Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Time-Scaled Coordination of Multiple Manipulators

Srinivas Akella
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180
Email:sakella@cs.rpi.edu

Jufeng Peng
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, New York 12180
Email: pengj@rpi.edu

***Abstract*— Coordinating multiple manipulators in a shared workspace while considering their dynamics is an important problem. This problem of collision-free coordination arises in assembly, materials transfer, and welding workcells. Previous approaches that considered robot dynamics have typically been restricted to coordinating just two or three manipulators, even when their paths are specified. We address the task of coordinating the motions of multiple manipulators when either their trajectories or their paths are given. By exploiting a fundamental time scaling law for manipulators based on their dynamics, we identify sufficient conditions for collision-free coordination of the robots when the velocity profiles can be uniformly time-scaled and the robot start times can be varied. We describe an approach that develops mixed integer programming formulations of these problems, where the time scaling factors are linear variables, to automatically minimize completion time. This method can potentially coordinate the motions of many manipulators.**

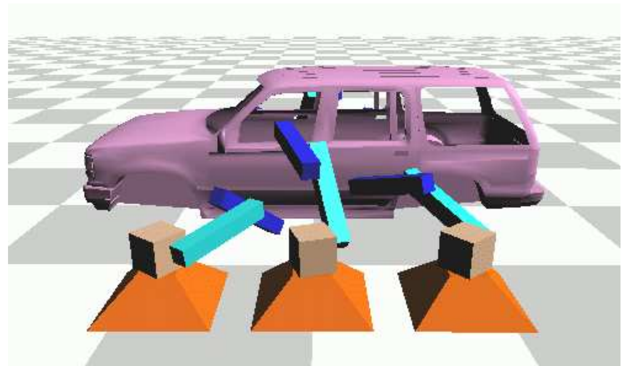

Fig. 1. Depiction of an example welding task that requires coordination of multiple manipulators.

I. Introduction

Time-optimal and collision-free coordination of multiple manipulator robots in a shared workspace while considering their dynamics is an important open problem. There are several applications that involve this coordination task. Consider scheduling the motions of multiple robots in a welding or assembly workcell to minimize the cycle time (Figure 1). Since the robots have overlapping workspaces, we must coordinate their motions to avoid collisions between robots. Additionally, the actuator torque limits restrict how fast or slow a manipulator may move along a path. The dynamics constraints make this a challenging problem, and previous work ([1], [2], [3], [4], [5], [6]) has typically been restricted to coordinating just two or three robots, even when their paths are specified.

We focus on coordinating the motions of multiple manipulators, while considering dynamics, when their trajectories or paths are provided. By trajectory, we mean the geometric specification of the path along with the timing of the robot's motion along the path (that is, path and velocity). Hollerbach [7] identified a fundamental time-scaling law for manipulator dynamics that can be used to determine the range of feasible trajectory speedups and slowdowns, without needing to recompute the dynamics. We exploit this time-scaling law, which decouples the path and timing along the path, to generate time-warped trajectories to coordinate multiple manipulators. We assume that the velocity profile of the given trajectory of each individual robot may be uniformly time scaled so its velocity profile shape is maintained, and the robot start times can be changed. As such, this paper builds on and generalizes our previous work on coordination of multiple robots with specified trajectories (Akella and Hutchinson [8]), which did not consider dynamics explicitly.

The specific problem that we consider in this paper is: *Given a set of manipulator robots with specified paths and velocity profiles on those paths, find a set of uniform time-scaled parameterizations for these paths such that the completion time for the set of robots is minimized, dynamics constraints along the paths are satisfied, and no collisions occur.*

Our goal is to find near minimum-time collision-free robot coordinations by computing the scaling factors by which the robot velocity profiles are uniformly time-scaled, as well as the robot start times. We identify sufficient conditions for collision-free coordination of multiple robots and formulate the task as an optimization problem using a mixed integer linear programming (MILP) formulation. The uniform time scaling law for manipulators identified by Hollerbach [7] leads to a formulation where the time scaling factors are linear variables. The primary advantage of this approach is its ability to potentially handle many robots, each with several degrees of freedom, while considering their dynamics.

The paper is organized as follows. Section II briefly discusses related work. Section III describes our previous work on coordinating multiple robots with specified trajectories when only their start times could be changed. Section IV outlines Hollerbach's time scaling law. Section V presents the main results of this paper. Using the time scaling law and sufficient conditions for collision-free time-scaled motion, we present a mixed integer programming formulation for time-scaled coordination of multiple robots with input trajectories. Section VI describes preliminary results from our implementation of the approach. Section VII outlines directions for future work.

II. Related Work

There are two main bodies of related work, with some overlap. One focuses on the coordination of multiple robots, typically without considering robot dynamics. The other focuses on trajectory optimization for a single robot while considering robot dynamics.

Multiple robot coordination: Motion planning for multiple robots is a broad research area (see [9] for an overview). The most general problem is to have each robot move from its initial to its goal configuration, while avoiding collisions with static obstacles or with other robots. Hopcroft, Schwartz, and Sharir [10] showed that even a simplified two-dimensional case of the problem is PSPACE-hard. Svestka and Overmars [11] developed a probabilistic roadmap (PRM) planner for path coordination of multiple car-like robots. Recently Sanchez and Latombe [12] used a single-query, bidirectional, lazy PRM variant for coordinated path planning of multiple robot arms, without considering dynamics.

A slightly more constrained version of the problem is obtained when all but one of the robots have specified trajectories. This is essentially the problem of planning a path for a single robot among moving obstacles, which has been treated by Reif and Sharir [13] and Kant and Zucker [14]. One can generalize this problem to obtain a heuristic solution to the problem of planning the motions of multiple robots. Erdmann and Lozano-Perez [15] assign priorities to robots and sequentially search for collision-free paths for the robots, in order of priority, in the configuration-time space. Fiorini and Shiller [16] developed a velocity space method for avoiding moving obstacles.

