}$. Trajectory execution is impossible in the inadmissible or infeasible regions of the phase plane, as in those regions constraints given in equation (11), (12), and (13) are not satisfied. At the boundary of the inadmissible regions the maximum acceleration and minimum acceleration are the same. Once we obtain minimum and maximum acceleration, $\ddot{s}_{min}$ and $\ddot{s}_{max}$ which satisfy the LPP, it is guaranteed that any value of $s \in [\,\ddot{s}_{min}, \ddot{s}_{max}\,]$ satisfies the constraints given by equation (11), (12), and (13). Similar proof of this fact is shown in Appendix of our earlier work[1].

Once the maximum and minimum acceleration are obtained, we can apply either Shin's algorithm[6] or Bobrow's algorithm[7] to generate the time optimal trajectory plan for the CMMS. One comment is appropriate at this point. As the algebraic form of extreme accelerations is not available from the LPP, we are not able to develop a systematic search scheme to find the switching points as such scheme requires an algebraic expression of the extreme accelerations[12].

IV. SUB-TIME-OPTIMAL TRAJECTORY PLANNING USING OPTIMAL LOAD DISTRIBUTION(OLD) SCHEMES

In this section we develop a load distribution strategy which enables us to find sub-time-optimal trajectories for CMMS. This method requires considerably less computation time as it utilizes simple algorithmic search methods instead of the LPP method. It generates the true time-optimal trajectory plans under certain limited conditions. This will be shown in Lemma 4.1.

4.1. Changing the Reference Frame of End-Effector Forces

In order to simplify the development of our algorithm we specify the end-effector forces in the object coordinate frame which is located at the center of mass of the carried object. Previously, the end-effector forces were specified in the base coordinate frame of each manipulator. Notice that the object coordinate frame is the world coordinate frame translated to the center of mass of the object. As suggested by Uchiyama and Dauchez[8] and Walker et al.[9], we can transform the end-effector force F_i expressed in the i-th base coordinate frame to the resultant force $\bar{F}_i$ in the object coordinate system.(See Figure 2.)

$$\bar{F}_i = B_i F_i \tag{14}$$

where B_i was defined in equation (3). Therefore, equation (6) can be simplified as follows.

$$P = B F = \sum_{i=1}^{n} B_i F_i = \sum_{i=1}^{n} B_i B_i^{-1} \bar{F}_i = \sum_{i=1}^{n} \bar{B}_i \bar{F}_i = \bar{B} \bar{F} \tag{15}$$

where $\bar{B} = [\, I \; I \; \ldots \; I_6 \,] \in R^{6 \times 6n}$, and $\bar{F} = [\, \bar{F}_1^T \; \bar{F}_2^T \; \ldots \; \bar{F}_n^T \,]^T \in R^{6n}$. Notice that $\bar{B}_i$ is an identity matrix and $\bar{B}$ is different from B in equation (2).

4.2. Load Distribution Scheme Based on The Generalized Inverse

We can obtain the solutions of the equation (15) by using the generalized inverse of matrix B.

$$\bar{F} = \bar{B}^+ P + (I_{6n} - \bar{B}^+ \bar{B}) \varepsilon \tag{16}$$

where $\bar{B}^+$ is the generalized inverse of $\bar{B}$ matrix, I_{6n} is the $6n \times 6n$ identity matrix, and $\varepsilon = [\varepsilon_1^T \; .. \; \varepsilon_n^T]^T \in R^{6n}$ with $\varepsilon_i \in R^6$ is an arbitrary vector. The choice of the generalized inverse matrix is open, and one possible criterion for selecting $\bar{B}^+$ is the one which yields

the weighted minimum norm of $\tilde{F}$, $\|\tilde{F}\| = (\tilde{F}^T A^{-1} F)^{1/2}$, where A is a positive definite matrix. Then, $\tilde{B}^+ = A\tilde{B}^T (B A \tilde{B}^T)^{-1}$, as shown in [10].

In order to simplify the problem we will assume that matrix A is composed of n-diagonal block matrices. The i-th block of matrix A is $a_i I_6$, where a_i is a positive number, then we have,

$$A = \begin{bmatrix} a_1 I_6 & 0 & \dots & 0 \\ 0 & a_2 I_6 & \dots & 0 \\ & & \vdots & \\ 0 & \dots & 0 & a_n I_6 \end{bmatrix} \tag{17}$$

Then,

$$\tilde{B}^+ = \frac{1}{a} \begin{bmatrix} a_1 I_6 \\ \vdots \\ a_n I_6 \end{bmatrix}, \text{ where } a = \sum_{i=1}^{n} a_i . \tag{18}$$

After letting $\alpha_i = \frac{a_i}{a}$, we identify the internal force, $\tilde{F}_I$, as

$$\tilde{F}_I = (I_{6n} - \tilde{B}^+ \tilde{B})\, \varepsilon = \begin{bmatrix} \varepsilon_1 - \alpha_1 \sum_{k=1}^{n} \varepsilon_k \\ \vdots \\ \varepsilon_n - \alpha_n \sum_{k=1}^{n} \varepsilon_k \end{bmatrix} . \tag{19}$$

Notice the sum of internal forces are zero. Since the vector ε is arbitrary, we may assume that $\sum_{k=1}^{n} \varepsilon_k = 0$ without loss of generality. Then, equation (16) becomes

$$\tilde{F} = \begin{bmatrix} \alpha_1 P \\ \vdots \\ \alpha_n P \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} . \tag{20}$$

We will use equation (20) to represent the object dynamics in the optimal load distribution(OLD) scheme. The desired grasping forces can be specified in OLD scheme by specifying the internal forces in the appropriate directions. We expect the OLD scheme to generate slower trajectory compared to the LPP approach, since the OLD scheme imposes more constraints on force distribution compared with the underspecified object dynamic equation (6) used in LPP approach and it does not exploit the freedom to choose the internal forces. The OLD approach requires substantially less computation time to generate the trajectory plans because it employs a simple algorithmic search as opposed to linear programming techniques. However, we will show in Lemma 4.1, OLD algorithm does generate the true time-optimal trajectory if the internal forces are specified and the motion of the object is purely translational or rotational about an axis throughout the path.

We can rewrite the dynamic equation of the manipulator using the equations (5), (14), and (20), then we obtain the dynamic equations of manipulators as

$$\tau_i = (D_i \tilde{q}_i' + \alpha_i J_i^T B_i^{-1} H)\ddot{s} + \{D_i \tilde{q}_i'' + \tilde{q}_i' C_i \tilde{q}_i'\} \dot{s}^2 + G_i + J_i^T B_i^{-1}(\alpha_i c + \varepsilon_i) \qquad (21)$$

where $\sum_{k=1}^{n} \varepsilon_k = 0$.

We can rewrite equation (21) in vector format as

$$\tau = (E + A_\alpha K)\ddot{s} + A_\alpha Y + d \qquad (22)$$

where

$$E(s) = \begin{bmatrix} \tilde{D}_n \tilde{\tilde{q}}_n' \\ \vdots \\ D_n \tilde{q}_n' \end{bmatrix}, A_\alpha = \begin{bmatrix} \alpha_1 I_6 & 0 & \cdots & 0 \\ 0 & \alpha_2 I_6 & 0 & \alpha 0 \\ & & \vdots & \\ 0 & \cdots & 0 & {}_n I_6 \end{bmatrix}, K = \begin{bmatrix} J_1^T B_1^{-1} H \\ J_n^T B_n^{-1} H \\ \vdots \\ J_n^T B_n^{-1} H \end{bmatrix},$$

$$Y = \begin{bmatrix} J_1^T B_1^{-1} c \\ \vdots \\ J_n^T B_n^{-1} c \end{bmatrix}, \text{ and } d = \begin{bmatrix} \{D_1 \tilde{q}_1'' + \tilde{q}_1' C_1 \tilde{q}_1'\} \dot{s}^2 + G_1 + J_1^T B_1^{-1} \varepsilon_1 \\ \vdots \\ \{D_n \tilde{q}_n'' + \tilde{q}_n' C_n \tilde{q}_n'\} \dot{s}^2 + G_n + J_n^T B_n^{-1} \varepsilon_n \end{bmatrix}.$$

Assuming that $n_i = 6$ for each robots, we have 6n equations of following form.

$$\bar{\tau}_i = (E_i + \alpha_I K_i)\ddot{s} + \alpha_I Y_i + d_i, \quad I = 1 + int(\frac{i-1}{6}) \qquad \text{for } i = 1,.., 6n. \qquad (23)$$

