Proceedings of the 34th
Conference on Decision & Control
New Orleans, LA - December 1995

WP02 3:50

JOINT-SPACE TRACKING OF WORKSPACE TRAJECTORIES IN CONTINUOUS TIME

Neil H. Getz* and Jerrold E. Marsden†

ABSTRACT

We present a controller for a class of robotics manipulators which provides exponential convergence to a desired end-effector trajectory using gains specified in joint-space. This is accomplished without appeal to the use of discrete inverse-kinematics algorithms, allowing the controller to be posed entirely in continuous time.

1. Introduction

This paper describes an application of dynamic inversion (see [1] in this proceedings) to the control of robotic manipulators. A dynamic inverter used to solve inverse-kinematics is combined with a tracking controller. The combined system is a dynamic controller for end-effector tracking, and provides exponentially decaying tracking error.

In Section 2, after some necessary definitions, we precisely define the robotic control problem in which we will be interested. In Section 3 we describe some current methods of robot manipulator control, looking very briefly at some of their strengths and shortcomings. In Section 4 we apply dynamic inversion to construct an exact tracking controller for the tracking of end-effector trajectories. In Section 5 an example of output tracking for a simple model of a two-link robot arm is used to illustrate the application of the implicit tracking theorem.

2. Problem Definition

Let the joint angles of the robotic manipulator be denoted $\theta \in \mathbb{R}^n$, and the corresponding generalized torques[1] be $\tau \in \mathbb{R}^n$. We will concern ourselves with the control of open-chain robotic manipulators[2] having equations of motion of the form

$$M(\theta)\ddot{\theta} = K(\theta, \dot{\theta}) + \tau \tag{1}$$

where the inertia matrix $M(\theta) \in \mathbb{R}^{n\times n}$ is positive definite and symmetric for all $\theta \in \mathbb{R}^n$. The vector $K(\theta, \dot{\theta})$ contains all Coriolis, centrifugal, frictional, damping, and gravitational forces.

We will refer to a vector $\theta \in \mathbb{R}^n$, each of whose elements parameterize a single joint of a manipulator, as the **configuration** of the manipulator, and the space of all configuration vectors as the **joint-space.** We will refer to a vector x, which parameterizes the position and orientation of the end-effector of the manipulator with respect to a fixed reference frame, as the **pose** of the end-effector and let $\mathcal{X}$ be the set of all such poses. Let $\mathcal{F} : \mathbb{R}^n \to \mathcal{X}; \theta \mapsto \mathcal{F}(\theta)$ denote the **forward-kinematics map** taking a configuration θ to its corresponding end-effector pose x. Depending upon the particular manipulator, x may take values in various sections of $SE(3)$, the group of positions and orientations in Euclidean space. We will view $\mathcal{X}$ through a coordinate chart from $\mathbb{R}^n$. The map $\mathcal{F}(\theta)$ will be assumed to be C^2. The codomain of $\mathcal{F}$ is the **workspace.**

The workspace tracking problem considered here is as follows:

Problem 2.1 Find a control $\tau(\theta, t)$ such that for all initial configurations $\theta_0 \in \mathbb{R}^n$ in an open subset of $\mathbb{R}^n$, the pose $x(t)$ of the end-effector converges exponentially to the desired end-effector trajectory $x_d(t)$. △

Assumption 2.2 Assume that the desired end-effector trajectory $x_d(t)$ is C^4, that the forward-kinematic map $\mathcal{F}(\theta)$ is also C^4, that $D\mathcal{F}(\theta)$ and its inverse are bounded, and that for all $z \in \mathcal{B}_r$, $D^2\mathcal{F}(z + \theta_*(t))$ is bounded. △

Given a particular end-effector pose $x_p \in \mathcal{X}$, the **inverse-kinematics problem** is to find θ_p satisfying $x_p = \mathcal{F}(\theta_p)$. In general, multiple solutions θ_p exist. We will restrict the space from which we draw desired output trajectories $x_d(t)$, $t \in \mathbb{R}$, to those output trajectories which have corresponding continuous isolated inverse-kinematic solutions $\theta_*(t)$. For simplicity we concerned ourselves with manipulators for which

* Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, California 94720, getz@eecs.berkeley.edu.

† Control and Dynamical Systems, California Institute of Technology, Pasadena, California, marsden@cds.caltech.edu.