If the problem is further constrained so that the paths of the robots are specified, one obtains a path coordination problem, where the robot dynamics are neglected. O'Donnell and Lozano-Perez [17] developed a method for path coordination of two robots. LaValle and Hutchinson also addressed a similar problem in [18], where each robot was constrained to remain on a specified configuration space roadmap during its motion. Simeon, Leroy, and Laumond [19] perform path coordination for a very large number of car-like robots (over a hundred robots in some examples), where robots with intersecting paths can be partitioned into smaller sets (of about ten robots). Akella and Hutchinson [8] recently developed an MILP formulation for the trajectory coordination of large numbers of robots by only changing robot start times.

Trajectory planning for a single robot: There is a large body of work on the time optimal control of a single manipulator with dynamics constraints. Bobrow, Dubowsky, and Gibson [20], and Shin and McKay [21] developed algorithms to generate the time-optimal velocity profile of a manipulator moving along a specified path. Subsequently Pfeiffer and Johanni [22], Slotine and Yang ([23]), and Shiller and Lu [24] refined these algorithms.

Trajectory planning directly in the $2n$-dimensional state space that considers both kinematic and dynamic constraints is called kinodynamic planning. Sahar and Hollerbach [25] and later Shiller and Dubowsky [26] developed algorithms for global near minimum-time trajectory generation for a manipulator with dynamics and actuator constraints using grid-based search spaces. Donald et al. [27] developed a polynomial time approximation algorithm to generate near time-optimal trajectories for a robot that satisfy kinematic and dynamic constraints. Donald and Xavier [28] extended this work to robot manipulators. Fraichard [29] described a trajectory planner for a car-like robot with dynamics constraints moving along a given path among moving obstacles. Recent work has focused on randomized kinodynamic planning, including the use of rapidly exploring random trees (Lavalle and Kuffner [30]) and probabilistic roadmaps (Hsu et al. [31]).

Multiple robot coordination with dynamics: Work on trajectory coordination with dynamics has focused almost exclusively on dual robot systems (Shin and Bien [2], Chang, Chung and Bien [3], Bien and Lee [4], Chang, Chung and Lee [6]). Lee and Lee [1] considered the effects of delays and velocity changes on motion time. Shin and Zheng [5] showed that for a two-robot system, generating time-optimal trajectories for each robot independently and then delaying the start time of one of the robots leads to a minimal finish time under certain assumptions. Moon and Ahmad [32] studied the time scaling of cooperative multi-robot trajectories with force interactions between the manipulators. They use linear programming to find the scaling constant range and quadratic programming to find the minimum energy coordination. However, they did not address collision avoidance of the manipulators.

Peng and Akella [33] developed an MILP formulation to coordinate the motions of many robots with specified paths while considering dynamics constraints; however the robots have simple double integrator dynamics. The RRT approach [30] is capable of generating collision-free trajectories for multiple robots. However it does not explicitly provide a method to optimize the coordination of the robots.

III. Background: Coordination of Multiple Robots with Specified Trajectories

In this section, we summarize our previous work (Akella and Hutchinson [8]) on coordinating multiple robots with specified trajectories. That work considered the *trajectory coordination problem:* Given a set of robots with specified trajectories, find the starting times for the robots such that the completion time for the set of robots is minimized and no collisions occur. It permitted only the robot start times to be modified, and did not consider robot dynamics explicitly.

A. Trajectories and Their Parameterizations

We first briefly review paths and their parameterizations. We denote the i^{th} robot by $\mathcal{A}_i$, a configuration space by $\mathcal{C}$, and a configuration by $\mathbf{q} \in \mathcal{C}$. By *path* we mean the geometric specification of a curve in configuration space

$$\gamma : \zeta \in [0,1] \mapsto \gamma(\zeta) = \mathbf{q} \in \mathcal{C}$$

A differentiable function τ given by

$$\tau : t \in [0,T] \mapsto \tau(t) = \zeta \in [0,1]$$

with $\tau(0) = 0$ and $\tau(T) = 1$ is a reparameterization of the path γ. Here t is a time variable, and T is some constant such that all robots will have completed their tasks by time T. A path together with a time parameterization defines a *trajectory*. To simplify notation, we often denote a trajectory as $\gamma(t)$.

For this problem, robot velocities are specified a priori by specifying an original parameterization for γ, say τ, such that the time derivatives of τ provide the desired velocity profile. Thus, any reparameterization τ' that gives the desired velocity profile will be such that, for any ζ value along the path, the time derivatives of τ' and τ agree. All such reparameterizations are obtained by merely changing the start time of task execution. That is,

$$\tau_i'(t) = \begin{cases} \tau_i(t - t_i^{start}) & : \quad t \geq t_i^{start} \\ 0 & : \quad t < t_i^{start} \end{cases}, \qquad (1)$$

in which $t_i^{start} \geq 0$ is the time at which robot $\mathcal{A}_i$ begins its motion, and τ_i is the originally specified parameterization.

B. Collision Zones: Geometry

We now describe a geometric characterization for collisions that can occur between robots, and identify sufficient conditions for collision-free coordination. Consider the set of points at which the i^{th} robot, $\mathcal{A}_i$, could possibly collide with the j^{th} robot, $\mathcal{A}_j$. For a specific value of ζ_i, the subset of the workspace that is occupied by the i^{th} robot is denoted by $\mathcal{A}_i(\gamma_i(\zeta_i))$. A collision between two robots corresponds to the situation in which $\mathcal{A}_i(\gamma_i(\zeta_i)) \cap \mathcal{A}_j(\gamma_j(\zeta_j)) \neq \emptyset$. For the i^{th} robot, we denote by $\mathcal{PB}_{ij}$ the set of values of ζ_i such that when robot $\mathcal{A}_i$ is at configuration $\gamma_i(\zeta_i)$ there exists a configuration of another robot, $\mathcal{A}_j$, such that the two robots collide:

$$\mathcal{PB}_{ij} = \{\zeta_i \mid \exists\, \zeta_j \in [0,1] \; s.t.\; \mathcal{A}_i(\gamma_i(\zeta_i)) \cap \mathcal{A}_j(\gamma_j(\zeta_j)) \neq \emptyset\}$$

