to appear in *International Journal of Robotics Research*

Collision-Free Trajectory Planning for a 3-DOF Robot with a Passive Joint

Kevin M. Lynch* Naoji Shiroma† Hirohiko Arai‡ Kazuo Tanie‡

*Mechanical Engineering Department
Northwestern University
Evanston, IL 60208 USA

†Inst. of Eng. Mechanics and Systems
University of Tsukuba
1-1-1 Tennodai, Tsukuba, 305-8573 Japan

‡Robotics Department
Mechanical Engineering Laboratory
Namiki 1-2, Tsukuba, 305-8564 Japan

August 26, 1999; revised August 25, 2000

Keywords: underactuated manipulator, trajectory planning, locally controllable nonholonomic system, passive joint, decoupling motions

Abstract

This paper studies motion planning from one zero velocity state to another for a three-joint robot in a horizontal plane with a passive revolute third joint. Such a robot is small-time locally controllable on an open subset of its zero velocity section, allowing it to follow any path in this subset arbitrarily closely. However, some paths are "preferred" by the dynamics of the manipulator in that they can be followed at higher speeds. In this paper we describe a computationally efficient trajectory planner which finds fast collision-free trajectories among obstacles. We are able to decouple the problem of planning feasible trajectories in the robot's six-dimensional state space into the computationally simpler problems of planning paths in the three-dimensional configuration space and time scaling the paths according to the manipulator dynamics. This decoupling is made possible by the existence of velocity directions, fixed in the passive link frame, which can be executed at arbitrary speeds. We have demonstrated the motion planner on an experimental underactuated manipulator. To our knowledge, it is the first implementation of a collision-free motion planning algorithm for a manipulator subject to a second-order nonholonomic constraint.

1 Introduction

Two goals of recent work on underactuated robotic manipulators are to build robots which exploit nonholonomic effects to control n robot degrees-of-freedom with fewer than n actuators, saving in weight and cost, and to control robots experiencing free-swinging joint failures. Examples of this work include the design of an n joint robot controlled by just two motors via nonholonomic gears (Sørdalen *et al.* [37]); control of underactuated robots in a gravity field, such as the Acrobot (Hauser and Murray [15]; Spong [38]; Berkemeier and Fearing [9]), a high bar robot (Takashima [41]), and a brachiation robot (Saito *et al.* [32]); and control of robots in zero gravity with brakes (Arai and Tachi [3]; Bergerman *et al.* [7]) and without brakes (Suzuki *et al.* [40]; De Luca *et al.* [12]; Nakamura *et al.* [28]; Mareczek *et al.* [25]; Arai *et al.* [4]). This research has focused on the design and control of underactuated manipulators, but there has been little work on planning fast, feasible trajectories among obstacles. In this paper we consider the problem of planning fast collision-free trajectories for a planar 3-DOF underactuated manipulator.

Except in special cases (see Oriolo and Nakamura [29]), each passive, free-swinging joint of an underactuated manipulator introduces a second-order nonholonomic constraint of the form

$$F(\Theta, \dot{\Theta}, \ddot{\Theta}) = 0, \tag{1}$$

where Θ is the n-vector of generalized coordinates. This constraint is called second-order nonholonomic because it cannot be integrated to give a velocity or configuration constraint. Planning collision-free trajectories for an underactuated manipulator is therefore a constrained motion planning problem in the robot's $2n$-dimensional state space. If the robot can be shown to be small-time locally controllable on an open subset O of its zero velocity section, however, then any path in O can be followed arbitrarily closely. This establishes the existence of a trajectory between zero velocity states in a connected component of O.

In this paper we study the particular case of collision-free motion planning between zero velocity states for a three-joint robot in a horizontal plane with a passive revolute third joint. Because there is a single passive joint, the robot is subject to a single second-order nonholonomic constraint of the form (1). We focus on the following aspects of the system:

- *Controllability of the robot.* It is straightforward to show that the robot is small-time locally controllable at zero velocity states. The robot can follow any path arbitrarily closely, and therefore is sufficient for most planar pick-and-place tasks.

- *Decoupling trajectory planning into kinematic path planning followed by time scaling.* This is a common strategy for motion planning for fully actuated robots. Path planning in configuration space is generally not possible for manipulators subject to second-order nonholonomic constraints, however, because the feasible motion directions are a function of the current velocity as well as the configuration. The ability to decouple the trajectory planning problem for the three-joint robot is based on the identification of two special velocity vector fields of the third (passive) link: translation along the length of the link and rotation about the center of percussion of the link with respect to the joint. A path following one of these body-fixed

"decoupling motions" can be executed at any speed while satisfying the nonholonomic constraint. Using these velocity vector fields we are able to adapt a configuration space path planner developed for systems with first-order body-fixed nonholonomic constraints, such as kinematic mobile robots (Barraquand and Latombe [6]) and quasistatic robotic pushing (Lynch and Mason [24]). The velocity of the robot need only be brought to zero at switches between the two velocity vector fields, so the planner minimizes the number of switches. The path returned by the planner can then be time-scaled according to the actuator limits to find the time-optimal trajectory along the path (Shin and McKay [35]; Bobrow *et al.* [10]; Pfeiffer and Johanni [30]; Slotine and Yang [36]; Shiller and Lu [34]). This approach decouples the problem of collision-free trajectory planning into the computationally simpler problems of path planning and time scaling.

- *Fast trajectories.* Although local controllability implies that the robot is capable of following any free path closely, most paths require the robot to remain near zero velocity. On the other hand, the trajectories found by our planner can be executed at high speeds. The trajectories are not globally time-optimal, however; that is precluded by the decoupling of the planning problem.

- *Implementation.* We have implemented our motion planner on an experimental underactuated manipulator. Using nonlinear feedback control (Arai *et al.* [4]) to stabilize the trajectories, we have demonstrated examples of motion planning among obstacles. To our knowledge, this is the first implementation of a collision-free motion planning algorithm for a manipulator subject to a second-order nonholonomic constraint.

This paper builds on previous work establishing the global controllability of a passive third link (Arai [2]) assuming no obstacles or other constraints on the motion of the third joint. The constructive proof outlined a set of maneuvers to move the link from any state to any other state. In particular, any zero velocity state can be achieved from any other by at most three segments: rotation about the center of percussion of the link, translation along the length of the link, and rotation. Later work addressed feedback control to stabilize the rotational and translational segments (Arai *et al.* [4]). In this paper we show that these motions uniquely satisfy a "decoupling property," and we use them to construct free trajectories in the presence of obstacles, joint limits, workspace limits, and actuator constraints.

The position of the center of percussion of the passive link has been recognized as a *differentially flat output* (Fliess *et al.* [14]; Murray *et al.* [27]; Martin *et al.* [26]; Faiz and Agrawal [13]). The angle and angular velocity of the link can be expressed as a function of higher-order derivatives of the position of the center of percussion, so that the trajectory of the center of percussion uniquely specifies the control inputs and the trajectory of the entire passive link, except when the center of percussion is not accelerating. The trajectory planning problem for the link subject to the nonholonomic constraint is then as simple as finding any trajectory of the center of percussion satisfying the endpoint constraints set by the initial and terminal configurations. While this is computationally simple, the difficulty is in incorporating obstacle constraints, joint limits, workspace limits, and actuator constraints. The computational simplicity of the original problem is lost. Work is underway to begin to address these issues in differentially flat linear systems (Agrawal *et al.* [1]).

Bergerman and Xu [8] have previously outlined an approach to motion planning for underactuated manipulators with brakes at the passive joints. The idea is to switch between controlling the active joints, when the brakes are engaged, and the passive joints, when the brakes are released. In this paper we study trajectory planning among obstacles for a particular underactuated robot without a brake, show that the problem can be decoupled into path planning followed by time scaling, and demonstrate the approach on an experimental underactuated manipulator. Although we focus on the 3-DOF robot, the idea of decoupling the trajectory planning problem can be applied to any system for which we can identify "decoupling" velocity vector fields (Bullo and Lynch [11]).

The trajectory planner we describe is also potentially useful for fully actuated 3-DOF robots with a weak actuator at the third joint. By designing trajectories which nominally require zero torque at the third joint, we let the dynamics of the motion assist the third joint. This is analogous to the way that a weightlifter executing a "clean" (lifting a barbell from the ground to rest on the shoulders) initially generates momentum of the barbell with the large muscles of the lower body to assist the weaker muscles of the upper body.

Section 2 provides some definitions. The controllability of the robot is demonstrated in Section 3 and the decoupling velocities used in the trajectory planner are derived in Section 3. The trajectory planner is described in Section 5. Section 6 shows the result of a trajectory implemented on our experimental underactuated manipulator. We conclude in Section 7.

2 Definitions

We define R to be a three-joint robot with a revolute third joint operating in a zero gravity plane. For concreteness, we assume the first two joints of R also are revolute to correspond with our experimental manipulator. (Our approach only requires that the third joint be able to translate freely on an open subset of the plane.) R may be subject to joint limits. We define the fully actuated version of R to be R_A. The underactuated robot R_U is identical to R_A except the third joint is unactuated.