Here $\bar{\tau}_i$ is the i-th element of vector τ and E_i, K_i, Y_i, and d_i are the i-th elements of respective vectors, E, K, Y, and d. The $int(\cdot)$ function truncates real number to an integer. Since $\bar{\tau}_i$ is bounded as in equation (4), we may conclude that the acceleration along the path $\ddot{s}$ is bounded by

$$g_i(\alpha_I) \le \ddot{s} \le f_i(\alpha_I) \qquad \text{for i = I,.., 6n} \qquad (24)$$

where

$$f_i(\alpha_I) = \frac{\tau_i^* - \alpha_I Y_i - d_i}{E_i + \alpha_I K_i} \text{ and } g_i(\alpha_I) = \frac{\tau_i^{**} - \alpha_I Y_i - d_i}{E_i + \alpha_I K_i}$$

and $\tau_i^* = \begin{cases} \bar{\tau}_i^+ & \text{if } \rho_i > 0 \\ \bar{\tau}_i^- & \text{if } \rho_i < 0 \end{cases}$, $\tau_i^{**} = \begin{cases} \bar{\tau}_i^- & \text{if } \rho_i > 0 \\ \bar{\tau}_i^+ & \text{if } \rho_i < 0 \end{cases}$, and $\rho_i = (E_i + \alpha_I K_i)$.

The feasible region which satisfies all 6n constraints in equation (24) is given by

$$g(\alpha) \le \ddot{s} \le f(\alpha) \qquad (25)$$

where $f(\alpha) = \min \{f_i(\alpha_I) | \text{ i = 1, 2,.., 6n }\}$ and $g(\alpha) = max \{g_i(\alpha_I) | i = 1, 2,.., 6n\}$, and $a = [\alpha_1 \ \alpha_2 \ .. \ \alpha_n]^T$. Then the optimal distribution factor α^* may be obtained as the one which generates the maximum or minimum acceleration.

The below Lemma 4.1 shows that OLD approach is equivalent to the true time-optimal trajectory(LPP) approach given in the Section 3 under the certain conditions.

Lemma 4.1 : In CMMS with the rigid contact of the object, if the internal forces are specified and the motion of the object is purely translational or rotational about an axis throughout the path, the maximum and minimum accelerations obtained from the OLD algorithm are exactly the same as the ones obtained from the LPP approach.

Proof :

Since both approaches are designed to optimize the acceleration on the phase plane subject to the same constraints except the internal force constraint and the object dynamic equations, we need to show that these constraints are equivalent under the given conditions. If we assume the internal forces are specified, then the internal force constraint is the same for both approaches. In order to show that the dynamic equations of motion for the object used in the both approaches are equivalent, first notice that $\bar{F}_i = F_i$ for rigid contacts from (3) and (14). Rewriting the dynamic equation of the object in LPP approach from equation (6), we have

$$B F = \sum_{i=1}^{n} F_i = P . \tag{26}$$

The object dynamics for OLD approach is from equation (20) as follows.

$$F_i = \alpha_i P + F_{Ii} \quad \text{for all } i. \tag{27}$$

Since the sum of the internal forces is zero, it is obvious that equation (27) implies the equation (26). Now, we need to show that equation(26) implies the equation (27) under the given conditions. Notice that

$$\sum_{i=1}^{n} F_i = \sum_{i=1}^{n} F_{Mi} + \sum_{i=1}^{n} F_{Ii} = \sum_{i=1}^{n} F_{Mi} \tag{28}$$

where F_{Mi} is the motion force generated by the i-th manipulator. In the purely translational object motions, the linear motion forces do not have any normal or binormal directional forces as shown in the Proposition 3.1. Thus the linear motion forces must be along the positive tangential direction which is the direction of the required force, that is, $F_{Mi} = \alpha_i P$ where α_i is a positive constant. Thus equation (26) implies equation (27) in this case. In purely rotational motion about an axis, the angular motion forces(moments) must be only about the axis of rotation. Otherwise, angular internal forces(moments) are generated, which would add to the internal moments and contradict the condition of the specified internal forces. Therefore, equation (26) implies equation (27) in this case, too. Hence, we showed the equivalence of dynamic equations given by (26) and (27), under the given conditions. Q.E.D.

4.3. Finding A Load Distribution Factor For Two Manipulators Case

If two manipulators are used to manipulate an object, only one independent distribution parameter $a = \alpha_1$ is required to describe the load distribution as $\alpha_2 = (I - a)$. Thus we are only required to find the scalar optimal distribution factor a. Now the object dynamics of equation (20) becomes

$$\dot{F} = \begin{bmatrix} \alpha P \\ (1-\alpha)P \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ -\varepsilon_1 \end{bmatrix}. \tag{29}$$

Then the acceleration is bounded by

$$g(\alpha) \leq s \leq f(\alpha) \tag{30}$$

where $f(\alpha) = \min \{f_i(\alpha) | i = 1,2,..,12\}$ and $g(\alpha) = max \{g_i(\alpha) | i = 1,2,..,12\}$, and

$$f_i(\alpha) = \frac{\tau_i - \tilde{\alpha} Y_i - d_i}{E_i + \tilde{\alpha} K_i} \text{ and } g_i(\alpha) = \frac{\tau_i^{**} - \tilde{\alpha} Y_i - d_i}{E_i + \tilde{\alpha} K_i} \tag{31}$$

Here $\tilde{\alpha} = a$ for i = 1,.., 6 and $\tilde{\alpha} = (I-a)$ for $i = 7,..,12$, and τ_i, τ_i^*, τ_i^{**}, E_i, K_i, Y_i, and d_i are defined similarly as in equation (23) and (24). We will call $f(\alpha)$ the minimum acceleration curve and $g(\alpha)$ the maximum acceleration curve. Then, the optimal distribution factor a* may be obtained as follows.

$$a^* = \begin{cases} arg \max_{0 \leq \alpha \leq 1} f(\alpha) & \text{in the acceleration region} \\ arg \min_{0 \leq \alpha \leq 1} g(\alpha) & \text{in the deceleration region.} \end{cases} \tag{32}$$

In order to attain the trajectory with minimum traversal time, we need to drive the manipulator at the maximum possible acceleration and brake it at the maximum possible deceleration. From equation (32), we conclude that the maximum possible acceleration or deceleration can be determined by selecting the optimal distribution factor a constrained by $0 \leq a \leq 1$. In the following, we describe line search algorithms which can be used to find optimal load distribution factors, exploiting the properties of the functions in equation (31) for the two manipulators case. First we consider the below lemma.

Lemma 4.2 : For the functions of the form $f_i(\alpha) = c_i + \frac{k_i}{\alpha + \beta_i}$, where c_i, k_i and β_i are known constants, (a) there exist at most two intersections between two distinct functions of the form $f_i(\alpha)$, and (b) the function $f_i(\alpha)$ is a monotonic function for all $a \in (-\infty, -\beta_i)$. This is also true for all $a \in (-\beta_i, \infty)$.

Proof :

(a) Assume that there exist intersections between two functions $f_1(\alpha)$ and $f_2(\alpha)$. Then equating $f_1(\alpha) = f_2(\alpha)$ gives us,

$$(c_1 - c_2)\alpha^2 + (c_1\beta_2 + c_1\beta_1 + k_1 - c_2\beta_1 - c_2\beta_2 - k_2)\alpha + (c_1\beta_1 + k_1)\beta_2 - (c_2\beta_2 + k_2)\beta_1 = 0 .$$

If $c_1 \neq c_2$, then we obtain a as follows.

$$\alpha = \frac{-\beta \pm \sqrt{\beta^2 - 4(c_1 - c_2)[(c_1\beta_1 + k_1)\beta_2 - (c_2\beta_2 + k_2)\beta_1]}}{2(c_1 - c_2)} . \quad (33)$$

where $\beta = c_1\beta_2 + c_1\beta_1 + k_1 - c_2\beta_1 - c_2\beta_2 - k_2$

If $c_1 = c_2 = c$ and $k_1 + k_2$, then we have,

$$\alpha = \frac{k_2\beta_1 - k_1\beta_2}{k_1 - k_2} . \quad (34)$$

If $c_1 = c_2 = c$ and $k_1 = k_2 + 0$, then it implies that intersection occurs under following condition.

$$k_1(\beta_2 - \beta_1) = 0 . \quad (35)$$

This implies that $\beta_2 = \beta_1$, which contradicts the assumption that functionsf, and f_2 are distinct. If $c_1 = c_2 = c$ and $k_1 = k_2 = 0$, this implies $f_1 = f_2 = c$ which also contradicts the assumption that functionsf, and f_2 are distinct. Thus, we showed that there exist at most two intersections between f_1 and f_2, if they exist and the functions are distinctly different.