$\mathcal{PB}_{ij}$ is the set of all points on the path of robot $\mathcal{A}_i$ at which $\mathcal{A}_i$ could collide with $\mathcal{A}_j$, and can be represented as a set of intervals

$$\mathcal{PB}_{ij} = \{[\zeta_{is}^k, \zeta_{if}^k]\} \qquad (2)$$

where each interval is a *collision segment*, and the subscripts s and f refer to the start and finish of the kth collision segment. For each collision segment in $\mathcal{PB}_{ij}$ there is at least one collision segment in $\mathcal{PB}_{ji}$ that could result in collision of the two robots. We will refer to these corresponding pairs of collision segments as *collision zones*, denoted by $\mathcal{PI}_{ij}$. The set of collision zones can be represented as a set of ordered pairs of intervals:

$$\mathcal{PI}_{ij} = \{< [\zeta_{is}^k, \zeta_{if}^k], [\zeta_{js}^k, \zeta_{jf}^k] >\}. \qquad (3)$$

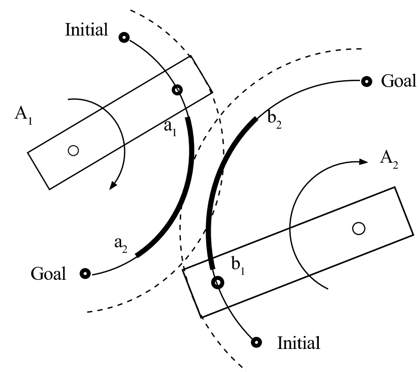

Fig. 2. Two single-link manipulators, with paths and collision zone (in bold) indicated.

Conceptually, intersection regions of the swept volumes of pairs of robots give the collision zones. Figure 2 is an example of two single-link manipulators with paths that overlap in a collision zone. Here $\mathcal{PB}_{12} = \{[a_1, a_2]\}$ and $\mathcal{PB}_{21} = \{[b_1, b_2]\}$. Collisions can occur only when $\zeta_1 \in [a_1, a_2]$ and $\zeta_2 \in [b_1, b_2]$. Thus, $\mathcal{PI}_{12} = \{< [a_1, a_2], [b_1, b_2] >\}$.

C. Collision Zones: Timing

The collision zones describe the geometry of possible collisions, but for scheduling the robots, we must describe the timing of the collisions. For a specified parameterization τ_i, the set of times at which it is possible that robot $\mathcal{A}_i$ could collide with robot $\mathcal{A}_j$ is given by:

$$\begin{aligned} \mathcal{TB}_{ij}(\tau_i) &= \{t \mid \mathcal{A}_i(\gamma_i(\tau_i(t))) \cap \mathcal{A}_j(\gamma_j(\zeta_j)) \neq \emptyset, \\ &\qquad \text{for some } \zeta_j \in [0,1], i \neq j\} \\ &= \tau_i^{-1}(\mathcal{PB}_{ij}). \end{aligned}$$

As with $\mathcal{PB}_{ij}$, the set $\mathcal{TB}_{ij}(\tau_i)$ can be represented by a set of intervals, indexed by superscript k, the endpoints of which are obtained by applying the inverse parameterization (i.e., τ_i^{-1}) to the endpoints of the intervals of $\mathcal{PB}_{ij}$ given in (2):

$$\mathcal{TB}_{ij}(\tau_i) = \{[\tau_i^{-1}(\zeta_{is}^k), \tau_i^{-1}(\zeta_{if}^k)]\} \qquad (4)$$

We refer to each interval as a *collision-time interval*. For notational convenience, we introduce the variables T_{is}^k and T_{if}^k given by $T_{is}^k = \tau_i{}^{-1}(\zeta_{is}^k)$ and $T_{if}^k = \tau_i{}^{-1}(\zeta_{if}^k)$, where T_{is}^k (respectively T_{if}^k) denotes the time at which $\mathcal{A}_i$ starts (resp. finishes) traversing its k^{th} collision segment if $t_i^{start} = 0$. Although the notation T_{is}^k is ambiguous about the particular other robot that is involved in the collision, the context will make clear which other robot is involved. See Figure 3 for a graphical illustration of these quantities.

As with collision zones, there is a natural correspondence between collision-time intervals in $\mathcal{TB}_{ij}$ and $\mathcal{TB}_{ji}$, and we refer to these pairs as *collision-time interval pairs*. For the two robots $\mathcal{A}_i$ and $\mathcal{A}_j$, we denote the set of all collision-time interval pairs by $\mathcal{CI}_{ij}$. The interval pairs in $\mathcal{CI}_{ij}(\tau_i, \tau_j)$ can be determined by applying the appropriate inverse parameterization to the endpoints of the collision zone intervals in $\mathcal{PI}_{ij}$.

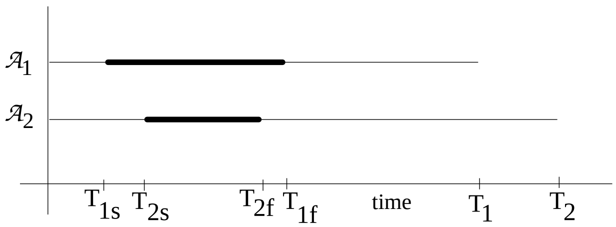

Fig. 3. Timelines for robots $\mathcal{A}_1$ and $\mathcal{A}_2$. The bold lines correspond to the collision-time intervals for the robots.