The configuration space C of the robot R is the three-dimensional joint space T^3, with joint angles $\Theta = (\theta_1, \theta_2, \theta_3)^T$. The state space is the tangent bundle TC and the state of the robot is written $(\Theta^T, \dot{\Theta}^T)^T$. The zero velocity section of the state space TC is $Z = \{(\Theta^T, \mathbf{0}^T)^T | \Theta \in C\}$.

We define a fixed world frame F at the first joint and a body frame F_b fixed to the third link of R_U at the joint. The x_b-axis of F_b passes through the center of mass of the third link (Figure 1). The mass of the link is m_3 and its inertia is I_3 about its center of mass. The center of mass lies a distance $r_3 > 0$ from the joint. The center of percussion (also known as the center of oscillation and the center of impact) of the link with respect to the joint is at $(\lambda, 0)$ in the frame F_b, where $\lambda = (r_3^2 + I_3/m_3)/r_3$.

The manipulator dynamic equations (given in detail in the Appendix) are written

$$\mathbf{M}(\Theta)\ddot{\Theta} + \mathbf{C}(\Theta, \dot{\Theta})\dot{\Theta} = \tau, \tag{2}$$

where $\mathbf{M}(\Theta)$ is the 3×3 inertia matrix, $\mathbf{C}(\Theta, \dot{\Theta})\dot{\Theta}$ is the 3×1 vector of centrifugal and Coriolis torques, and $\tau = (\tau_1, \tau_2, \tau_3)^T$ is the 3×1 vector of joint torques. Since $\tau_3 = 0$ for a passive

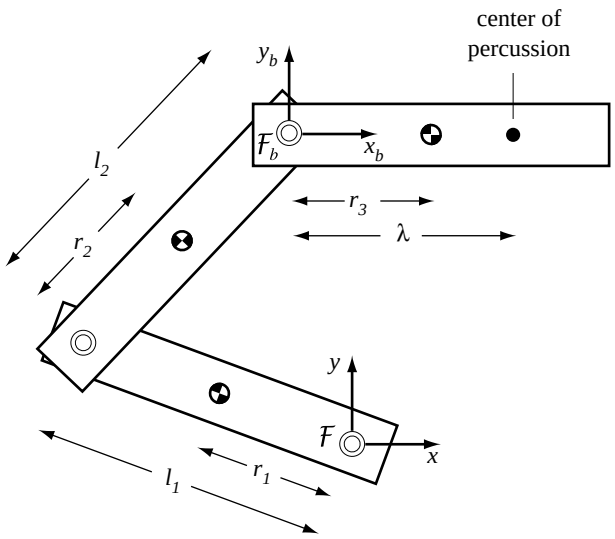

Figure 1: Notation for the underactuated robot with a passive third joint.

third joint, the third element of the vector equation (2) can be simplified to yield the second-order nonholonomic constraint

$$-\ddot{x}\sin\phi + \ddot{y}\cos\phi + \lambda\ddot{\phi} = 0, \tag{3}$$

where $\mathbf{x} = (x, y, \phi)^T \in \mathbf{R}^2 \times S^1$ is the configuration of the link frame F_b in the world frame F, given by

$$\begin{aligned} x &= l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ y &= l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \\ \phi &= \theta_1+\theta_2+\theta_3. \end{aligned}$$

The velocity of F_b in F is written $\dot{\mathbf{x}} = (\dot{x}, \dot{y}, \dot{\phi})^T$. The velocity of F_b relative to F as viewed in the current F_b is called the body velocity $\dot{\mathbf{x}}_b = (\dot{x}_b, \dot{y}_b, \dot{\phi}_b)^T$. The two velocities are related by $\dot{\mathbf{x}} = T\dot{\mathbf{x}}_b$, where

$$T = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

Making the substitution $\ddot{\mathbf{x}} = \dot{T}\dot{\mathbf{x}}_b + T\ddot{\mathbf{x}}_b$ (where $\ddot{\mathbf{x}}_b = d(\dot{\mathbf{x}}_b)/dt$), the nonholonomic constraint (3) can be written in body coordinates as

$$\dot{x}_b\dot{\phi}_b + \ddot{y}_b + \lambda\ddot{\phi}_b = 0. \tag{5}$$

Obstacles are stationary and known. Joint limits and obstacles define a closed subset of C of collision configurations, leaving an open set of free configurations. We restrict the robot to avoid the kinematic singularity at $\sin\theta_2 = 0$, so we divide C into the two open sets given by $\theta_2 \in (0, \pi)$ and $\theta_2 \in (-\pi, 0)$, the RIGHTY and LEFTY configuration spaces. (There are no singularities if the

first two joints are prismatic.) The *free space* C_{free} of the robot R is the open subset of C satisfying the obstacle, joint limit, and singularity constraints.

A *path* $\Theta(s)$ from Θ_0 to Θ_1 is a continuous map $\Theta : [0,1] \to C$, where $\Theta(0) = \Theta_0$ and $\Theta(1) = \Theta_1$. A *trajectory* $\Theta(s(t))$ from $(\Theta_0^T, \mathbf{0}^T)^T$ to $(\Theta_1^T, \mathbf{0}^T)^T$ is obtained from a path by a time-scaling function $s : [0,T] \to [0,1]$ which assigns a configuration on the path to each time $t \in [0,T]$. s is a twice-differentiable function such that $s(0) = 0, s(T) = 1, \dot{s}(0) = \dot{s}(T) = 0$, and $\dot{s}(t) > 0$ for all $t \in (0,T)$. (Note that for a trajectory to be physically feasible, it must also satisfy torque constraints.) A *free path* from Θ_0 to Θ_1 is defined as $\Theta : [0,1] \to C_{free}, \Theta(0) = \Theta_0, \Theta(1) = \Theta_1$, and any feasible time scaling is a *free trajectory* from $(\Theta_0^T, \mathbf{0}^T)^T$ to $(\Theta_1^T, \mathbf{0}^T)^T$. Our definition of free space precludes paths passing through $\theta_2 = 0$ and $\theta_2 = \pi$; while this is a natural restriction for the Cartesian motions of the third link which are used in this paper, this restriction is artificial for joint space motion planning (e.g., Lozano-Pérez and Wesley [22]).

3 Controllability of the Robot

The following proposition establishes the existence of free trajectories for the underactuated robot.

Proposition 1 *If Θ_0 and Θ_1 lie in the same connected component of C_{free}, then there exists a free trajectory from $(\Theta_0^T, \mathbf{0}^T)^T$ to $(\Theta_1^T, \mathbf{0}^T)^T$ for the robot R_U. Equivalently, if there exists a free trajectory from $(\Theta_0^T, \mathbf{0}^T)^T$ to $(\Theta_1^T, \mathbf{0}^T)^T$ for the fully actuated robot R_A, then there exists a free trajectory from $(\Theta_0^T, \mathbf{0}^T)^T$ to $(\Theta_1^T, \mathbf{0}^T)^T$ for R_U with a passive third joint.*

Remark: Proposition 1 also holds when the third link carries a payload such that the center of mass of the combined link and payload does not coincide with the third joint. Thus the robot R_U can be used as a point-to-point pick-and-place robot. When the mass or inertia of the payload, or its configuration within the gripper, is unknown, it is necessary to estimate (e.g., Atkeson *et al.* [5]) the center of percussion of the free link and payload to apply the motion planning and control scheme described in this paper.

Proof: To prove the proposition, we begin by ignoring the first two links of the robot and demonstrate small-time local controllability at zero velocity of the third link when control forces can be applied through the passive joint (the origin of F_b) along the x_b and y_b axes. We then show that this property implies small-time local controllability at zero velocity of the entire robot at any configuration $\Theta \in C_{free}$. Proposition 1 follows directly from small-time local controllability.

Consider the dynamics of the third link written in the control form

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}) + u_1 \mathbf{g}_1(\mathbf{z}) + u_2 \mathbf{g}_2(\mathbf{z}), \;\; u_1, u_2 \in [-\delta, \delta], \delta > 0 \tag{6}$$

where $\mathbf{z} = (\mathbf{x}^T, \dot{\mathbf{x}}^T)^T$ is the state of the link and u_1 and u_2 are the control forces (normalized by dividing by the mass of the link m_3) applied through the passive joint along the x_b and y_b axes, respectively. The corresponding control vector fields are $\mathbf{g}_1 = (0,0,0,\cos\phi,\sin\phi,0)^T$ and $\mathbf{g}_2 = (0,0,0,-\sin\phi,\cos\phi,-1/\lambda)^T$. The drift vector field is $\mathbf{f} = (\dot{x},\dot{y},\dot{\phi},\dot{\phi}^2 r_3 \cos\phi, \dot{\phi}^2 r_3 \sin\phi, 0)^T$. The

centrifugal acceleration terms in $\mathbf{f}$ arise from our choice of reference point at the joint of the third link, not at the center of mass.