(b) Since $f_i(\alpha)$ is continuous over $a \in (-\infty, -\beta_i)$, it is enough to show that $f_i'(\alpha) \geq 0$ in order to prove that $f_i(\alpha)$ is an increasing function over $a \in (-\infty, -\beta_i)$. Similarly, $f_i'(\alpha) \leq 0$ for a decreasing function. Since $f_i'(\alpha) = \frac{-k_i}{(\alpha+\beta_i)^2}$, it is obvious that f_i is an increasing function for $k_i \leq 0$ or a decreasing function for $k_i \geq 0$ over $a \in (-\infty, -\beta_i)$. The proof of the second statement follows from the proof of the first statement. Q.E.D.

The above lemma suggests that we may employ line search algorithms to find the optimal distribution factor a*, since the functions of $f_i(\alpha)$ and $g_i(\alpha)$ in equation (31) have the form of $c_i + \frac{k_i}{\alpha + Pi}$. As an example, consider the four different functions of $f_i(\alpha)$ and $g_i(\alpha)$ of the form $c_i + \frac{k_i}{\alpha + Pi}$. As shown in Figure 3, two functions are increasing and the other two are decreasing over $a \in (0, 1)$. The minimum acceleration curve $f(\alpha)$ among the four functions, and the maximum acceleration curve $g(\alpha)$ among the four functions are shown in the figure. From equation (32), we can conclude that the optimal distribution factor $\alpha^* = arg \max_{0 \leq \alpha \leq 1} f(\alpha) = 0.22$ in the acceleration region. Similarly, $a^* = arg \min_{0 \leq \alpha \leq 1} g(\alpha) = 0.27$ in the deceleration region of the path.

If at least one of the functions is not continuous over $a \in [0, 1]$, i.e., if $a \in [0, 1]$ is a point of discontinuity, we need to divide the given line search into two admissible regions

of [0, $\bar{\alpha}$] and [$\bar{\alpha}$, 1]. After obtaining the optimal distribution factors $\alpha^*_1 \in [0, a]$ and $\alpha^*_2 \in [a,1]$, one can determine the optimal distribution factor as a^*= arg mar $\{f(\alpha^*_1), f(\alpha^*_2)\}$ in the acceleration region and a^*=arg min $\{g(\alpha^*_1), g(\alpha^*_2)\}$ in the deceleration region from (32). The Algorithm 4.1 constructs the piecewise minimum acceleration curve $f(\alpha)$ defined in equation (30) over a number of intervals.

Algorithm 4.1 (Minimum acceleration curve)

Step 1 ; Assuming that all the functions of f_i in (31) are continuous over (a,, α_f), let $\bar{f}_1 = \{ f_j \mid f_j(\alpha_o) = \min_k f_k(\alpha_o)$, k = 1, .., 12).

Step 2 ; For i = 1 , 2,.., 12, do Step 3, 4, and 5.

Step 3 ; Find all intersection points between $\bar{f}_i$ and all remaining functions $f_j \neq \bar{f}_i$, j = 1,..., 12. For the points $a \in [\alpha_{i-1}, \alpha_f]$ where intersection occur, let the point a which results in the minimum function value of $\bar{f}_i(\alpha)$ be α_i. Also let the crossing function at α_i with $\bar{f}_i$ be $\bar{f}_{i+1}$. If there exist more than one crossing function at α_i, then let the one with least slope be $\bar{f}_{i+1}$.

Step 4 ; If there is no $a \in [\alpha_{i-1}, \alpha_f]$ where an intersection occurs, let $\alpha_i = \alpha_f$ and stop.

Step5 ; Set $i = i + 1$.

The above algorithm generates the minimum acceleration curve of $f(\alpha)$ defined in (30) as follows.

$$f(\alpha) = \begin{cases} \bar{f}_1(\alpha) & \text{for } a \in [\alpha_o, \alpha_1 \text{ I} \\ \bar{f}_m(\alpha) & \text{for } \alpha \in [\alpha_{m-1}, \alpha_m \text{ I} \end{cases} \tag{36}$$

where $\alpha_m = \alpha_f$ and $\bar{f}_i = \{ f_j \mid f_j(\alpha) = \min_k f_k(\alpha)$, k = 1, .., 12 for a s $[\alpha_{i-1}$, a ,])

In the below, we will state some lemmas on the minimum acceleration curve which will be used in the proof of the optimal load distribution algorithms detailed later. Also, we define the candidate point as the intersection point between an increasing function and a decreasing function of (31) in the remainder of this paper. Figure 3 shows four such candidate points denoted as σ_i for i = 1, .., 4.

Lemma **4.3:** There exists one or no candidate point on the minimum acceleration curve $f(\alpha)$, generated by Algorithm 4.1.

Proof : (By Contradiction)

Suppose that there exist two candidate points on the minimum acceleration curve, namely, at α_1 and α_2. Without loss of generality, assume that $\alpha_1 < \alpha_2$. Let f_i^+ and f_i^- be the increasing and decreasing functions which intersect at α_i such that $f_i^+(\alpha_i) = f_i^-(\alpha_i)$. Consider the following two cases.

(i) If $f_1^-(\alpha_1) < f_2^-(\alpha_2)$: Since f_1^- is a decreasing function and $\alpha_1 < \alpha_2$, we have $f_1^-(\alpha_2) < f_1^-(\alpha_1)$. This implies that $f_1^-(\alpha_2) < f_2^-(\alpha_2)$. It means that the candidate point $f_2^-(\alpha_2)$ is not on the minimum acceleration curve.

(ii) If $f_1^-(\alpha_1) \geq f_2^-(\alpha_2)$: Since f_2^+ is an increasing function and $\alpha_1 < \alpha_2$, we have $f_2^+(\alpha_2) > f_2^+(\alpha_1)$. This implies that $f_1^-(\alpha_1) > f_2^+(\alpha_1)$ since $f_2^+(\alpha_2) = f_2^-(\alpha_2)$. It means that the candidate point $f_1^-(\alpha_1)$ is not on the minimum acceleration curve.

Q.E.D.

The previous lemma tells us that the minimum acceleration curve is made of either just decreasing functions, or just increasing functions, or at most there can be one intersection of an increasing function and a decreasing function. This leads us to the below lemma.

Lemma **4.4:** Assume that there exist n_z candidate points, which are the intersection points between an increasing function f_i^+ and a decreasing function f_i^- at $\alpha_i \in [a_i, \alpha_f]$ for $i = 1, 2, \ldots, n_z$. Let $\alpha_q = \arg\ min\ \{f_i^-(\alpha_i)$, for $i = 1, 2, \ldots, n, \}$. Assume that the point $f_q^-(\alpha_q)$ is not on the minimum acceleration curve $f(\alpha)$. Then there is no candidate point on the minimum acceleration curve.

Proof : (By Contradiction)

Suppose that there exists a candidate point on minimum acceleration curve $f(\alpha)$ at α_p, where $p \neq q$ and $p \in \{1, 2, \ldots, n_z\}$. Without loss of generality, we assume that $\alpha_p > \alpha_q$. Since f_q^- is a decreasing function at α_q, we have $f_q^-(\alpha_p) < f_q^-(\alpha_q)$. Also from the definition of α_q, we have $f_q^-(\alpha_q) = min\ \{f_i^-(\alpha_i)$, for $i = 1, 2, \ldots, n, \}$, and it means that $f_q^-(\alpha_q) \leq f_p^-(\alpha_p)$. Thus, we have $f_q^-(\alpha_p) < f_p^-(\alpha_p)$. It implies that the candidate point $f_p^-(\alpha_p)$ is not on the minimum acceleration curve. **Q.E.D.**

Therefore, from the above lemma we conclude that if the candidate point with the minimum acceleration is not on the minimum acceleration curve, there is no candidate point on the minimum acceleration curve. Now, consider the next lemma.

Lemma 4.5: Assume that there exists a candidate point at α_k on the minimum acceleration curve $f(\alpha)$, generated by Algorithm 4.1. Then $\alpha_k = arg\ max\ \{f(\alpha)$ for $a \in [a_o, \alpha_f]\}$.