We represent $\mathcal{CI}_{ij}$ as a set of ordered pairs of intervals

$$\mathcal{CI}_{ij}(\tau_i, \tau_j) \quad = \quad \{< [T^k_{is}, T^k_{if}], [T^k_{js}, T^k_{jf}] >\} \qquad (5)$$

where the first interval $[T^k_{is}, T^k_{if}]$ of the kth pair corresponds to robot $\mathcal{A}_i$ and the second interval $[T^k_{js}, T^k_{jf}]$ corresponds to robot $\mathcal{A}_j$. During the time interval $[T^k_{is}, T^k_{if}]$, $\mathcal{A}_i$ is in a specific collision zone and $\mathcal{A}_j$ is in this collision zone during time interval $[T^k_{js}, T^k_{jf}]$. If $[T^k_{is}, T^k_{if}]$ and $[T^k_{js}, T^k_{jf}]$ do not overlap, then the two robots cannot be in the k^{th} collision zone simultaneously, and therefore no collision will occur in this collision zone. This observation forms the basis for the sufficient conditions given in Section III-D.

When the parameterizations are restricted to those that only delay the robot start times, we have parameterizations of the form

$$\tau'(t + t^{start}) = \zeta = \tau(t), \qquad (6)$$

for each value of $\zeta \in [0, 1]$. Inverting the parameterizations τ' and τ we obtain

$$\tau'^{-1}(\zeta) = \tau^{-1}(\zeta) + t^{start}. \qquad (7)$$

Using this notation, we can write $\mathcal{CI}_{ij}(\tau'_i, \tau'_j)$ as

$$\begin{aligned}\mathcal{CI}_{ij}(\tau'_i, \tau'_j) = \{ < &[T^k_{is} + t^{start}_i, T^k_{if} + t^{start}_i], \\ &[T^k_{js} + t^{start}_j, T^k_{jf} + t^{start}_j] >\}.\end{aligned}$$

D. Sufficient Conditions for Collision-free Scheduling

To prevent collisions between two robots $\mathcal{A}_i$ and $\mathcal{A}_j$, it is sufficient to ensure that the times at which $\mathcal{A}_i$ could collide with $\mathcal{A}_j$ do not coincide with the times at which $\mathcal{A}_j$ could collide with $\mathcal{A}_i$. This can be assured if the two robots are not simultaneously in any collision zone belonging to $\mathcal{PI}_{ij}$. This amounts to ensuring that there is no overlap between the two intervals of any collision-time interval pair for the two robots. If $[T^k_{is} + t^{start}_i, T^k_{if} + t^{start}_i] \cap [T^k_{js} + t^{start}_j, T^k_{jf} + t^{start}_j] = \emptyset$ for every collision-time interval pair in $\mathcal{CI}_{ij}(\tau'_i, \tau'_j)$, then no collision can occur. This sufficient condition leads to an optimization problem:

Given a set of robots with specified trajectories, find the starting times for the robots such that the completion time for the set of robots is minimized and no two intervals of any collision-time interval pair overlap.

In Section III-E, we outline a mixed integer linear program that solves this optimization problem. The sufficient condition is clearly not a necessary condition. For example, in a follow-the-leader situation where the robots move in the same direction along their paths in the collision zone, the follower robot is delayed unduly as it waits for the leader to exit the collision zone before it enters the collision zone. For now, we note that the sufficient conditions provide a conservative strategy that guarantees that no collision occurs between the two robots. See [8] for the necessary conditions that provide the minimum time collision-free schedule.

E. Coordination of Multiple Robots with Specified Trajectories

We developed a mixed integer linear programming (MILP) formulation for coordinating the motions of multiple robots with specified trajectories, where only the start times can be modified (Akella and Hutchinson [8]). The start time for robot $\mathcal{A}_i$ is t^{start}_i, which is to be computed, and T_i is the motion time required for robot $\mathcal{A}_i$ to traverse its entire trajectory when starting at time $t^{start}_i = 0$. The maximum time for robot $\mathcal{A}_i$ to complete its motion, $t^{start}_i + T_i$, is its *completion time*. The completion time for the set of robots, $t_{complete}$, is the time when the last robot completes its task. Therefore $t_{complete} \geq t^{start}_i + T_i$ for all robots.

Consider trajectory coordination of a pair of robots $\mathcal{A}_i$ and $\mathcal{A}_j$ with specified trajectories. For each robot, identify its kth collision zone and compute the time interval during which it is in its collision zone. The collision-time interval $[T^k_{is}, T^k_{if}]$ of robot $\mathcal{A}_i$, where subscripts s and f indicate start and finish times respectively, indicates when robot $\mathcal{A}_j$ can collide with it. The collision-time interval $[T^k_{js}, T^k_{jf}]$ of robot $\mathcal{A}_j$ is similarly computed. Ensuring the robots are not in their collision zones at the same time yields a disjunctive "or" constraint that can be converted to an equivalent pair of constraints using an integer zero-one variable δ_{ijk} and M, a large positive number ([34]). Here M can be chosen greater than $\sum_{i=1}^{N_{robots}} T_i$. When robot $\mathcal{A}_i$ enters the collision zone first, the constraint $t^{start}_i + T^k_{if} < t^{start}_j + T^k_{js}$ holds and $\delta_{ijk} = 0$, and when robot $\mathcal{A}_j$ enters the collision zone first, the constraint $t^{start}_j + T^k_{jf} < t^{start}_i + T^k_{is}$ holds and $\delta_{ijk} = 1$.