It is a simple matter to show that the system (6) satisfies Sussmann's [39] conditions for local controllability at zero velocity provided $r_3 \neq 0$, i.e., the joint is not coincident with the center of mass (see, for example, Reyhanoglu *et al.* [31]; Lewis and Murray [21]; Arai *et al.* [4]; Lynch [23]). Briefly, the vector fields $\mathbf{g}_1$, $\mathbf{g}_2$, and the Lie bracket terms $[\mathbf{g}_1,\mathbf{f}],[\mathbf{g}_2,\mathbf{f}],[\mathbf{g}_2,[\mathbf{g}_1,\mathbf{f}]],[\mathbf{f},[\mathbf{g}_2,[\mathbf{g}_1,\mathbf{f}]]]$ span the six-dimensional tangent space at any state $(\mathbf{x}^T,\mathbf{0}^T)^T$, and the "bad" bracket terms $[\mathbf{g}_1,[\mathbf{f},\mathbf{g}_1]]$ and $[\mathbf{g}_2,[\mathbf{f},\mathbf{g}_2]]$ are neutralized. (See Sussmann [39] for a description of bad brackets and neutralization.) The Lie bracket term $[\mathbf{g}_2,[\mathbf{g}_1,\mathbf{f}]] = (0,0,0,-(\sin\phi)/\lambda,(\cos\phi)/\lambda,0)^T$ corresponds to translation of the link along the body y_b-axis. Approximate motion in this direction is generated by switching between $\mathbf{g}_1$ and $\mathbf{g}_2$, and is therefore slow relative to motion in $\mathbf{g}_1$ and $\mathbf{g}_2$. This is similar to a car's parallel-parking motion.

Now consider the Cartesian forces which can be applied at the third joint of R_U, $(\mathbf{J}_a^T)^{-1}\tau_a$, where $\mathbf{J}_a$ is the Jacobian relating the angular velocities of the actuated joints 1 and 2 to the velocity $(\dot{x},\dot{y})^T$ of the third joint, and $\tau_a = (\tau_1,\tau_2)^T$ is the torque of the actuated joints. $\mathbf{J}_a$ is nonsingular away from $\theta_2 = 0,\pi$. Therefore, for any $\Theta \in C_{free}$ and any set of joint torques Υ containing a neighborhood of $(0,0)^T$ in the τ_a space (e.g., the symmetric joint torque limits $\Upsilon = \{(\tau_1,\tau_2)^T \,|\, |\tau_i| \leq \tau_{i,max}, \tau_{i,max} > 0, i = 1,2\}$), we can choose a sufficiently small $\delta > 0$ in (6) such that the set of forces $F = \{(\mathbf{J}_a^T)^{-1}\tau_a | \tau_a \in \Upsilon\}$ is a superset of the forces in (6). Therefore, the third link of R_U is locally controllable at all robot configurations $\Theta \in C_{free}$.

Finally, small-time local controllability of the third link directly implies small-time local controllability of the robot R_U at all zero velocity states $(\Theta^T,\mathbf{0}^T)^T, \Theta \in C_{free}$. (The inverse kinematics mapping $\mathbf{r}^{-1} : \mathbf{x} \to \Theta$ is smooth and one-to-one when restricted to the RIGHTY or LEFTY connected component of C_{free}, and $\mathbf{r}^{-1}(\mathbf{x}) \in \text{int}(\mathbf{r}^{-1}(B(\mathbf{x})))$, where $B(\mathbf{x})$ is any neighborhood of $\mathbf{x}$ and $\mathbf{r}^{-1}(B(\mathbf{x})) = \{\mathbf{r}^{-1}(\mathbf{x}_0) | \mathbf{x}_0 \in B(\mathbf{x})\}$.) Small-time local controllability implies that for any state $(\Theta^T,\mathbf{0}^T)^T, \Theta \in C_{free}$ and any neighborhood W of $(\Theta^T,\mathbf{0}^T)^T$, $(\Theta^T,\mathbf{0}^T)^T$ is interior to the set of states reachable by trajectories remaining in W. In particular, there is an open accessible set of the zero velocity section Z with $(\Theta^T,\mathbf{0}^T)^T$ in the interior. These open sets form a finite subcover of any path from Θ_0 to Θ_1 in C_{free}, so the path can be followed arbitrarily closely.

For any free trajectory of R_A from $(\Theta_0^T,\mathbf{0}^T)^T$ to $(\Theta_1^T,\mathbf{0}^T)^T$, the corresponding path lies in C_{free}. Because R_U is small-time locally controllable on C_{free}, R_U can follow the path from Θ_0 to Θ_1 arbitrarily closely. □

Proposition 1 implies that any path in C_{free} can be followed arbitrarily closely. Arbitrary paths may make extensive use of Lie bracket motions, however, requiring R_U to stay near the zero velocity section, resulting in slow execution times (see Figure 2). To find paths that can be executed quickly, we prefer to minimize the use of Lie bracket motions.

4 Decoupling Motions

Our goal is to develop a collision-free trajectory planner which (1) is computationally efficient and (2) yields trajectories which can be executed quickly. Our approach is based on the existence of

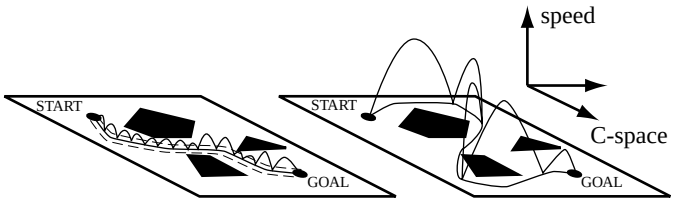

Figure 2: A conceptual representation of two collision-free trajectories between the same start and goal configurations for a small-time locally controllable system. The planes represent the configuration space, and the normal direction represents speed along the path. The figure on the left shows an example path and a tube of free space around it. To approximately follow the path and stay within the configuration tube, the system must return to the zero velocity section often, resulting in slow motion. This is the case for trajectories that make extensive use of Lie bracket motions. The path on the right can be executed much more quickly—the system can move far from the zero velocity section.

body-fixed "decoupling" motions for the third link subject to the second-order nonholonomic constraint (3). These decoupling motions allow the trajectory planning problem to be decoupled into the computationally simpler problems of collision-free path planning in the configuration space followed by time scaling. By the nature of the decoupling motions, the paths found by the path planner can be followed at high speeds.

We will consider decoupling motions for a second-order mechanical system

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \begin{pmatrix} \mathbf{M}_a(\mathbf{q}) \\ \mathbf{M}_u(\mathbf{q}) \end{pmatrix} \ddot{\mathbf{q}} + \begin{pmatrix} \mathbf{C}_a(\mathbf{q},\dot{\mathbf{q}}) \\ \mathbf{C}_u(\mathbf{q},\dot{\mathbf{q}}) \end{pmatrix} \dot{\mathbf{q}} = \begin{pmatrix} \mathbf{u} \\ \mathbf{0} \end{pmatrix}, \tag{7}$$

where $\mathbf{q} \in \Re^n$ is the configuration and $\mathbf{u} \in \Re^m$ is the control. Let M be an open subset of the configuration space, and let (7) be equilibrium controllable on M. Equilibrium controllability is a weaker condition than small-time local controllability at zero velocity (Lewis and Murray [21]). The $k = n - m$ second-order nonholonomic constraints are written $\mathbf{M}_u(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_u(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0}$.

Consider a trajectory $\mathbf{q}(s(t))$ of (7), where s is a time-scaling function. The velocity and acceleration of the system are written

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{\partial \mathbf{q}}{\partial s}\dot{s} = \mathbf{v}(\mathbf{q})\dot{s} \\ \ddot{\mathbf{q}} &= \dot{\mathbf{v}}(\mathbf{q})\dot{s} + \mathbf{v}(\mathbf{q})\ddot{s} = \frac{\partial \mathbf{v}}{\partial \mathbf{q}}\mathbf{v}(\mathbf{q})\dot{s}^2 + \mathbf{v}(\mathbf{q})\ddot{s}, \end{aligned}$$

where $\mathbf{v}(\mathbf{q}) = \frac{\partial \mathbf{q}}{\partial s}$ is a velocity vector field for which $\mathbf{q}(s(t))$ is an integral curve. $\dot{s}$ and $\ddot{s}$ are the velocity and acceleration along the integral curve, respectively.

Definition 1 *The velocity vector field* $\mathbf{v}(\mathbf{q})$ *on* $\Re^n$ *is called a* decoupling motion on M *for the system (7) if and only if* $\mathbf{M}_u(\mathbf{q})(\frac{\partial \mathbf{v}}{\partial \mathbf{q}}\mathbf{v}(\mathbf{q})\dot{s}^2 + \mathbf{v}(\mathbf{q})\ddot{s}) + \mathbf{C}_u(\mathbf{q},\mathbf{v}(\mathbf{q})\dot{s})\mathbf{v}(\mathbf{q})\dot{s} = 0$ *for all* $\mathbf{q} \in M$ *and any time scaling* $s(t)$.

In other words, if $\mathbf{v}(\mathbf{q})$ is a decoupling motion, then for any integral curve of $\mathbf{v}(\mathbf{q})$, the system can travel at any speed $\dot{s}$ along the curve, with any acceleration $\ddot{s}$ along the curve, while satisfying the nonholonomic constraints.

Definition 2 *The system (7) is* locally kinematically controllable on M *if there exists a set of decoupling motions* $\mathbf{v}_1(\mathbf{q}), \ldots, \mathbf{v}_p(\mathbf{q})$ *on* M *such that the kinematic system*

$$\dot{\mathbf{q}} = \sum_{i=1}^{p} w_i \mathbf{v}_i(\mathbf{q}), \;\; (w_1, \ldots, w_p) \in \{(\pm 1, 0, \ldots, 0), (0, \pm 1, 0, \ldots, 0), \ldots, (0, \ldots, 0, \pm 1)\}, \qquad (8)$$

is small-time locally controllable at all $\mathbf{q} \in M$.