Proof : (By Contradiction)

Suppose that $f(\alpha_j) = max\ \{ f(\alpha)$ for $\alpha \in [a_o, \alpha_f]\}$, where $a_j \in [a_o, \alpha_f]$ is any point other than α_k. Since we know that there exists at most one candidate point on the minimum acceleration curve from the Lemma 4.3, $f(\alpha_j)$ is not a candidate point. Without loss of generality, we assume that $\alpha_k < \alpha_j$. Let f_k^+ and f_k^- be the increasing and the decreasing functions at α_k. Then, we have $f_k^-(\alpha_j) < f_k^-(\alpha_k)$. Also we know that $f(\alpha_j) \leq f_k^-(\alpha_j)$ from the fact that $f(\alpha)$ is the minimum acceleration curve. Thus we conclude that $f(\alpha_j) < f_k^-(\alpha_k) = f(a_d)$. This shows that $a_j \neq arg\ max\ \{f(\alpha)$ for $a \in [a_o, \alpha_f]\}$. **Q.E.D.**

Based on the previous lemmas, we have devised two search schemes to find the optimal load distribution factor in the acceleration region(OLDA). The Algorithm 4.2 utilizes the minimum acceleration curve concept, while the Algorithm 4.3 employs the candidate points to determine the optimal load distribution factors. Later, we will compare the computational efficiencies of the two algorithms.

Algorithm 4.2 (OLDA using Minimum Acceleration Curve)

Step 1 ; Construct the minimum acceleration curve as in Algorithm 4.1 until it encounters a decreasing function $\bar{f}_i$ at α_{i-1}. Let α_{i-1} be the optimal load distribution factor a^* and stop.

Step 2 ; If the minimum acceleration curve does not encounter a decreasing function for $a \in [a_o, \alpha_f]$, let the optimal distribution factor $a^* = \alpha_f$ and stop.

Theorem 4.1 : The optimal load distribution factor obtained from the Algorithm 4.2 is optimal in the sense that $a^* = arg \max_{\alpha_o \leq \alpha \leq \alpha_f} f(\alpha)$ yields the maximum acceleration at a point in the phase plane, where $f(\alpha)$ is the minimum acceleration curve.

Proof :

First consider when Algorithm 4.2 encounters a decreasing function at $i = 1$, then the minimum acceleration curve is only composed of decreasing functions. This results in $a^* = a_o$. If the decreasing function is encountered when $i > 1$, then α_{i-1} is the candidate point on the minimum acceleration curve, and α_{i-1} is the optimal distribution factor from Lemma 4.5. If the minimum acceleration curve does not encounter any decreasing

functions for $a \in [a_o, \alpha_f]$, then the minimum acceleration curve is composed of only increasing functions. Thus, $a^* = \alpha_f$. **Q.E.D.**

Algorithm 4.3 (OLDA using Candidate Points)

Step 1 ; Assuming that the functions f_j, $j = 1, 2, .., 12$, in (31) are continuous over (a_o, α_f), find all candidate points (α_i, σ_i) where $\alpha_i \in [a_o, \alpha_f]$ and $\sigma_i = f_i^+(\alpha_i)$, $i = 1, 2, .., n_z$. Note that f_i^+ and f_i^- are the increasing and decreasing functions which intersect at α_i, thus $\sigma_i = f_i^+(\alpha_i) = f_i^-(\alpha_i)$.

Step 2 ; Let the candidate point with the lowest acceleration occur at α_N, that is, $\alpha_N = arg\ min\ \{\sigma_i, i = 1, 2, .., n_z\}$. If $\sigma_N = \min\{f_i(\alpha_N), i = 1, 2, .., 12\}$, go to Step 3. Otherwise, go to Step 4.

Step 3 ; Then the optimal load distribution factor be α_N and stop.

Step 4 ; Let $f_M = \{f_j \mid f_j(\alpha_o) = \min_k f_k(\alpha_o), k = 1, .., 12\}$. If f_M is decreasing over (a_o, α_f), then the optimal distribution factor $a^* = a_o$ and stop. If f_M is increasing over (a_o, α_f), then the optimal distribution factor $a^* = \alpha_f$ and stop.

Theorem 4.2 : The optimal load distribution factor obtained from the Algorithm 4.3 is optimal in the sense that $\alpha^* = arg \max_{\alpha_o \le \alpha \le \alpha_f} f(\alpha)$ yields the maximum acceleration at a point in the phase plane, where $f(\alpha)$ is the minimum acceleration curve.

Proof :

There are two cases to consider. If $\sigma_N = \min\{f_i(\alpha_N), i = 1, 2, .., 12\}$, then we know that σ_N is the point on the minimum acceleration curve. Since (α_N, σ_N) is a candidate point, it implies that $\alpha_N = arg\ max\{f(\alpha)$ for $a \bullet [a, \alpha_f]\}$ from Lemma 4.5. If $\sigma_N \neq \min\{f_i(\alpha_N), i = 1, 2, .., 12\}$, σ_N is not on the minimum acceleration curve, which implies that there is no candidate points on the minimum acceleration curve from Lemma 4.4. Therefore, the optimal load distribution factor is either α_o or α_f, depending on the minimum acceleration curve $f(\alpha)$. If $f(\alpha)$ is decreasing, $a^* = a_o$, and if $f(\alpha)$ is increasing, $a^* = \alpha_f$. **Q.E.D.**

We can compare the relative efficiency of the two algorithms 4.2 and 4.3 by estimating the number of calculations required to find the intersection points at one particular point in the path. Since the computation time required to select the minimum value is significantly less compared to time required to perform algebraic calculations, we will ignore the computation time required to find the minimum values in both algorithms. For simplicity,

we will assume that there exists only one intersection point between two different functions. This is a valid assumption since the intersections can occur at most in a pair as in (33), and both of them can be calculated simultaneously without significant increase in the computation time. Let the number of functions considered in Algorithm 4.2 and 4.3 be N. In Algorithm 4.2, the maximum number of calculation will consist of finding all the intersection points between two different functions, that is, $\binom{N}{2} = \frac{N^2 - N}{2}$ at each path point under consideration. The average number of calculations is $\binom{N}{2} - \binom{N/2}{2} = \frac{3N^2 - 2N}{8}$, if we assume that the the first decreasing function in the minimum acceleration curve occurs in the middle of the search in Algorithm 4.2. Assuming that the number of the increasing functions is N_i and the remaining functions are decreasing, the number of calculations required to find the candidate point in Algorithm 4.3 is $N_i (N - N_i)$. The maximum of number of calculations occurs at $N_i = \frac{N}{2}$, thus the maximum number of intersection calculation in Algorithm 4.3 is $\frac{N}{2} \cdot \frac{N}{2} = \frac{N^2}{4}$ at each trajectory sampling point. (Note that if N is an odd number the bound still remains the same, as the maximum number is $\frac{(N-1}{2} \left(\frac{1}{2} - \frac{N^2 - 1}{4} \right) \leq \frac{N^2}{4}$.) When N = 12, which is typical for two robots case, the maximum number of calculations required to find the intersection points is only 36 in the Algorithm 4.3, compared to the maximum of 66 calculations and the average of 51 calculations required in Algorithm 4.2. This clearly shows that the Algorithm 4.3 is more efficient than the Algorithm 4.2 in general.

In the below, we state OLDD(Optimal Load Distribution in Deceleration region) using the candidate points as in Algorithm 4.3 without proof. One can easily prove the optimality similarly from the proof of Theorem 4.2. Naturally, the proof of optimality for the Algorithm 4.4 requires the construction of the ***maximum acceleration curve*** and lemmas similar to those presented for the ***minimum acceleration curve.***

Algorithm 4.4 (OLDD using Candidate Points)

Step 1 ; Assuming that the functions g_j , j = 1, 2, .., 12, in (31) are continuous over (a, , α_f), find all candidate points (α_i , σ_i) where $\alpha_i \in$ [a,, α_f] and $\sigma_i = g_i^+(\alpha_i)$, i = 1, 2, .., n_z . Note that g_i^+ and g_i^- are the increasing and decreasing functions which intersect at α_i , thus $\sigma_i = g_i^+(\alpha_i) = g_i^-(\alpha_i)$.

Step 2 ; Let the candidate point with the maximum acceleration occur at α_N, that is, $\alpha_N = arg \max \{ \sigma_i , i = 1, 2, .., n, \}$. If $\sigma_N = max \{ g_i(\alpha_N), i = 1, 2, .., 12 \}$, go to Step 3. Otherwise, go to Step 4.