Let N_{robots} be the number of robots. Let N_{ij} denote the number of collision-time interval pairs for robots $\mathcal{A}_i$ and $\mathcal{A}_j$, i.e., $N_{ij} = |\mathcal{CI}_{ij}|$. We wish to minimize the completion time while ensuring the robots are not in their shared collision zones at the same time. The MILP formulation that gives a collision-free solution for this coordination task is:

$$\begin{aligned}
&\text{Minimize } t_{complete} \\
&\text{subject to} \\
&\quad t_{complete} - t^{start}_i - T_i \geq 0, \quad 1 \leq i \leq N_{robots} \\
&\quad t^{start}_i + T^k_{if} - t^{start}_j - T^k_{js} - M\delta_{ijk} \leq 0, \\
&\quad t^{start}_j + T^k_{jf} - t^{start}_i - T^k_{is} - M(1 - \delta_{ijk}) \leq 0, \\
&\qquad\quad \text{for all } < [T^k_{is}, T^k_{if}], [T^k_{js}, T^k_{jf}] > \in \mathcal{CI}_{ij}, \\
&\qquad\quad \text{for } 1 \leq i < j \leq N_{robots} \\
&\quad \delta_{ijk} \in \{0, 1\}, \quad 1 \leq i < j \leq N_{robots}, \quad 1 \leq k \leq N_{ij} \\
&\quad t^{start}_i \geq 0, \quad 1 \leq i \leq N_{robots}.
\end{aligned}$$

IV. Dynamic Time-Scaling Law for a Manipulator

To incorporate dynamics into the coordination approach of the previous section, we use Hollerbach's result on uniform time scaling of a manipulator trajectory [7]. This result enables a decoupling of the manipulator path and its timing law, without recomputing dynamics. Assume a manipulator arm with n joints whose dynamics is described by:

$$u(t) = M(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + G(q(t))$$

where u is the $n \times 1$ vector of input joint torques, q is the $n \times 1$ vector of joint generalized coordinates, $M(q)$ is the $n \times n$ inertia matrix, $C(q, \dot{q})$ is the $n \times n$ Coriolis matrix, and $G(q)$ is the $n \times 1$ gravitational torque vector. The actuator torques must lie between their maximum and minimum values u_{max} and u_{min}, which are assumed constant. These actuator torque limits restrict how fast or slow a manipulator may move along a trajectory. The time-scaling law identifies valid trajectory modifications that utilize the available torque range and avoid actuator torque limit violations, without requiring inverse dynamics recomputation.

Consider the acceleration and velocity dependent torque component u_a that remains after removing the position dependent gravitational torque vector:

$$u_a(t) = u(t) - G(q) = M(q)\ddot{q} + C(q, \dot{q})\dot{q}$$

Let $u_a^s(t)$ be the scaled acceleration and velocity dependent torque component required after time scaling. The simplest case of time scaling is the case of constant time scaling, where the time scaling function is $r(t) = ct$ for $c > 0$. The time scaling c assumes the values $1 > c > 0$ for slower motions and $\infty > c > 1$ for faster motions. Hollerbach showed that if the trajectory is uniformly time scaled by a constant time scaling value c, the acceleration and velocity dependent torque u_a is scaled by a value of c^2. That is, when the travel time T^{travel} along a path is uniformly scaled by a value c to become T^{travel}/c, the torque required for executing the new trajectory is scaled by a value of c^2, up to the gravitational torque contribution. The scaled torque $u_a^s(t)$ is given by

$$u_a^s(t) = c^2 u_a(ct)$$

and the total torque after scaling $u^s(t)$ is given by

$$u^s(t) = c^2 u_a(ct) + G(q(ct)).$$

For a given input trajectory, we can compute the interval of allowed time scaling values for each joint and intersect these intervals to obtain an interval of allowed time scaling values $[c^{min}, c^{max}]$ for the manipulator based on the torque limits of its actuators, as described in [7]. We can therefore identify allowable speeds of movement for a given trajectory without dynamics recomputation.

V. Time-Scaled Coordination of Multiple Manipulators

We now combine the results described in the previous two sections to tackle the *Time-Scaled Coordination Problem: Given a set of manipulator robots with input paths and velocity profiles on those paths, find a set of uniform time-scaled parameterizations for these paths such that the completion time for the set of robots is minimized, dynamics constraints along the paths are satisfied, and no collisions occur.*

A. Assumptions

We make the following assumptions to generate a collision-free time-scaled coordination of the robot trajectories:

1. Trajectories for the manipulators are initially provided.
2. The only moving obstacles in the workspace are the robots.
3. Each robot does not collide with any other robot when it is at its start or goal configurations.
4. Each robot moves monotonically along its path, that is, the robot does not back up along its path.
5. The robot motions are sampled at sufficient resolution so that no collisions occur during the motion between successive collision-free configurations.
6. The dynamics of each robot is known accurately.

B. Time-Scaled Coordination of Multiple Single-Link Robots

Consider time-scaled coordination of multiple manipulators given their trajectories. For clarity, we first discuss only single-link robots. From the actuator torque limits and input trajectory, we can compute an interval of allowed time scaling values $[c_i^{min}, c_i^{max}]$ for each manipulator $\mathcal{A}_i$. When the trajectory of robot $\mathcal{A}_i$ is uniformly time scaled by a value c_i, a travel time T_i^{travel} in the input trajectory is scaled by a factor s_i to become $s_i T_i^{travel}$, where $s_i = 1/c_i$. When $s_i > 1$, the motion of robot $\mathcal{A}_i$ is slowed down (time dilation), and when $s_i < 1$, the motion of robot $\mathcal{A}_i$ is sped up (time contraction). Clearly $s_i^{min} = 1/c_i^{max}$ and $s_i^{max} = 1/c_i^{min}$.

Now multiple robots, pairs of which may have multiple collision zones, must be coordinated. As before, the binary variable δ_{ijk} is defined to be 0 if robot $\mathcal{A}_i$ enters its k^{th} collision zone with robot $\mathcal{A}_j$ before robot $\mathcal{A}_j$ and to be 1 if robot $\mathcal{A}_j$ enters its corresponding k^{th} collision zone before robot $\mathcal{A}_i$. A valid value for M is $M > \sum_{i=1}^{N_{robots}} s_i^{max} T_i$. The MILP formulation to coordinate the time scaled motions of the single-link robots is therefore:

Minimize $t_{complete}$
subject to

$$\begin{aligned}
&t_{complete} - t_i^{start} - s_i T_i \geq 0, \quad 1 \leq i \leq N_{robots} \\
&t_i^{start} + s_i T_{if}^k - t_j^{start} - s_j T_{js}^k - M\delta_{ijk} \leq 0 \\
&t_j^{start} + s_j T_{jf}^k - t_i^{start} - s_i T_{is}^k - M(1 - \delta_{ijk}) \leq 0, \\
&\qquad \text{for all } < [T_{is}^k, T_{if}^k], [T_{js}^k, T_{jf}^k] > \in \mathcal{CI}_{ij}, \\
&\qquad \text{for } 1 \leq i < j \leq N_{robots} \\
&\delta_{ijk} \in \{0, 1\}, \quad 1 \leq i < j \leq N_{robots}, \quad 1 \leq k \leq N_{ij} \\
&t_i^{start} \geq 0, \quad 1 \leq i \leq N_{robots} \\
&s_i^{max} \geq s_i \geq s_i^{min}, \quad 1 \leq i \leq N_{robots}.
\end{aligned}$$

Solving this MILP gives the time scaling factors s_i and start times t_i^{start} for the individual robots. The resulting solution is guaranteed to be a collision-free time-scaled trajectory coordination strategy for all the robots. Figure 4 shows the timelines for two robots with multiple collision intervals, and Figure 5 shows the collision-free time-scaled coordination of the timelines of the robots. The completion time constraints and disjunctive collision-time interval constraints are necessary for only those robots that may collide. Note that the MILP always has a feasible solution; just move the robots in sequence with only one robot in motion at any given instant. Also, the above formulation is not guaranteed to give the true global optimum since it uses sufficient conditions for collision avoidance; this is a conservative strategy that does not permit follow-the-leader motions in a collision zone.

The time-scaled coordination of multiple manipulators is NP-hard, and follows directly from the complexity of coordinating multiple robots with specified trajectories ([8]).

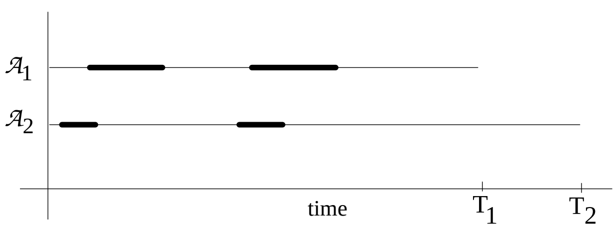

Fig. 4. Timelines for robots $\mathcal{A}_1$ and $\mathcal{A}_2$ with multiple collision intervals.

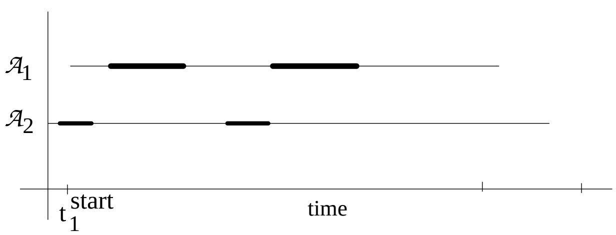

Fig. 5. Collision-free time-scaled timelines for robots $\mathcal{A}_1$ and $\mathcal{A}_2$, with $\mathcal{A}_1$ being delayed at its start and $\mathcal{A}_2$ having its timeline shrunk.

C. Time-Scaled Coordination of Multi-Link Robots

To coordinate manipulator robots with multiple links, we consider motions of the individual links. An articulated robot $\mathcal{A}_i$ consists of a set of links $\{\mathcal{A}_{il}\}$, where link $\mathcal{A}_{il}$ belongs to robot $\mathcal{A}_i$. For a specified trajectory, the motions of links of an articulated robot are separated by constant time offsets. Let $\mathcal{A}_{il}$ begin moving time T_{il}^o after the first moving link of $\mathcal{A}_i$ starts moving. That is, $t_{il}^{start} = t_i^{start} + T_{il}^o$ where t_{il}^{start} is the start time of link $\mathcal{A}_{il}$. The completion time for $\mathcal{A}_{il}$ is $t_i^{start} + T_{il}^o + T_{il}$, where T_{il} is the motion time of $\mathcal{A}_{il}$. Note that the start time and motion time of a link may depend on the start and motion times of links that precede it in the kinematic chain.

When an articulated robot's trajectory is time scaled, every link $\mathcal{A}_{il}$ of robot $\mathcal{A}_i$ has the same time scaling factor s_i. Therefore $t_{il}^{start} = t_i^{start} + s_i T_{il}^o$ where t_i^{start} is the start time of robot $\mathcal{A}_i$. The minimum and maximum scaling factors s_i^{min} and s_i^{max} for each robot are determined as described previously, and s_i lies in the range $[s_i^{min}, s_i^{max}]$. Let N_{links} be the total number of robot links. Then the MILP formulation for time-scaled coordination of a set of articulated robots is:

Minimize $t_{complete}$
subject to

$$
\begin{aligned}
&t_{complete} - t_i^{start} - s_i T_{il}^o - s_i T_{il} \geq 0, \quad 1 \leq i \leq N_{links} \\
&t_i^{start} + s_i T_{il}^o + s_i T_{if}^{kl} - t_j^{start} - s_j T_{jm}^o - s_j T_{js}^{km} \\
&\quad -M\delta_{ijk}^{lm} \leq 0, \\
&t_j^{start} + s_j T_{jm}^o + s_j T_{jf}^{km} - t_i^{start} - s_i T_{il}^o - s_i T_{is}^{kl} \\
&\quad -M(1 - \delta_{ijk}^{lm}) \leq 0, \\
&\qquad \text{for all } < [T_{is}^{kl}, T_{if}^{kl}], [T_{js}^{km}, T_{jf}^{km}] > \in \mathcal{CI}_{ij}, \\
&\qquad \text{for } \ 1 \leq i < j \leq N_{robots} \text{ and } i \neq j, \\
&\delta_{ijk}^{lm} \in \{0, 1\}, \ \ 1 \leq i < j \leq N_{robots}, \ \ 1 \leq k \leq N_{ij} \\
&t_i^{start} \geq 0, \ \ 1 \leq i \leq N_{robots}, \\
&s_i^{max} \geq s_i \geq s_i^{min}, \ \ 1 \leq i \leq N_{robots}.
\end{aligned}
$$

The completion time constraints are necessary for all links of a robot that can potentially have a collision. The collision-time interval constraints are necessary for only those robots that have one or more links involved in a potential collision.