If the system is locally kinematically controllable on its configuration space, we can employ a path planner for the kinematic system (8), and any time-scaling of the resulting path will satisfy the second-order nonholonomic constraints. This is the case for the underactuated robot R_U.

Proposition 2 *The underactuated robot* R_U *is locally kinematically controllable on* C_{free} *by the decoupling motions* $\mathbf{v}_1(\mathbf{x}) = (\cos\phi, \sin\phi, 0)^T$ *and* $\mathbf{v}_2(\mathbf{x}) = (-\sin\phi, \cos\phi, -1/\lambda)^T$*, expressed in the coordinates of the third link. These motions are fixed in the frame* F_b*, and can be written as* $\dot{\mathbf{x}}_b = (1, 0, 0)^T$ *(translation along the length of the link) and* $\dot{\mathbf{x}}_b = (0, 1, -1/\lambda)^T$ *(rotation about the center of percussion), respectively.*

Proof: The second-order nonholonomic constraint (3) can be rewritten

$$(-\sin\phi, \cos\phi, \lambda)\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{v}(\mathbf{x})\dot{s}^2 + \mathbf{v}(\mathbf{x})\ddot{s}\right) = 0.$$

For this constraint to hold for all $\dot{s}, \ddot{s}$, we must separately have

$$(-\sin\phi, \cos\phi, \lambda)\left(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{v}(\mathbf{x})\right) = 0 \qquad (9)$$

$$(-\sin\phi, \cos\phi, \lambda)\mathbf{v}(\mathbf{x}) = 0. \qquad (10)$$

Equation (10) is only satisfied by linear combinations of $\mathbf{v}_1(\mathbf{x})$ and $\mathbf{v}_2(\mathbf{x})$. Plugging $\mathbf{v}(\mathbf{x}) = \alpha\mathbf{v}_1(\mathbf{x}) + \beta\mathbf{v}_2(\mathbf{x})$ into Equation (9), after some simplification we get

$$\alpha\beta = 0.$$

In other words, one of α, β must be zero, and the other is unconstrained. Hence the velocity vector fields $\mathbf{v}_1(\mathbf{x})$ and $\mathbf{v}_2(\mathbf{x})$ (and their scalar multiples) are the only decoupling motions for R_U. Small-time local controllability of the kinematic system on C_{free} follows easily (see, for instance, the proof of controllability of the kinematic car in Latombe [18]). □

Remark: In the presence of gravity, the underactuated robot remains differentially flat (Martin *et al.* [26]) but is no longer locally kinematically controllable.

We have constructed a computationally efficient collision-free trajectory planner for R_U based on the following three observations:

- *Trajectory planning in the six-dimensional state space can be decoupled into path planning in the three-dimensional configuration space followed by time-optimal time scaling of the path.* For the underactuated robot, the decoupling vector fields correspond to body-fixed motions, allowing us to adapt path planners for car-like mobile robots with a body-fixed nonholonomic constraint (Barraquand and Latombe [6]; Laumond *et al.* [19]). In the resulting paths, the velocity of the third link need only be brought to zero when switching between $\mathbf{v}_1$ and $\mathbf{v}_2$. Therefore, we design the planner to minimize the number of switches, implicitly minimizing the use of Lie bracket motions.

  Decoupling trajectory planning into path planning followed by time scaling greatly reduces the computational complexity.

- *If the path planner is complete, then a trajectory will be found for any two configurations in the same connected component of C_{free}.* We are using a modification of a path planning algorithm designed for mobile robots (Barraquand and Latombe [6]) and robotic pushing (Lynch and Mason [24]). This algorithm is *resolution-complete*, meaning that if a free path exists, the planner will find a free path for a sufficiently small choice of a parameter to the planner.

- *The paths found by the kinematic motion planner can be executed at high speeds.* The nonholonomic constraint does not constrain the speed of motion along paths following $\mathbf{v}_1$ or $\mathbf{v}_2$. Limits on execution speed arise solely from actuator limits.

5 The Trajectory Planner

5.1 Path Planning

The path planner is a simple best-first search in the configuration space along the vector fields $\mathbf{v}_1$ and $\mathbf{v}_2$. Starting from the initial link configuration $\mathbf{x}_{init}$ (and corresponding initial joint configuration Θ_{init}), the planner integrates forward along each of $+\mathbf{v}_1, -\mathbf{v}_1, +\mathbf{v}_2, -\mathbf{v}_2$ for a time δt, yielding four new link configurations. Each new collision-free configuration $\mathbf{x}_{new}$ (and corresponding joint configuration Θ_{new}) is added to a search tree T and to a sorted list *OPEN* of configurations in T whose successors have not yet been generated. Configurations in *OPEN* are sorted by the costs of their paths. The first configuration in *OPEN* is then expanded. This process continues until a path is found to a user-specified goal neighborhood $G(\mathbf{x}_{goal})$ or until *OPEN* is empty (failure). The planner is not exact, as it only finds a path to a goal neighborhood. An exact planner, which could also be applied to this problem, is described by Laumond *et al.* [19].

Because the robot must be brought to zero velocity to switch between velocity directions, the cost function is the number of switches in the velocity direction. If δt is set small enough, the planner will find a path when one exists.

```
program path_planner
  initialize T, OPEN with link start configuration x_init
  while OPEN not empty
    x ← first in OPEN, remove from OPEN
    if x is in the goal region
      report success
    if x is not near a previously occupied configuration
      mark x occupied
      for each of +v1, −v1, +v2, −v2
        integrate forward a time δt to x_new
        calculate Θ_new by inverse kinematics
        if path to Θ_new is collision-free
          make x_new a successor to x in T
          compute cost of path to new config x_new
          place x_new in OPEN, sorted by cost
  report failure
end
```

5.1.1 Collision Detection

The robot links and obstacles are represented as polygons, and collision detection in the current implementation consists of polygon intersection at each new configuration Θ_{new}. (To make collision detection conservative, we could grow the links or obstacles.) A link configuration $\mathbf{x}$ is also considered to be in collision if there is no solution to the inverse kinematics or if $|\sin\theta_2| \leq \varepsilon$, where $\varepsilon > 0$ is a design parameter specifying how close the robot may approach the singularity.

The planner also prunes configurations that are sufficiently near another configuration reached with the same or lower cost and the same motion direction. In the current implementation, two configurations $\mathbf{x}_1$ and $\mathbf{x}_2$ are considered sufficiently near if they occupy the same cell of a predefined grid on the three-dimensional link configuration space $\mathbf{R}^2 \times S^1$.

5.1.2 Parameters

The user must specify the size of the goal region, the integration step δt, and the resolution of the $\mathbf{R}^2 \times S^1$ grid used to check for prior occupancy. These parameters are interdependent. In particular, the step size δt and the dimensions of the grid cells should be proportional to the dimensions of the goal region. Each step should always take the configuration out of the current grid cell, and the step should not be so large as to easily jump across the goal region $G(\mathbf{x}_{goal})$.

5.1.3 Complexity

The size of the $\mathbf{R}^2 \times S^1$ grid for pruning is c^3, where c is the number of discretization intervals along the x, y, and ϕ axes. (x and y are confined to a rectangular subset of $\mathbf{R}^2$ covering all possible positions of the joint of the third link. The number of discretization intervals along each axis need not be equal, but here we assume it for simplicity.) Each configuration must undergo a collision-check which takes time mn in our implementation, where m and n are the number of vertices in the robot and the obstacles, respectively. Because path costs are always integral (the number of switches), the sorted list *OPEN* is represented by a 1-d array of linked lists, where each array index represents the cost of the paths to the configurations in its linked list. Inserting a new configuration into *OPEN* consists of simply appending it to the end of the appropriate linked list, and therefore takes constant time. All other operations (including incremental computation of path cost) take constant time, giving a time complexity of $O(mnc^3)$. The volume V of the goal region $G(\mathbf{x}_{goal})$ should be proportional to c^{-3}, so the running time is inversely proportional to V, $O(mnV^{-1})$.

5.2 Time Scaling

Each rotational or translational motion segment specifies a path $\Theta(s)$ where $\Theta(0) = \Theta_0$ and $\Theta(1) = \Theta_1$. For each segment, we would like to find the time-optimal time scaling of the path $s(t)$ satisfying actuator limits. The minimum-time trajectory for the entire path found by the planner is the concatenation of the time-optimal motion segments.

The motion of the third link is always a translation or a rotation about the center of percussion, so we choose the path parameter s for each motion segment to be the translation distance or rotation angle, normalized so that the full motion is unit. The configuration of the free link $\mathbf{x}(s)$ during translation from $\mathbf{x}(0) = \mathbf{x}_i = (x_i, y_i, \phi_i)^T$ to $\mathbf{x}(1) = \mathbf{x}_f = (x_f, y_f, \phi_i)^T$ is given as

$$\begin{aligned} \phi(s) &= \phi_i \\ x(s) &= s(x_f - x_i) + x_i \\ y(s) &= s(y_f - y_i) + y_i, \end{aligned}$$

and during rotation from $\mathbf{x}(0) = \mathbf{x}_i = (x_i, y_i, \phi_i)^T$ to $\mathbf{x}(1) = \mathbf{x}_f = (x_f, y_f, \phi_f)^T$ (ϕ_i and ϕ_f defined such that $\phi(s)$ is continuous)

$$\begin{aligned} \phi(s) &= s(\phi_f - \phi_i) + \phi_i \\ x(s) &= x_c - r_3 \cos\phi(s) \\ y(s) &= y_c - r_3 \sin\phi(s), \end{aligned}$$

where (x_c, y_c) is the stationary center of percussion expressed in the world frame F.