Step 3 ; Then the optimal load distribution factor be α_N and stop.

Step 4 ; Let $g_M = \{ g_j \mid \max_j g_j(\alpha_o), j = 1, 2, .., 12 \}$. If g_M is decreasing over (a,, α_f), then the optimal distribution factor $a^* = \alpha_f$ and stop. If g_M is increasing over (a,, α_f), then the optimal distribution factor $\alpha^* = a$, and stop.

V. SIMULATION RESULTS

Consider two planar robots of three degree-of-freedom manipulating a bar in the vertical plane as shown in Figure 4. We assume that the end-effectors grasp the object rigidly, so there is no relative movement between the end-effectors and the object. Let l_{ij} and m_{ij} be the length and mass, respectively, of the j-th link of the *i-th* robot. We are given $l_{11} = l_{12} = l_{21} = l_{22} = 1$ m , $l_{13} = l_{23} = 0.1m$ and $m_{11} = 5$ kg, $m_{12} = 4$ kg, $m_{13} = 0.5$ kg, $m_{21} = 5$ kg, $m_{22} = 4$ kg, and $m_{13} = 0.5$ kg. The mass of the object is 2 kg. We also assume that each link of the robot is a cylinder with radius $r_{ij} = 0.1$ m for all i, j. Then the inertia seen at joint j of the i-th robot is then given by

$$I_{ij} = \frac{1}{12} m_{ij} (3 r_{ij}^2 + l_{ij}^2) . \qquad (37)$$

The world coordinate reference frame is attached to the base of manipulator 1 at O_1. Thus O_1 has coordinates (0, 0) and O_2 is fixed at (0.7, 0) in the world coordinate frame. The trajectory of the object center of mass is a straight line from the initial position (0.35, 1.0) to the final position (0.65, 1.4). No rotations are specified along the trajectory.

5.1. Zero Internal Forces

In this example, we assume the two manipulators have the same torque capabilities. The input torque limits are $\tau_{11}^+ = 100$ Nm, $\tau_{12}^+ = 80$ Nm, $\tau_{13}^+ = 50$ Nm, $\tau_{21}^+ = 100$ Nm, $\tau_{22}^+ = 80$ Nm, $\tau_{23}^+ = 50$ Nm, and $\tau_{ij}^- = - \tau_{ij}^+$. We will assume the desired internal forces are zero in this example. From the LPP approach, we obtain the optimal phase plane curve which has one switching point at $s = 0.6$, as shown in Figure 5(a), and the resulting optimal traversal time is 339 *msec.* The computation time required to generate the trajectories on a Gould NP1 machine is 5.78 *sec* for LPP method. Since we used 108 phase plane steps to obtain the trajectory plan, each step takes the average computation time of 53 *msec.*

A traversal time of 339 msec is also obtained from the OLD algorithm as the internal forces were set to zero and no rotation of the object occurs during the motion. This result agrees with the conclusion of Lemma 4.1. The computation time required to generate the trajectory is, however, 0.8 *sec* in the OLD algorithm, much faster than 5.78 *sec* required in the LPP method. Here 79 phase plane steps were required, and each step took an average computation time of 10 msec. This is only 19 % of the computation time required by the LPP approach. The graph showing the optimal load distribution factor throughout the entire movement of the object is given in Figure 5(b). The input torque profiles are shown in Figure 5(c). In Figure 5(b), we notice that the object load is almost fully taken by the second robot in the deceleration region. We believe that this happens because the optimal load distribution depends on the configurations of the each manipulator. As we show in section 5.3, the load distribution also depends on the capacity of the manipulators.

5.2. Internal Force Constraints

In this example, we consider the same two robots used in the previous example. In order to show the sub-optimality of the OLD algorithm, we imposed some internal force constraints in the LPP algorithm. The internal force constraints we imposed on the object were -10 N $\leq f_{in}$ I 1 0 N and -10 Nm $\leq f_{iz} \leq$ 10 Nm, where f_{in} is the normal directional forces exerted by the *i-th* end-effector and f_{iz} is the angular moment along the z direction exerted by the *i-th* end-effector. The final traversal time we obtained from the LPP approach is 316 msec which is slightly faster than the traversal time 339 msec obtained from the example of Section 5.1. The phase plane curve for this trajectory is given in Figure 6(a). It shows that the switching occurs at s = 0.63. The end-effector forces along the tangential and normal direction and angular moment along the z direction are shown in Figure 6(b). The joint torque profiles are shown in Figure 6(c). As we can see in Figure 6(c), the utilization of the input torques is increased in the deceleration region, compared with Figure 5(c), which helped reduction of the traversal time compared to zero internal force case of Section 5.1. The computation time required to obtain this trajectory was 6.46 *sec*.

5.3. OLD Approach with Two Different Robots

In this example, we consider two robots with different joint torque capabilities. The input joint torque limits are τ_{11}^{+} = 100 Nm, τ_{12}^{+} = 80 Nm, τ_{13}^{+} = 50 Nm, τ_{21}^{+} = 70 Nm, τ_{22}^{+} = 50 Nm, τ_{23}^{+} = 30 Nm, and $\tau_{ij}^{-} = -\tau_{ij}^{+}$. The second robot now has a lower torque capacity, while the first robot has the same capacity as the one used in

previous simulations. We assumed that zero internal forces are specified in this example. As seen in Figure 7(b), the load distribution factor remained between 0.4 and 1.0 during the entire trajectory execution. Compared to the case in Section 5.1, the optimal load distribution has been shifted more to the robot with the larger capacity. The final traversal time we obtained is 393 *msec* which is slower than 339 *msec* of Section 5.1. The computation time required by the OLD algorithm is 0.64 *sec*. The phase plane curve of the optimal trajectory is shown in Figure 7(a), and the input torque profiles are given in Figure 7(c). It shows that the switching occurs at s = 0.65.

VI. DISCUSSIONS AND CONCLUSIONS

In this paper we presented two schemes to generate the time-optimal trajectory planning and sub-time-optimal trajectory planning for cooperative multi-manipulator system. The first approach used the linear programming problem(LPP) technique to obtain the maximum/minimum acceleration in the phase plane, and allowed us obtain the time-optimal trajectories. However, the computation time required by the mathematical programming method prohibits any possibility of using this algorithm in the real time applications. The second approach(OLD) which employs the algebraic search algorithms to determine the maximum/minimum acceleration in the phase plane generates slightly slower trajectory than the first approach does in most cases. However, it requires only a fraction of computation time compared to the LPP approach, which does provide the possibility of utilizing this method in the real time control problem.

In cases where the internal forces are specified and the motion of the object is purely translational or rotational throughout the path, the OLD approach produces exactly the same trajectories as the LPP approach does but only in a fraction of time compared to the LPP approach. This was proven in Lemma 4.1 and validated by the example in Section 5.1. This result is quite significant considering that the computation time has been reduced by one-fifth and the simulation in Section 5.2 shows that the constraining the internal forces in the certain range does not lead to much reduction of the traversal time. In many applications of CMMS, we will have to constrain the internal forces in a certain range, if not zero, in order to prevent the breakage or slipping the object during the motion.

REFERENCES

[1] S. B. Moon and S. Ahmad, "Time scaling of cooperative multi-robot trajectories," IEEE Trans. Syst., Man, Cybern., vol. 21, pp. 900-908, July 1991.

[2] S. B. Moon and S. Ahmad, "Time scaling of trajectories for cooperative multi-robot systems," in Proc. 29th IEEE Conf. Decision and Control, Dec. 1990, pp.1120-1125.

[3] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME* J. Dyn. Syst., *Meas.*, Contr., vol. 105, pp. 102-106, Mar. 1984.

[4] S. B. Moon and S. Ahmad, "Time optimal trajectories for cooperative multi-robot systems," in Proc. 29th IEEE Conf. Decision and Control, Dec. 1990, pp.1126-1127.

[5] J. E. Bobrow, J. M. McCarthy, and V. K. Chu, "Minimum-time trajectories for two robots holding the same workpiece," in Proc. 29th IEEE Conf. Decision and Control, Dec. 1990, pp.3102-3107.

[6] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," IEEE Trans. Automat. Contr., vol. AC-30, no. 6, pp. 531-541, June 1985.

[7] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," Int. J. Robotics Res., vol. 4, no. 3, pp. 3-17, 1985.