D. Specifying Sequencing Constraints

In certain tasks, it may be necessary for one robot to complete a particular operation or reach a certain point before another robot performs a subsequent operation. This can occur in sequenced assembly tasks, or in welding workcells where the primary welds must be completed before secondary welds. Consider the constraint that $\mathcal{A}_i$ has to reach $\mathbf{q}_i$ before $\mathcal{A}_j$ reaches $\mathbf{q}_j$. For the unmodified trajectories, let the time taken for $\mathcal{A}_i$ to reach $\mathbf{q}_i$ be T_{q_i} and for $\mathcal{A}_j$ to reach $\mathbf{q}_j$ be T_{q_j}. The time-scaled sequencing constraint can then be written as $t_i^{start} + s_i T_{q_i} < t_j^{start} + s_j T_{q_j}$. Such constraints for multiple robots can be easily added to the formulation.

E. Time-Scaled Coordination Given Input Paths

Consider the time-scaled coordination task when only the paths for the individual robots are specified. The time-optimal coordination of multiple manipulator robots when only the paths are specified is an open problem. We outline a method to provide feasible and potentially near-optimal coordinated schedules that respect the dynamics constraints. First generate the time-optimal trajectory for each individual robot along its path, following the methods of Bobrow, Dubowsky, and Gibson [20] and Shin and McKay [21]. Now the problem can be transformed to the problem of time-scaling the individual time-optimal trajectories. This will result in a feasible solution that respects the dynamics constraints. Further, it provides an upper bound on the time-optimal schedule for the robots given their paths. Note that since at least one of the joint actuators is always saturated along the time-optimal velocity profile, each robot's motion may only be slowed down or remain unchanged.

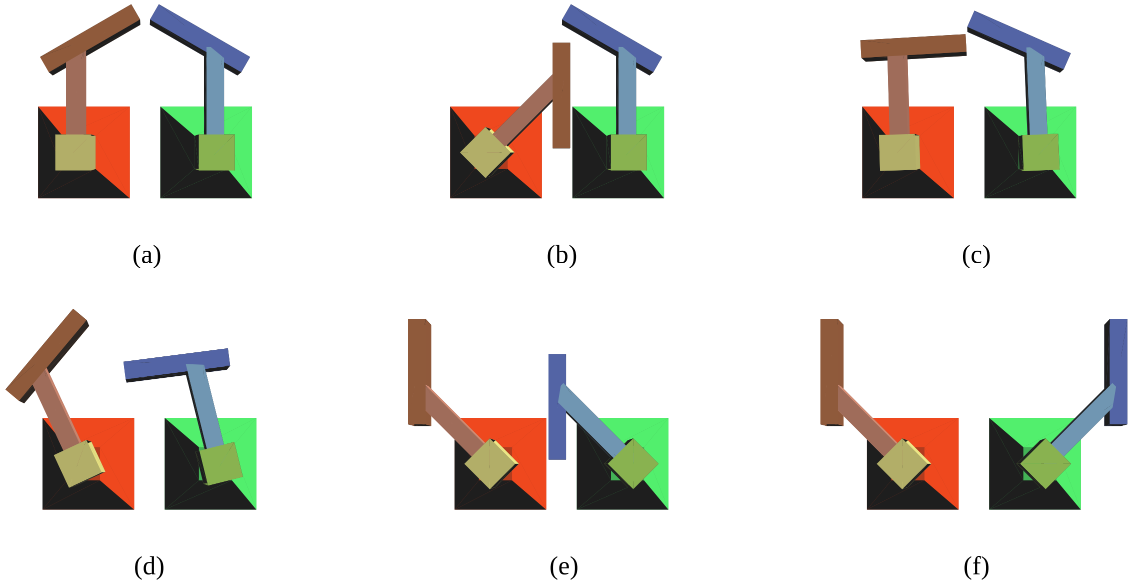

Fig. 6. Snapshots of time-scaled coordination of two manipulators, each with two revolute joints. (View from left to right, top row first.)

VI. Implementation

We have implemented time-scaled coordination of manipulators with input trajectories, and demonstrated the approach on up to 6 manipulators (Figures 1 and 6). We compute the collision zones using the PQP collision detection package (Larsen et al. [35]). We generate the MILP formulation from the collision-time interval pairs, and solve it using the CPLEX [36] optimization package. We currently select the permitted ranges for the time scaling factors. Valid ranges may also be computed directly from the robots' dynamics and trajectories. As Table I shows, solving time-scaled coordination problems takes the same order of magnitude of time as solving trajectory coordination problems with no scaling. Example animations may be seen at `www.cs.rpi.edu/˜sakella/timescale/`.

TABLE I

Sample run times on an IBM RS6000 for the MILP formulation, computed for three time scaling factor ranges. Range I has no scaling, range II= $[1.001, 1.1]$, and range III= $[0.9, 1.1]$.

Num. of robots	Num. of links	Num. of collision zones	MILP I (secs)	MILP II (secs)	MILP III (secs)
2	3	3	0.04	0.033	0.0367
2	4	14	0.06	0.07	0.0567
6	18	24	0.177	0.28	0.3467

We have also experimented with time-scaled coordination of up to 12 polyhedral robots modeled as double integrators (since their dynamics are similar to single-link manipulators and Cartesian manipulators). Sometimes the best solutions do not always have all scaling factors at their minimum values (i.e., not all robots move as fast as they can). Further, even when the scaling range only permits robots to slow down, the completion time is sometimes an improvement over the case with no scaling.

VII. Conclusion

We have developed an optimization formulation to enable the uniform time-scaled coordination of multiple manipulators with input trajectories or input paths. The principal advantage of our MILP formulation is that it potentially permits the collision-free coordination of a large number of manipulators, while considering their dynamics. The problem complexity depends primarily on the number of collision zones, and to a lesser extent on the number of robots and their number of degrees of freedom. Although the problem of time-scaled trajectory coordination of multiple robots is NP-hard, the availability of efficient integer programming solvers makes this approach practical for industrial automation problems, which typically involve less than twenty manipulators.