The robot's forward kinematics are given by $\mathbf{x} = \mathbf{r}(\Theta)$, yielding the following expressions for $\Theta, \dot{\Theta}, \ddot{\Theta}$ as a function of $s, \dot{s}, \ddot{s}$:

$$\Theta = \mathbf{r}^{-1}(\mathbf{x}) \tag{11}$$

$$\dot{\Theta} = (\mathbf{r}_\Theta)^{-1}\mathbf{x}_s\dot{s} \tag{12}$$

$$\ddot{\Theta} = (\mathbf{r}_\Theta)^{-1}(\mathbf{x}_s\ddot{s} + \mathbf{x}_{ss}\dot{s}^2 - ((\mathbf{r}_\Theta)^{-1}\mathbf{x}_s)^T\mathbf{r}_{\Theta\Theta}((\mathbf{r}_\Theta)^{-1}\mathbf{x}_s)\dot{s}^2), \tag{13}$$

where the subscripts s and Θ denote derivatives with respect to the scalar s and the vector Θ, with $\mathbf{r}_\Theta$ the manipulator Jacobian and $\mathbf{r}_{\Theta\Theta}$ the $3\times3\times3$ Hessian. (The term $((\mathbf{r}_\Theta)^{-1}\mathbf{x}_s)^T\mathbf{r}_{\Theta\Theta}((\mathbf{r}_\Theta)^{-1}\mathbf{x}_s)\dot{s}^2$ can be written $\dot{\mathbf{r}}_\Theta\dot{\Theta}^2$.) Since the paths found by the planner are confined to either the RIGHTY or LEFTY subset of free space, these expressions yield unique solutions for $\Theta, \dot{\Theta}, \ddot{\Theta}$.

Substituting Equations (11)–(13) into the manipulator dynamics (2) yields a set of dynamic equations of the form

$$\mathbf{a}(s)\ddot{s} + \mathbf{b}(s)\dot{s}^2 = \tau.$$

We assume the actuator torques are bounded by constants

$$\tau_i^{min} \le \tau_i \le \tau_i^{max}, \;\; i = 1, 2.$$

(The constraint $\tau_3 = 0$ is satisfied automatically by any time scaling of paths found by the path planner.) These torque constraints place bounds on $\ddot{s}$ for a given state $(s, \dot{s})$. The problem is to find the fastest trajectory in the $(s, \dot{s})$ phase plane from $(0,0)$ to $(1,0)$ satisfying the torque constraints.

The minimum-time time scaling problem has been solved by algorithms proposed by Bobrow *et al.* [10], Shin and McKay [35], Pfeiffer and Johanni [30], Slotine and Yang [36], and Shiller and Lu [34]. The idea is illustrated in Figure 3. At a state $(s, \dot{s})$, the set of possible path accelerations $[\ddot{s}_{min}, \ddot{s}_{max}]$ defines a cone of possible motion directions in the $(s, \dot{s})$ phase plane. The minimum-time trajectory is given by the curve from $(0,0)$ to $(1,0)$ that maximizes the area underneath it while maintaining the tangent within the cone at all points on the curve. A consequence of this is that the curve always follows the upper or lower bound of the cone (maximum or minimum acceleration).[1] The heart of the time-scaling problem is to find the switching points between maximum and minimum acceleration.

In this paper we use a modification of the algorithm of Bobrow *et al.* [10] to find the optimal time scaling. We do not give the details of the algorithm; details on this and similar algorithms can be found in the references above.

5.3 Results

The planner and time-scaling algorithm are implemented in C on a Sun Ultra 30. A graphical interface written in Tcl/Tk was developed to allow the user to easily create problems to test the planner.

Figure 4 shows a result of the planner applied to a robot with the kinematic and dynamic parameters listed in Table 1. The robot is confined to the $\sin\theta_2 > 0$ connected component of C_{free}. We set the goal region to $\pm1.7^\circ$ and $\pm6\%$ of the length of the third link. The $\mathbf{R}^2 \times S^1$ grid cell dimensions are 1.7° and 6% of the length of the third link, so the volume of $G(\mathbf{x}_{goal})$ is 8 times that of a grid cell. The step length is chosen just large enough that a step will always take the link out of the current cell.

We applied the time-scaling algorithm to obtain the time-optimal trajectory following the path of Figure 4. The results are shown in Figure 5. During the time-optimal motion, at least one of

[1] Except at dynamic singularities; see Shiller and Lu [34].

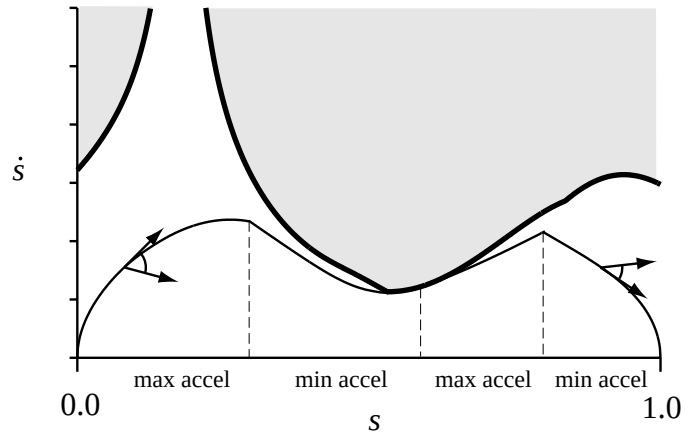

Figure 3: At each point $(s, \dot{s})$ in the phase plane we can draw a motion cone defined by the minimum and maximum accelerations $\ddot{s}$ satisfying the torque constraints. The time-optimal trajectory from $(0, 0)$ to $(1, 0)$ is the curve that maximizes the area underneath it while remaining on the minimum or maximum boundary of the motion cones. In this example, the trajectory switches between minimum and maximum acceleration three times. The heavy curve indicates the states where the cone collapses to a single tangent vector, and the gray region represents states where the cone disappears—no feasible torques will keep the manipulator on the path.

Joint i	l_i (m)	r_i (m)	m_i (kg)	I_i (kg-m^2)	λ (m)	τ_i^{max} (N-m)	τ_i^{min} (N-m)
1	0.3	0.15	2.0	0.02		20	-20
2	0.3	0.15	1.0	0.01		10	-10
3		0.15	0.5	0.004125	0.205	0	0

Table 1: Kinematic parameters, inertial parameters, and actuator limits for the simulated robot.

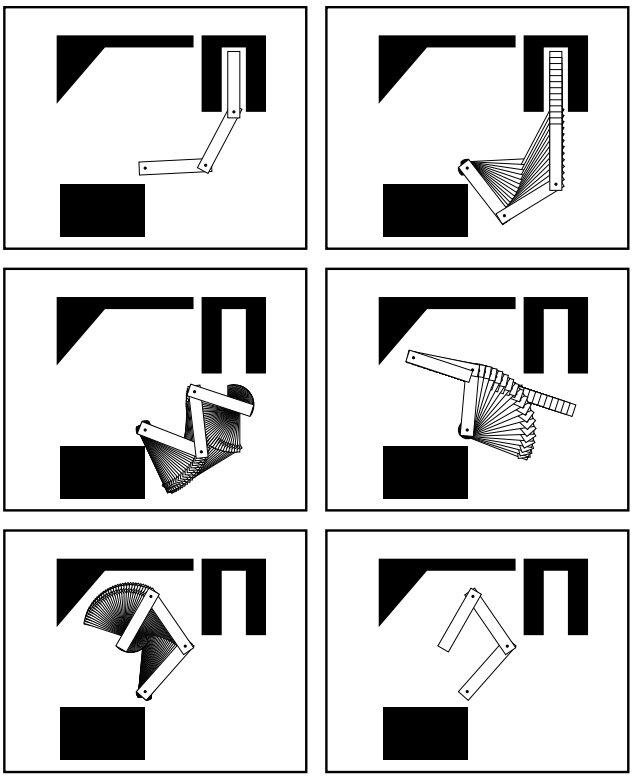

Figure 4: A four segment path for a robot with no joint limits.