[8] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in Proc. I988 IEEE Conf. Robotics and Automation, pp. 350-356.

[9] I. D. Walker, R. A. Freeman, and S. I. Marcus, "Internal object loading for multiple cooperating robot manipulators," in Proc. 1989 IEEE Conf. Robotics and Automation, pp. 606-611.

[10] C. R. Rao, Linear Statistical Inference and Its Applications, John Wiley & Son, 1973.

[11] S. Ahmad, "Control of cooperative multiple flexible joint robots," to appear in IEEE Trans. Syst., Man, Cybern., March 1993.

[12] J. E. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," IEEE Trans. Robotics and Automat., vol. RA-5, no. 1, pp. 118-124, 1989.

[13] N. H. McClamroch and H. Huang, "Dynamics of a closed chain manipulator," in Proc. *1985* American Controls Conf., pp. 50-54.

[14] J. Y. S. Luh and Y. F. Zheng, "Constrained relations between two coordinated industrial robots for motion control," Znt. J. Robotics Res., vol. 6, no. 3, pp. 3-17, 1987.

[15] D. E. Orin and S. Y. Oh, "Control of force distribution in robotic mechanisms containing closed kinematic chains," *ASME* J. Dyn. Syst., *Meas.,* Contr., vol. 102, pp. 134-141, June 1981.

[16] Y. F. Zheng and J. Y. S. Luh, "Optimal load distribution for two industrial robots handling a single object," *ASME* J. Dyn. Syst., *Meas.,* Contr., vol. 111, pp. 232-237, June 1989.

[17] C. R. Carignan and D. L. Akin, "Optimal force distribution for payload positioning using a planar dual-arm robot," *ASME* J. Dyn. Syst., *Meas.,* Contr., vol. 111, pp. 205-210, June 1989.

[18] Y. Chen, "Structure of the time-optimal control law for multiple arms handling a common object along specified paths," ZEEE Trans. Automat. Contr., vol. AC-37, no. 10, pp. 1648-1652, Oct. 1992.

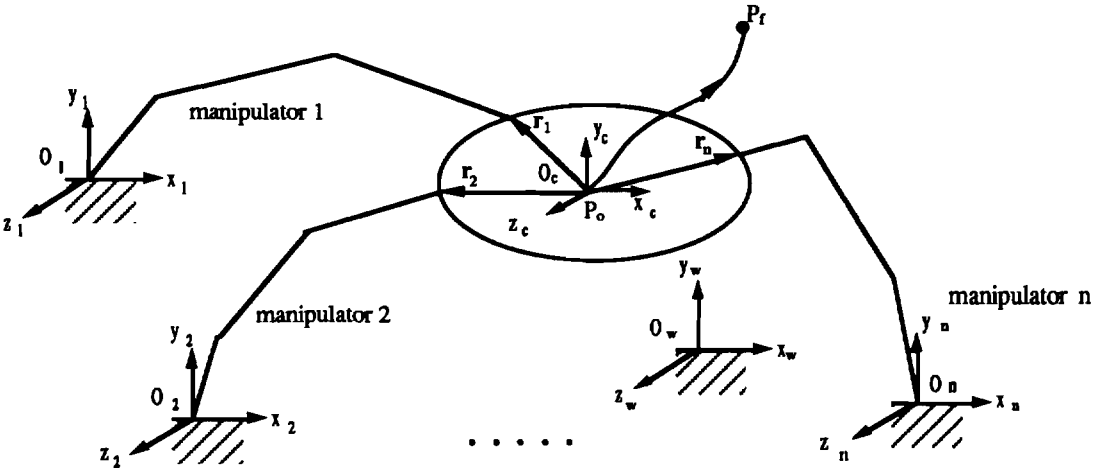

Figure 1. Schematic diagram for cooperative multi-manipulatorsystem.

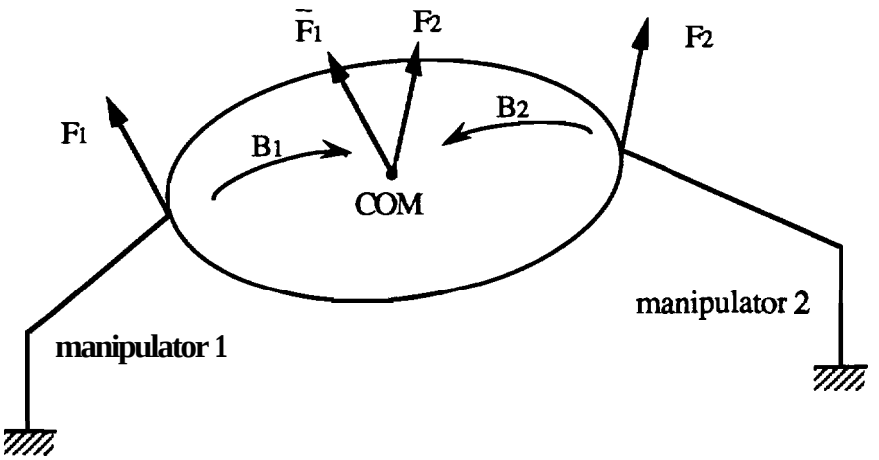

Figure 2. Resultant forces expressed in the different coordinate frames.

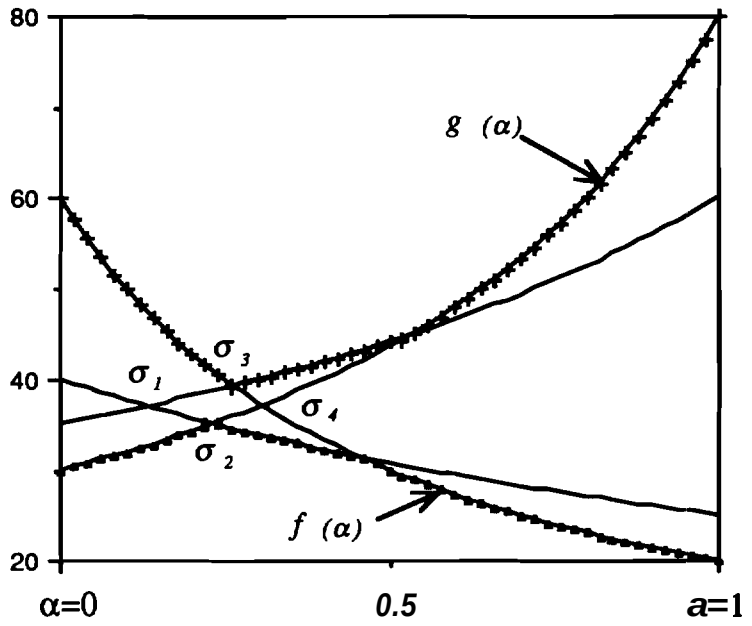

Figure 3. Four different functions of $f_i(\alpha)$ and $g_i(\alpha)$ of the form $c_i + \frac{k_i}{\alpha + Pi}$.

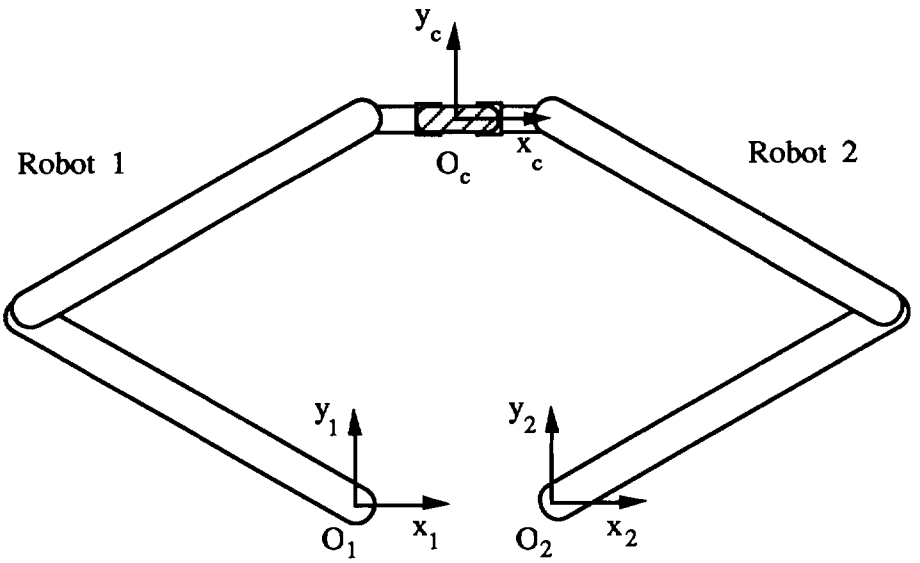

Figure 4. Two three degree-of-freedom manipulators of CMMS.