This work represents a step towards time-optimal coordination of multiple manipulators. There are several directions for future work. The uniform time-scaling formulation provides an upper bound on the true optimal coordination of a set of manipulators with specified paths. Incorporating less conservative conditions for collision avoidance will improve solution quality. Analyzing the gap between the time-scaled coordination described here and the true time optimal coordination is important, as is developing techniques for generating the time-optimal coordinated trajectories subject to dynamics constraints. Extending the time-scaled coordination approach to manipulators with elastic joints, based on recent work by De Luca and Farina [37], and exploring extensions to other robot systems would broaden the scope of this approach. Examining alternative solutions generated by the MILP can help optimize different actuator performance requirements and improve robot and workcell design. Finally, extensions to on-line coordination of robots (as in [38]) that involve timing uncertainties would be useful.

ACKNOWLEDGMENTS

This work was supported in part by RPI and by NSF under CAREER Award No. IIS-0093233. Mark Moll's question sparked this paper. Thanks to Seth Hutchinson for earlier collaboration, and Prasad Akella and Charles Wampler for helpful discussions. Andrew Andkjar implemented animation software that interfaced with PQP.

REFERENCES

[1] B. H. Lee and C. S. G. Lee, "Collision-free motion planning of two robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 1, pp. 21–32, January/February 1987.

[2] Y. Shin and Z. Bien, "Collision-free trajectory planning for two robot arms," *Robotica*, vol. 7, pp. 205–212, 1989.

[3] C. Chang, M. J. Chung, and Z. Bien, "Collision-free motion planning for two articulated robot arms using minimum distance functions," *Robotica*, vol. 8, pp. 137–144, 1990.

[4] Z. Bien and J. Lee, "A minimum time trajectory planning method for two robots," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 414–418, June 1992.

[5] K. G. Shin and Q. Zheng, "Minimum-time collision-free trajectory planning for dual robot systems," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 641–644, Oct. 1992.

[6] C. Chang, M. J. Chung, and B. H. Lee, "Collision avoidance of two robot manipulators by minimum delay time," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 517–522, Mar. 1994.

[7] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 1, pp. 102–106, Mar. 1984.

[8] S. Akella and S. Hutchinson, "Coordinating the motions of multiple robots with specified trajectories," in *IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 624–631.

[9] J.-C. Latombe, *Robot Motion Planning.* Norwell, MA: Kluwer Academic Publishers, 1991.

[10] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the "warehouseman's problem"," *International Journal of Robotics Research*, vol. 3, no. 4, pp. 76–88, 1984.

[11] P. Švestka and M. Overmars, "Coordinated path planning for multiple robots," *Robotics and Autonomous Systems*, vol. 23, no. 3, pp. 125–152, Apr. 1998.

[12] G. Sanchez and J. Latombe, "On delaying collision checking in PRM planning — application to multi-robot coordination," *International Journal of Robotics Research*, vol. 21, no. 1, pp. 5–26, Jan. 2002.

[13] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," in *Proceedings of the 26th Annual Symposium on the Foundations of Computer Science*, Portland, Oregon, Oct. 1985, pp. 144–154.

[14] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, Fall 1986.

[15] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 4, pp. 477–521, 1987.

[16] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, July 1998.

[17] P. A. O'Donnell and T. Lozano-Perez, "Deadlock-free and collision-free coordination of two robot manipulators," in *IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 484–489.

[18] S. M. LaValle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, Dec. 1998.

[19] T. Simeon, S. Leroy, and J.-P. Laumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, Feb. 2002.

[20] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, Fall 1985.

[21] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 6, pp. 531–541, June 1985.

[22] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pp. 115–123, Apr. 1987.

[23] J.-J. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 118–124, Feb. 1989.

[24] Z. Shiller and H. H. Lu, "Computation of path constrained time optimal motions with dynamic singularities," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 34–40, Mar. 1992.

[25] G. Sahar and J. M. Hollerbach, "Planning of minimum-time trajectories for robot arms," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 97–140, Fall 1986.

[26] Z. Shiller and S. Dubowsky, "On computing the global time-optimal motion of robotic manipulators in the presence of obstacles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 785–797, Dec. 1991.

[27] B. R. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM*, vol. 40, no. 5, pp. 1048–1066, Nov. 1993.

[28] B. R. Donald and P. Xavier, "Provably good approximation algorithms for optimal kinodynamic planning for Cartesian robots and open chain manipulators," *Algorithmica*, vol. 14, no. 6, pp. 480–530, Dec. 1995.

[29] T. Fraichard, "Trajectory planning in dynamic workspace: a 'state-time space' approach," *Advanced Robotics*, vol. 13, no. 1, pp. 75–94, 1999.

[30] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–401, May 2001.

[31] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, Mar. 2002.

[32] S. B. Moon and S. Ahmad, "Time scaling of cooperative multi-robot trajectories," in *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, May 1990, pp. 506–511.

[33] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," in *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, Eds. Heidelberg, Germany: Springer-Verlag, 2003, pp. 221–237.

[34] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization.* New York: John Wiley and Sons, 1988.

[35] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha, "Fast distance queries using rectangular swept sphere volumes," in *IEEE International Conference on Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 3719–3726.

[36] *CPLEX 6.6 User's Manual*, ILOG, Inc., Incline Village, NV, 1999.

[37] A. De Luca and R. Farina, "Dynamic scaling of trajectories for robots with elastic joints," in *IEEE International Conference on Robotics and Automation*, Washington, D.C., May 2002, pp. 2346–2442.

[38] T. Y. Li and J. C. Latombe, "On-line manipulation planning for two robot arms in a dynamic environment," *International Journal of Robotics Research*, vol. 16, no. 2, pp. 144–167, 1997.