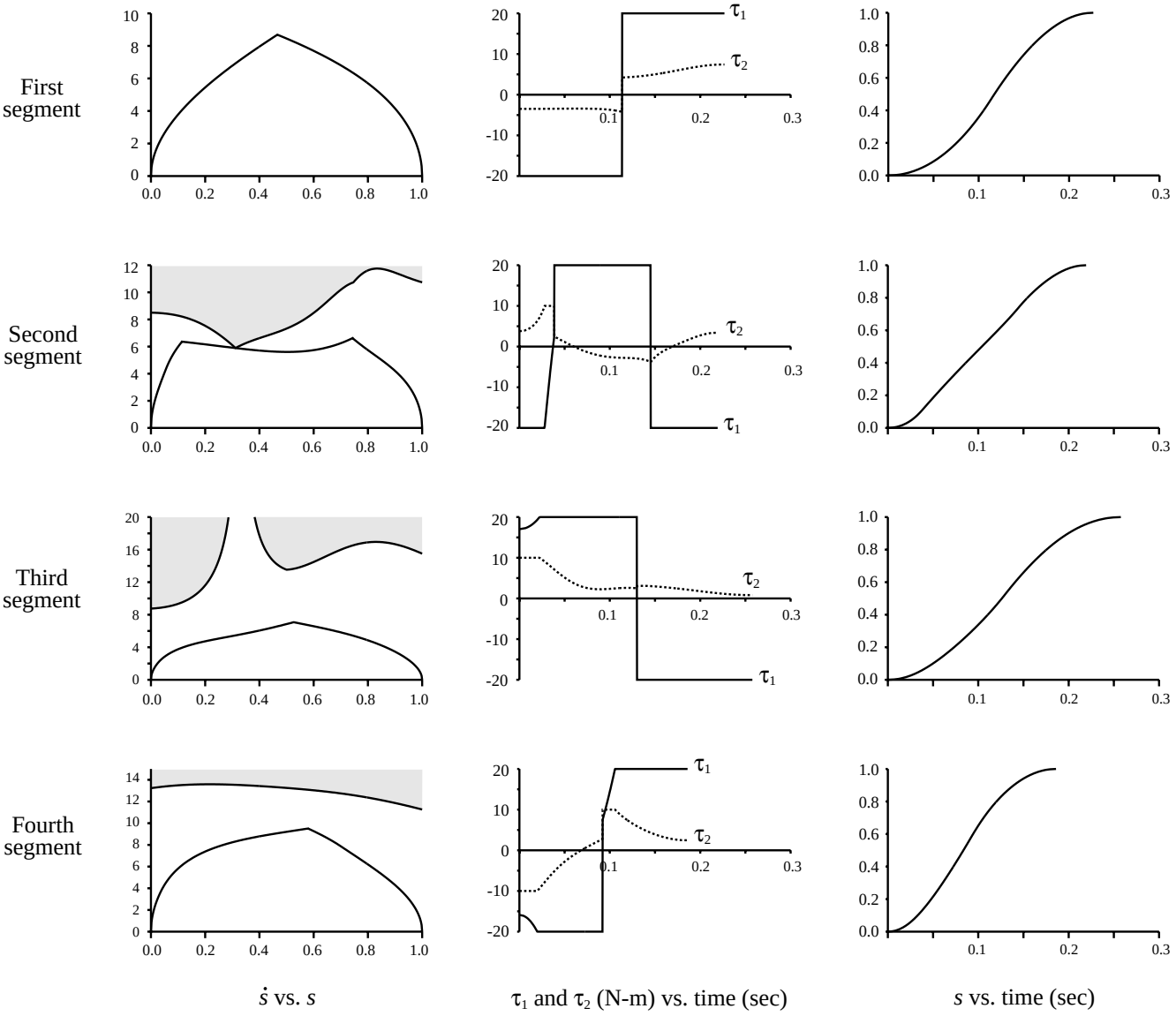

Figure 5: Results of time scaling the path of Figure 4. The first, third, and fourth motion segments contain only one switching point; the second segment has three.

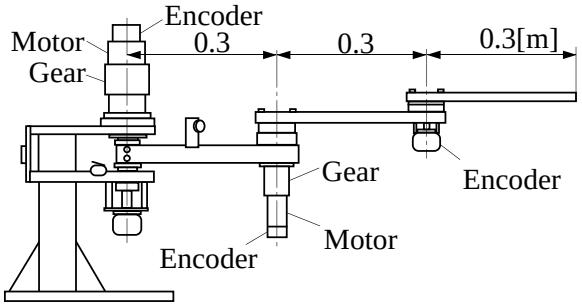

Figure 6: The MEL underactuated manipulator.

the robot's two actuators is always saturated. In this example, the actuator limits are given by $|\tau_1| \leq 20$ N-m, $|\tau_2| \leq 10$ N-m. The time-optimal trajectory following the first motion segment of Figure 4 takes 0.227 s; the second segment takes 0.219 s; the third segment takes 0.258 s; and the fourth segment takes 0.186 s, for a total motion time of 0.890 s.

In this example we chose $\varepsilon = 0.15$, giving the configuration constraint $|\sin\theta_2| > 0.15$. Small values of ε allow the Jacobian $\mathbf{r}_\Theta$ to become nearly singular, implying large joint velocities $\dot{\Theta}$ for fixed Cartesian velocities $\dot{\mathbf{x}}$. If we would like the robot to be able to operate near singularities, the step size δt should be decreased to enforce a bound on joint motions during a step. This increases the computational complexity, which is proportional to $(\delta t)^{-3}$. An alternative approach is to choose δt based on the principal axes of the manipulability ellipsoid (Yoshikawa [42]) at the current configuration, similar to adaptive stepsizing in numerical integration.

It remains an open issue how to best choose a step size δt given a goal region size and some characterization of the "tightness" of the space (a function of the obstacles, joint limits, and ε).

The planning time for the example here was about 20 seconds, with time scaling taking a negligible fraction of the time. Typical planning time is between 10 and 120 seconds.

6 Experiment

We tested a planned motion on the experimental three degree-of-freedom planar manipulator of Figure 6. The length of each link is 0.3 m. The first and second joints are actuated by 35 W and 20 W DC servo motors, respectively. The third joint is passive and has neither an actuator nor a brake. The angle of each joint is measured by a rotary encoder. A personal computer (80486 CPU, 50 MHz) is used as the controller. The distance λ between the center of percussion of the free link and the passive joint is 0.205 m, as with the simulated robot of the previous section.

Because time-optimal trajectories cause actuators to saturate, they allow no margin for error-recovery in feedback control of the nominal trajectories. One possibility is to use a conservative estimate of the available torque in the time-scaling phase to ensure that some torque remains for error correction. In our experiments, however, we empirically chose times for each motion segment for fast but robust performance.

When a planned motion is executed on the real manipulator, trajectory error accumulates due

to initial position error, model error, etc. This error can be reduced by nonlinear feedback control (Arai *et al.* [4]). During translation the third link acts either as a pendulum or an inverted pendulum, depending on the current direction of acceleration. Exact linearization is used to find a linear state feedback controller which stabilizes the link to the trajectory. During rotation, two separate linear controllers are used to stabilize the trajectory, one during phases of low angular velocity and another during phases of high angular velocity. These controllers calculate the desired acceleration of the passive joint $(\ddot{x},\ddot{y})$. This acceleration is integrated to yield the virtual reference velocity $(\dot{x}_m,\dot{y}_m)$ and virtual reference position (x_m,y_m). These references are commanded to the servo system. The desired acceleration $(\ddot{x},\ddot{y})$ is also used as a feedforward term. This method, known as virtual internal model following control (Kosuge *et al.* [17]), suppresses unknown disturbance forces such as friction.

To accommodate any remaining error, we grow the obstacles in the planner by the maximum estimated position error of any point on the robot. We also choose joint limits to artificially restrict the robot's motion, so the real joint limits will not be encountered during execution.

Figure 7 shows an 11 segment path found by the planner and the actual path executed by the robot under feedback control. The square 0.26 m $\times$ 0.26 m obstacles in the real scene (right side of Figure 7) have been grown by 0.02 m in each direction to yield the 0.3 m $\times$ 0.3 m obstacles used in the planner (left side of Figure 7). We also limited the joint angles to $-1.17\,\text{rad} < \theta_1 < 1.17\,\text{rad}$ (where 0 is vertical on the page) and $0.5\,\text{rad} < \theta_2 < 1.95\,\text{rad}$ (i.e., $\varepsilon = \sin 0.5$). The manipulator's real joint limits are $-1.57\,\text{rad} < \theta_1 < 1.57\,\text{rad}$, $\theta_2 < 2.356\,\text{rad}$.

As Figure 7 shows, the robot follows the planned path fairly closely. Execution times of the segments were chosen experimentally to make control robust. For reference, motion segments 1 and 2 were executed in approximately 0.36 s and 2.4 s, respectively. A pause of 0.2 s was also inserted between motion segments. The path was executed without the obstacles; the obstacles have been superimposed on the data. In this example, the 0.02 m error estimate is a bit optimistic—the robot nearly brushes an obstacle in the fourth and fifth motion segments. A video of an experiment is available at `http://lims.mech.northwestern.edu/~lynch/research/videos`.

This example path was fairly challenging to implement, as it consists of many motion segments and moves the robot over a large portion of its workspace. Small out-of-plane motion of the third link contributes to errors in control, and currently the controller is not robust for all paths that the planner generates. Nevertheless, this example demonstrates the ability to plan and execute collision-free motions for an underactuated manipulator among obstacles.