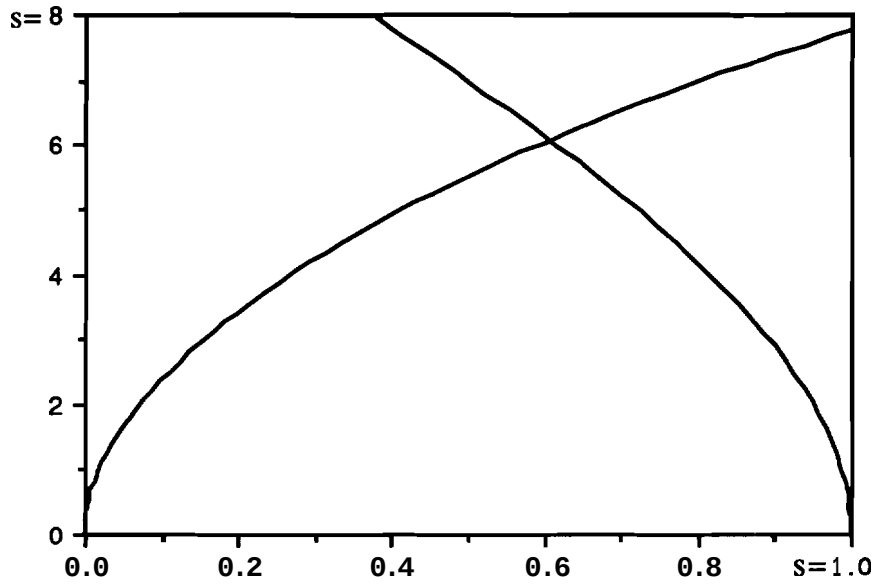

Figure 5.(a) The phase plane curves for zero internal forces case.

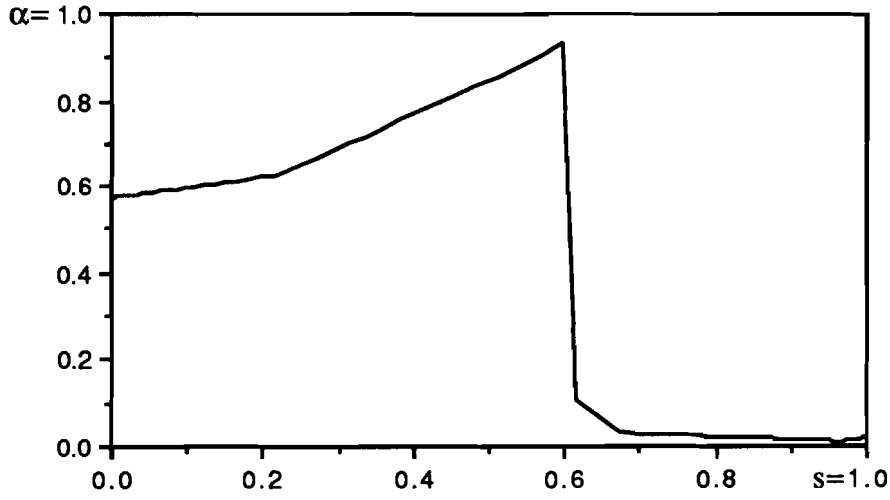

Figure 5.(b) The optimal load distribution factor for zero internal forces case.

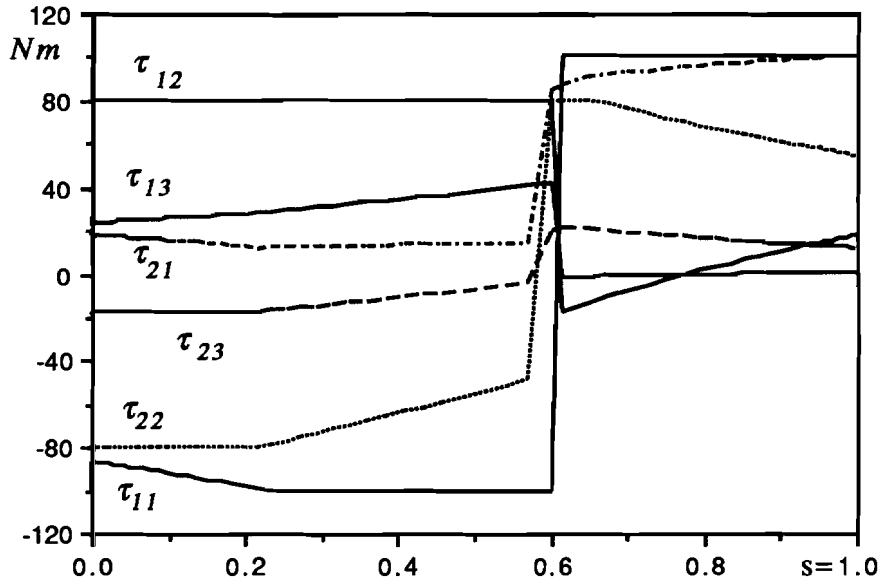

Figure 5.(c) The torque profiles for zero internal forces case.

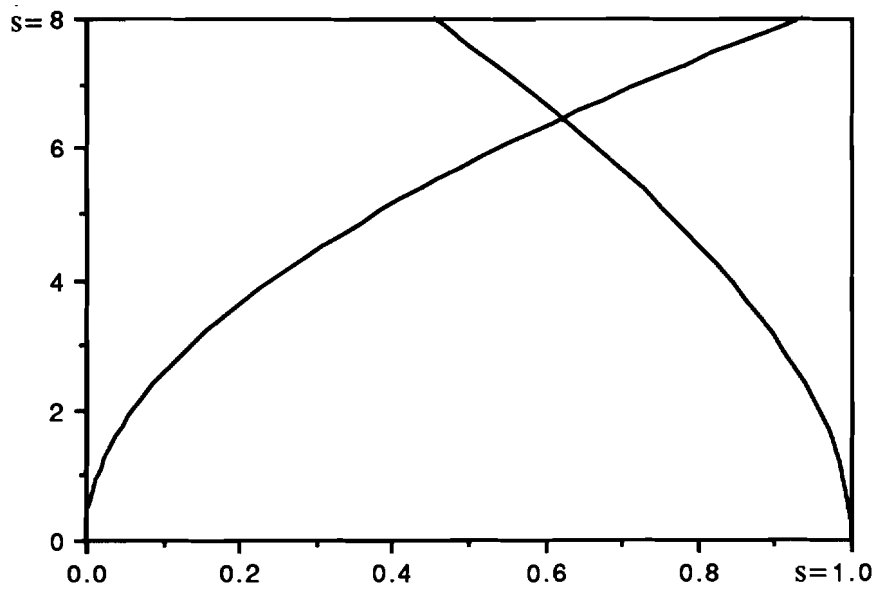

Figure 6.(a) The phase plane curves for internal forces constraints case.

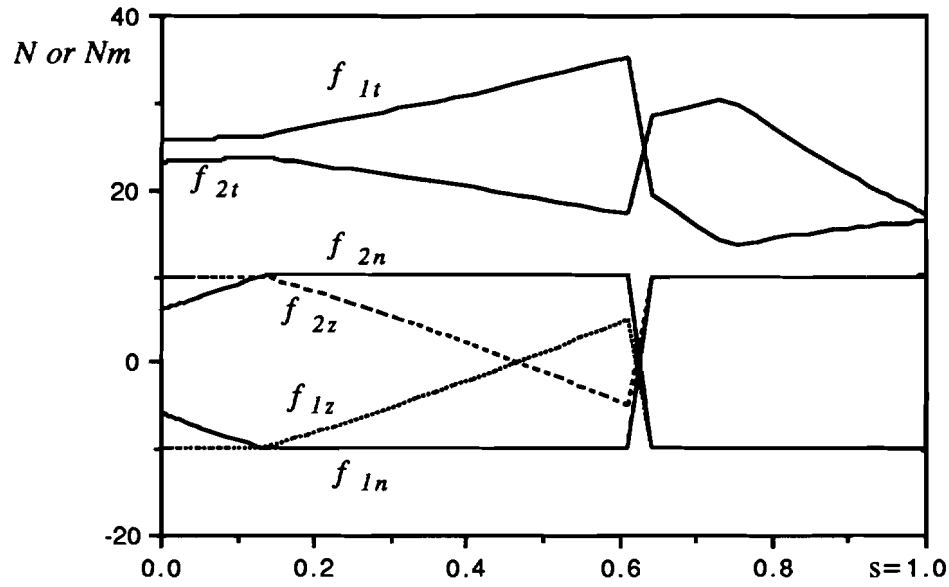

Figure 6.(b) The force profiles for internal forces constraints case.

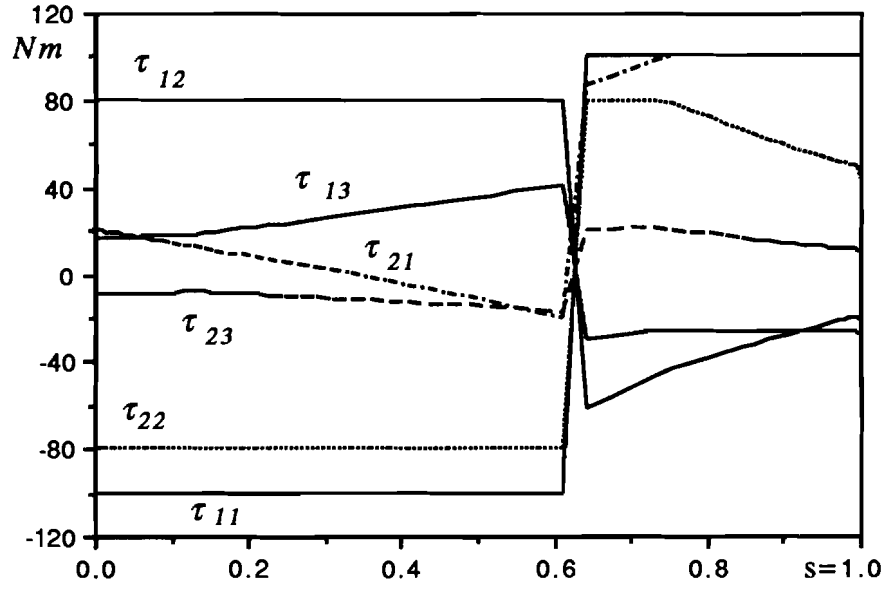

Figure 6.(c) The torque profiles for internal forces constraints case.

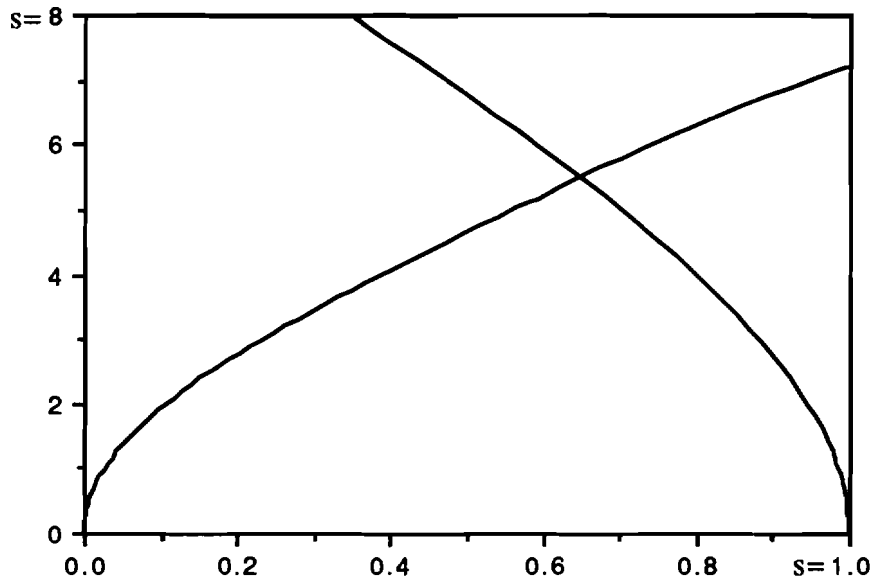

Figure 7.(a) The phase plane curves for optimal load distribution approach with two different robots.

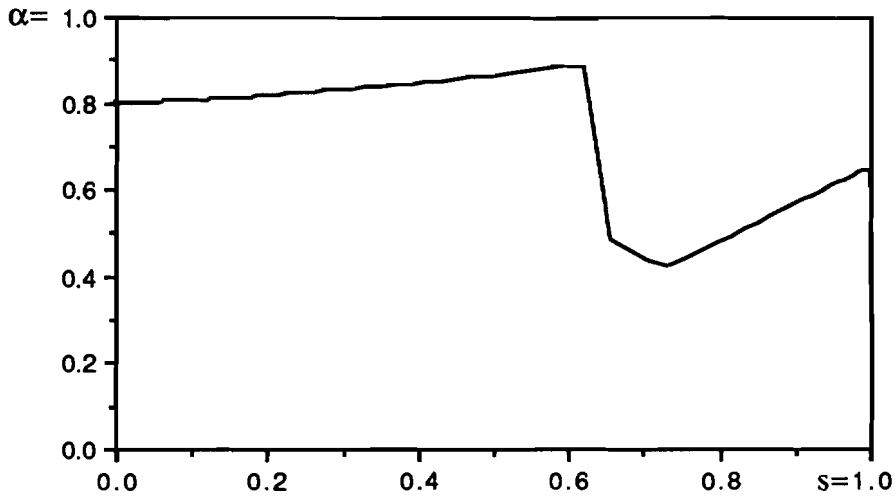

Figure 7.(b) The optimal load distribution factor for optimal load distribution approach with two different robots.

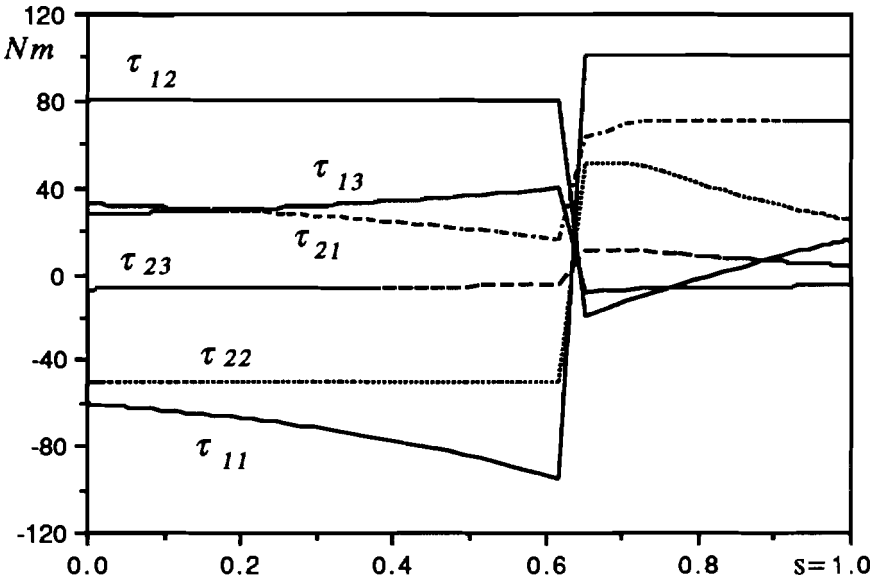

Figure 7.(c) The torque profiles for optimal load dismbution approach with two different robots.

Manuscript received

The authors are with School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

FIGURE CAPTIONS

Figure 1. Schematic diagram for cooperative multi-manipulator system.

Figure 2. Resultant forces expressed in the different coordinate frames.

Figure 3. Four different functions of $f_i(\alpha)$ and $g_i(\alpha)$ of the form $c_i + \frac{k_i}{\alpha + \beta_i}$.

Figure 4. Two three degree-of-freedom manipulators of CMMS.

Figure 5.(a) The phase plane curves for zero internal forces case.

Figure 5.(b) The optimal load distribution factor for zero internal forces case.

Figure 5.(c) The toque profiles for zero internal forces case.

Figure 6.(a) The phase plane curves for internal forces constraints case.

Figure 6.(b) The force profiles for internal forces constraints case.

Figure 6.(c) The toque profiles for internal forces constraints case.

Figure 7.(a) The phase plane curves for optimal load distribution approach with two different robots.

Figure 7.(b) The optimal load distribution factor for optimal load distribution approach with two different robots.

Figure 7.(c) The toque profiles for optimal load distribution approach with two different robots.