7 Conclusion

We have demonstrated that a free trajectory exists between any two zero velocity states in the same connected component of the free space for a three degree-of-freedom robot with a passive joint. We have shown that motion planning for this system can be decoupled into path planning and time scaling, and we have presented a motion planner that is resolution-complete—for a sufficiently small step size δt, the planner will find a path when one exists. The planner minimizes the number of times the robot velocity must come to zero, producing a fast trajectory when time-scaled accord-

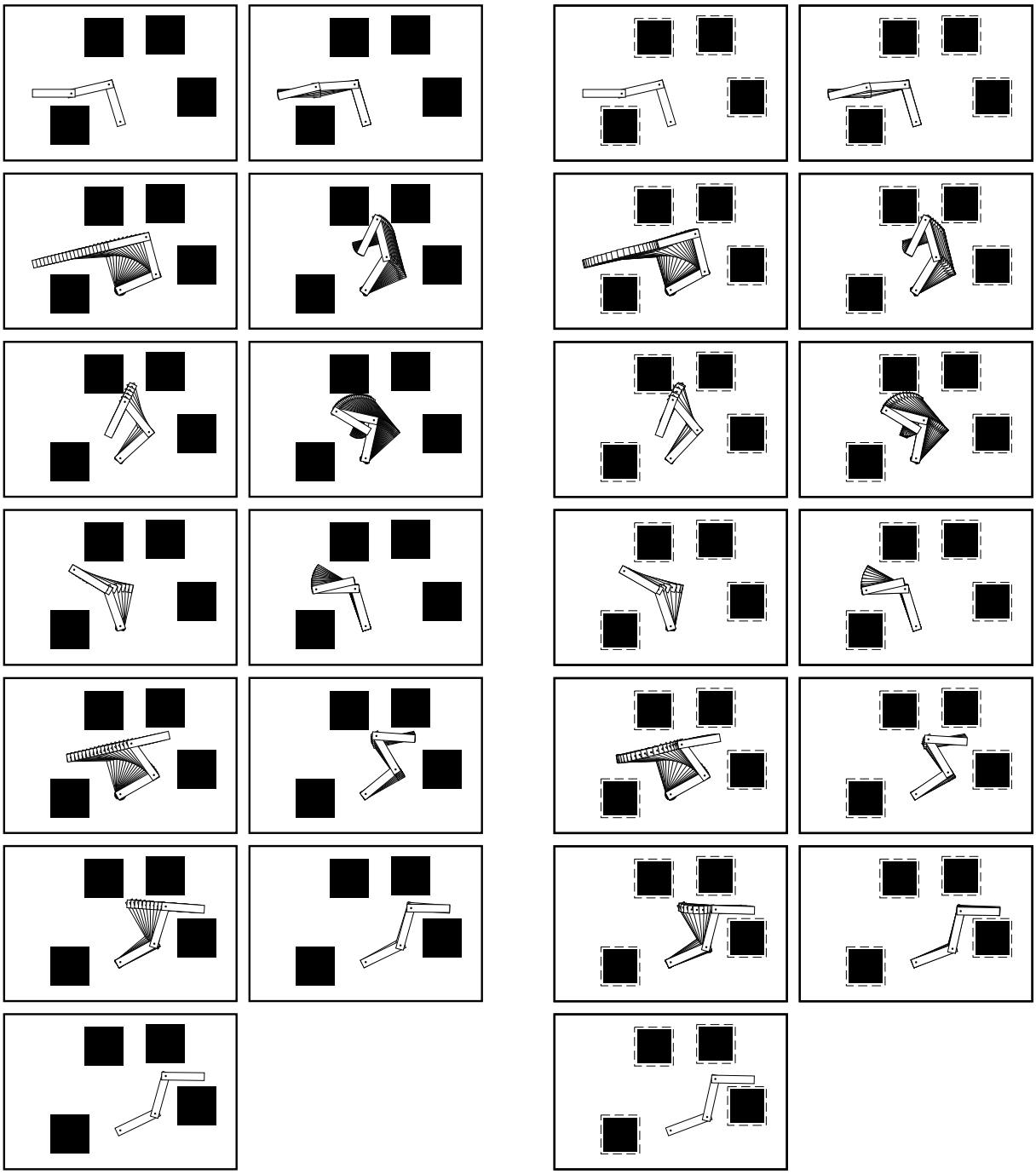

Figure 7: The planner found the 11 segment path on the left using grown obstacles. The right side shows the results of executing the path on the experimental manipulator under feedback control. The dashed lines show the borders of the grown obstacles used in the planner.

ing to the manipulator dynamics. Finally, we have demonstrated the feasibility of the approach by implementing a planned path on an experimental underactuated manipulator. The capability to automatically plan and execute collision-free trajectories makes it possible to eliminate an actuator from a planar three joint robot used for point-to-point tasks such as pick-and-place.

There are several ways this work can be extended:

- Adaptive control could be used to estimate the location of the center of percussion of the free link to make path planning and control robust to changes in the payload.
- By analyzing and bounding control errors, an estimate could be obtained of the fastest time scaling that can be followed robustly.
- The basic approach can be generalized to other systems with second-order nonholonomic constraints admitting the decoupling into path planning and time scaling.
- This paper has presented a reduced complexity approach to finding sub-time-optimal collision-free trajectories in the robot's $2n$-dimensional state space. Nonlinear optimization or other techniques could be used to search for a time-optimal trajectory directly, instead of decomposing the problem into path planning and time-scaling problems. However, brute force search has high computational complexity and gradient-descent approaches are limited to finding local optima and tend to encounter convergence problems. Some combination of search and optimization may provide an efficient way of finding globally optimal trajectories, similar to the approach employed by Shiller and Dubowsky [33] for fully actuated robots.

  The very different approach of randomized motion planning (e.g., Hsu *et al.* [16]; LaValle and Kuffner [20]) shows promise for fast trajectory planning directly in the system state space. Future work could adapt the approach to underactuated manipulators. Current implementations require a discretization in the control space and often search only for a feasible solution, rather than an efficient solution. These planners tend to produce solutions with discontinuous controls, and a method for smoothing the solutions subject to second-order nonholonomic constraints would be desirable.

Appendix

The dynamic equations of motion for a three joint revolute robot in a horizontal plane, ignoring joint friction, are:

$$\mathbf{M}(\Theta)\ddot{\Theta} + \dot{\Theta}^T\Gamma(\Theta)\dot{\Theta} = \tau,$$

where

$$\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}$$

$$
\begin{aligned}
M_{11} &= I_1 + I_2 + I_3 + m_1 r_1^2 + m_2 l_1^2 + m_2 r_2^2 + m_3 l_1^2 + m_3 l_2^2 + m_3 r_3^2 \\
&\quad + 2m_2 l_1 r_2 c_2 + 2m_3 l_1 l_2 c_2 + 2m_3 l_2 r_3 c_3 + 2m_3 l_1 r_3 c_{23} \\
M_{12} = M_{21} &= I_2 + I_3 + m_2 r_2^2 + m_3 l_2^2 + m_3 r_3^2 + m_2 l_1 r_2 c_2 + m_3 l_1 l_2 c_2 + 2m_3 l_2 r_3 c_3 + m_3 l_1 r_3 c_{23} \\
M_{13} = M_{31} &= I_3 + m_3 r_3^2 + m_3 l_2 r_3 c_3 + m_3 l_1 r_3 c_{23} \\
M_{22} &= I_2 + I_3 + m_2 r_2^2 + m_3 l_2^2 + m_3 r_3^2 + 2m_3 l_2 r_3 c_3 \\
M_{23} = M_{32} &= I_3 + m_3 r_3^2 + m_3 l_2 r_3 c_3 \\
M_{33} &= I_3 + m_3 r_3^2
\end{aligned}
$$

and $c_2 = \cos\theta_2, c_{23} = \cos(\theta_2 + \theta_3), s_2 = \sin\theta_2, s_{23} = \sin(\theta_2 + \theta_3)$, m_i is the mass of link i, l_i is the length of link i, r_i is the distance from joint i to the center of mass of link i, and I_i is the inertia of link i about its center of mass. Link masses and inertias include actuator masses and inertias. Γ is the $3 \times 3 \times 3$ matrix of Christoffel symbols given by

$$
\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i}\right).
$$

For the three joint planar robot, we have

$$
\begin{aligned}
\Gamma_{111} &= 0 \\
\Gamma_{112} = \Gamma_{121} &= -m_2 l_1 r_2 s_2 - m_3 l_1 l_2 s_2 - m_3 l_1 r_3 s_{23} \\
\Gamma_{113} = \Gamma_{131} &= -m_3 l_2 r_3 s_3 - m_3 l_1 r_3 s_{23} \\
\Gamma_{122} &= -m_2 l_1 r_2 s_2 - m_3 l_1 l_2 s_2 - m_3 l_1 r_3 s_{23} \\
\Gamma_{123} = \Gamma_{132} &= -m_3 l_2 r_3 s_3 - m_3 l_1 r_3 s_{23} \\
\Gamma_{133} &= -m_3 l_2 r_3 s_3 - m_3 l_1 r_3 s_{23} \\
\Gamma_{211} &= m_3 l_1 l_2 s_2 + m_2 l_1 r_2 s_2 + m_3 l_1 r_3 s_{23} \\
\Gamma_{212} = \Gamma_{221} &= 0 \\
\Gamma_{213} = \Gamma_{231} &= -m_3 l_2 r_3 s_3 \\
\Gamma_{222} &= 0 \\
\Gamma_{223} = \Gamma_{232} &= -m_3 l_2 r_3 s_3 \\
\Gamma_{233} &= -m_3 l_2 r_3 s_3 \\
\Gamma_{311} &= m_3 l_2 r_3 s_3 + m_3 l_1 r_3 s_{23} \\
\Gamma_{312} = \Gamma_{321} &= m_3 l_2 r_3 s_3 \\
\Gamma_{313} = \Gamma_{331} &= 0 \\
\Gamma_{322} &= m_3 l_2 r_3 s_3 \\
\Gamma_{323} = \Gamma_{332} &= 0 \\
\Gamma_{333} &= 0.
\end{aligned}
$$

The equations can also be written in the form

$$
\mathbf{M}(\Theta)\ddot{\Theta} + \mathbf{C}(\Theta, \dot{\Theta})\dot{\Theta} = \tau,
$$

where $\mathbf{C}$ is the 3×3 Coriolis matrix of elements $\mathbf{C}_{ij} = \Sigma_{k=1}^{n} \Gamma_{ijk} \dot{\theta}_k$.

$$\mathbf{C} = \begin{pmatrix} \Gamma_{111}\dot{\theta}_1 + \Gamma_{112}\dot{\theta}_2 + \Gamma_{113}\dot{\theta}_3 & \Gamma_{121}\dot{\theta}_1 + \Gamma_{122}\dot{\theta}_2 + \Gamma_{123}\dot{\theta}_3 & \Gamma_{131}\dot{\theta}_1 + \Gamma_{132}\dot{\theta}_2 + \Gamma_{133}\dot{\theta}_3 \\ \Gamma_{211}\dot{\theta}_1 + \Gamma_{212}\dot{\theta}_2 + \Gamma_{213}\dot{\theta}_3 & \Gamma_{221}\dot{\theta}_1 + \Gamma_{222}\dot{\theta}_2 + \Gamma_{223}\dot{\theta}_3 & \Gamma_{231}\dot{\theta}_1 + \Gamma_{232}\dot{\theta}_2 + \Gamma_{233}\dot{\theta}_3 \\ \Gamma_{311}\dot{\theta}_1 + \Gamma_{312}\dot{\theta}_2 + \Gamma_{313}\dot{\theta}_3 & \Gamma_{321}\dot{\theta}_1 + \Gamma_{322}\dot{\theta}_2 + \Gamma_{323}\dot{\theta}_3 & \Gamma_{331}\dot{\theta}_1 + \Gamma_{332}\dot{\theta}_2 + \Gamma_{333}\dot{\theta}_3 \end{pmatrix}.$$

Acknowledgments

This work was performed while the first author was an STA postdoctoral fellow at the Biorobotics Division of the Mechanical Engineering Laboratory. We thank the Science and Technology Agency of Japan and the Robotics Department of MEL for their support, the anonymous reviewers and Francesco Bullo and Steve LaValle for their comments, and Costa Nikou for developing the original Tcl/Tk interface.

References

[1] S. K. Agrawal, N. Faiz, and R. M. Murray. Feasible trajectories of linear dynamic systems with inequality constraints using higher-order representations. In *IFAC*, July 1999.

[2] H. Arai. Controllability of a 3-DOF manipulator with a passive joint under a nonholonomic constraint. In *IEEE International Conference on Robotics and Automation*, pages 3707–3713, 1996.

[3] H. Arai and S. Tachi. Position control system of a two degree of freedom manipulator with a passive joint. *IEEE Transactions on Industrial Electronics*, 38(1):15–20, Feb. 1991.

[4] H. Arai, K. Tanie, and N. Shiroma. Nonholonomic control of a three-dof planar underactuated manipulator. *IEEE Transactions on Robotics and Automation*, 14(5):681–695, Oct. 1998.

[5] C. G. Atkeson, C. H. An, and J. M. Hollerbach. Estimation of inertial parameters of manipulator loads and links. In *International Symposium on Robotics Research*, pages 221–228. Cambridge, Mass: MIT Press, 1985.

[6] J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.

[7] M. Bergerman, C. Lee, and Y. Xu. Experimental study of an underactuated manipulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2: 317–322, 1995.

[8] M. Bergerman and Y. Xu. Planning collision-free motions for underactuated manipulators in constrained configuration space. In *IEEE International Conference on Robotics and Automation*, pages 549–555, 1997.

[9] M. D. Berkemeier and R. S. Fearing. Tracking fast inverted trajectories of the underactuated acrobot. *IEEE Transactions on Robotics and Automation*, 15(4):740–750, Aug. 1999.

[10] J. E. Bobrow, S. Dubowsky, and J. S. Gibson. Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research*, 4(3):3–17, Fall 1985.

[11] F. Bullo and K. M. Lynch. Kinematic controllability and decoupled trajectory planning for underactuated mechanical systems. Submitted for publication.

[12] A. De Luca, R. Mattone, and G. Oriolo. Control of underactuated mechanical systems: Applications to the planar 2R robot. In *IEEE International Conference on Decision and Control*, pages 1455–1460, 1996.

[13] N. Faiz and S. K. Agrawal. Optimal planning of an under-actuated planar body using higher order method. In *IEEE International Conference on Robotics and Automation*, pages 736–741, 1998.

[14] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[15] J. Hauser and R. M. Murray. Nonlinear controllers for non-integrable systems: The acrobot example. In *American Control Conference*, pages 669–671, 1990.

[16] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. In *The Fourth Workshop on the Algorithmic Foundations of Robotics*, Boston, MA, 2000. A. K. Peters.

[17] K. Kosuge, K. Furuta, and T. Yokoyama. Virtual internal model following control of robot arms. In *IEEE International Conference on Robotics and Automation*, pages 1549–1554, 1987.

[18] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[19] J.-P. Laumond, P. E. Jacobs, M. Taïx, and R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577–593, Oct. 1994.

[20] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *The Fourth Workshop on the Algorithmic Foundations of Robotics*, Boston, MA, 2000. A. K. Peters.

[21] A. D. Lewis and R. M. Murray. Configuration controllability of simple mechanical control systems. *SIAM Journal on Control and Optimization*, 35(3):766–790, May 1997.

[22] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, Oct. 1979.

[23] K. M. Lynch. Controllability of a planar body with unilateral thrusters. *IEEE Transactions on Automatic Control*, 44(6):1206–1211, June 1999.

[24] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 15(6):533–556, Dec. 1996.

[25] J. Mareczek, M. Buss, and G. Schmidt. Robust global stabilization of the underactuated 2-DOF manipulator R2D1. In *IEEE International Conference on Robotics and Automation*, pages 2640–2645, 1998.

[26] P. Martin, S. Devasia, and B. Paden. A different look at output tracking: Control of a VTOL aircraft. In *IEEE International Conference on Decision and Control*, pages 2376–2381, 1994.

[27] R. M. Murray, M. Rathinam, and W. Sluis. Differential flatness of mechanical control systems: A catalog of prototype systems. In *ASME Int Mech Eng Congress and Expo*, 1995.

[28] Y. Nakamura, T. Suzuki, and M. Koinuma. Nonlinear behavior and control of a nonholonomic free-joint manipulator. *IEEE Transactions on Robotics and Automation*, 13(6):853–862, 1997.

[29] G. Oriolo and Y. Nakamura. Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators. In *Conference on Decision and Control*, pages 2398–2403, 1991.

[30] F. Pfeiffer and R. Johanni. A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, RA-3(2):115–123, 1987.

[31] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, and I. Kolmanovsky. Nonlinear control of a class of underactuated systems. In *IEEE International Conference on Decision and Control*, pages 1682–1687, 1996.

[32] F. Saito, T. Fukuda, and F. Arai. Swing and locomotion control for a two-link brachiation robot. *IEEE Control Systems Magazine*, 14(1):5–12, 1994.

[33] Z. Shiller and S. Dubowsky. On computing the global time-optimal motions of robotic manipulators in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 7(6):785–797, Dec. 1991.

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

[35] K. G. Shin and N. D. McKay. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30(6):531–541, June 1985.

[36] J.-J. E. Slotine and H. S. Yang. Improving the efficiency of time-optimal path-following algorithms. *IEEE Transactions on Robotics and Automation*, 5(1):118–124, Feb. 1989.

[37] O. J. Sørdalen, Y. Nakamura, and W. J. Chung. Design of a nonholonomic manipulator. In *IEEE International Conference on Robotics and Automation*, pages 8–13, 1994.

[38] M. W. Spong. Swing up control of the acrobot. In *IEEE International Conference on Robotics and Automation*, pages 2356–2361, 1994.

[39] H. J. Sussmann. A general theorem on local controllability. *SIAM Journal on Control and Optimization*, 25(1):158–194, Jan. 1987.

[40] T. Suzuki, M. Koinuma, and Y. Nakamura. Chaos and nonlinear control of a nonholonomic free-joint manipulator. In *IEEE International Conference on Robotics and Automation*, pages 2668–2675, 1996.

[41] S. Takashima. Control of gymnast on a high bar. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1424–1429, Osaka, Japan, 1991.

[42] T. Yoshikawa. Analysis and control of robot manipulators with redundancy. In *First International Symposium on Robotics Research*, 1983.