[1] We will use the term "torques" to mean both control forces and control torques.

[2] By *open-chain* robotic manipulator, we mean a finite sequence of rigid links, the first link being hinged to the ground, with all successive links hinged to the previous link by a joint. The end of the last link is presumed to be free to move in the workspace.

the dimensions of the joint-space and workspace are equal, and for which $D\mathcal{F}(\theta)$ is nonsingular almost everywhere. For such manipulators, the inverse function theorem tells us that for a pose trajectory $x_d(t)$, $t \in \mathbb{R}_+$, which never leaves the workspace, if $\theta_*(t)$ satisfies $\mathcal{F}(\theta_*(t)) = x_d(t)$ for all $t \geq 0$, and if for each t, $D\mathcal{F}(\theta_*(t))$ is non-singular, then $\theta_*(t)$ is an *isolated* inverse-kinematic solution corresponding to $x_d(t)$.

Since there are, in general, multiple continuous isolated solutions $\theta_*(t)$ of $\mathcal{F}(\theta) = x_d(t)$ we will assume that a particular one has been chosen. It is only a matter of choice of initial conditions for the tracking controller described below (Proposition 4.1) that will cause the manipulator to follow one inverse-kinematic solution over another.

3. Manipulator Tracking-Control Methodologies

Current techniques of tracking control for robotic manipulators can be divided roughly into two classes according to whether the controller is realized in terms of workspace coordinates or joint-space coordinates. These classes are as follows:

1. Joint-Space Control of Joint-Space Trajectories. A discrete inverse-kinematics algorithm is applied to a time-parameterized sequence of chosen points $\{x_d(t_k)\}$, called **via points,** along a continuous desired pose trajectory $t \mapsto x_d(t) \in \mathcal{X}$. This discrete inversion produces a corresponding time-parameterized sequence of joint-angle vectors $\{\theta_d(t_k)\}$. One may then create, via a spline, a smooth time-parameterized curve $\tilde{\theta}(t) \in \mathbb{R}^n$ through the sequence $\{\theta_d(t_k)\}$, and then track $\tilde{\theta}(t)$ using a tracking control algorithm described in terms of $\ddot{\tilde{\theta}}(t)$, $\dot{\tilde{\theta}}(t)$, and $\tilde{\theta}(t)$. See [2, 3].

2. Workspace Control of Workspace Trajectories. One transforms the dynamic equations of the robot into workspace coordinates by differentiating $x = \mathcal{F}(\theta)$ twice,

$$\begin{aligned} \dot{x} &= D\mathcal{F}(\theta)\dot{\theta} \\ \ddot{x} &= \textstyle\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta)\dot{\theta}_i\dot{\theta} + D\mathcal{F}(\theta)\ddot{\theta} \end{aligned} , \qquad (2)$$

solving for $\ddot{\theta}$,

$$\ddot{\theta} = D\mathcal{F}(\theta)^{-1}\left(\ddot{x} - \sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta)\dot{\theta}_i\dot{\theta}\right) \qquad (3)$$

substituting the result for $\ddot{\theta}$ into the manipulators dynamical equations (1)

$$\begin{aligned} &M(\theta)D\mathcal{F}(\theta)^{-1}\ddot{x} = \\ &D\mathcal{F}(\theta)^{-1}\left(\textstyle\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta)\dot{\theta}_i\dot{\theta}\right) K(\theta, \dot{\theta}) + \tau, \end{aligned} \qquad (4)$$

and left multiplying by $(D\mathcal{F}(\theta)^{-1})^T$,

$$\begin{aligned} &(D\mathcal{F}(\theta)^{-1})^T M(\theta) D\mathcal{F}(\theta)^{-1}\ddot{x} = (D\mathcal{F}(\theta)^{-1})^T \cdot \\ &\left(D\mathcal{F}(\theta)^{-1}\left(\textstyle\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta)\dot{\theta}_i\dot{\theta}\right) K(\theta, \dot{\theta})\right) \\ &+(D\mathcal{F}(\theta)^{-1})^T\tau \end{aligned} \qquad (5)$$

One then chooses gain matrices B^1 and B^2 for error feedback in terms of workspace errors $x - x_d(t)$ and $\dot{x} - \dot{x}_d(t)$, to obtain a tracking controller for tracking the desired $x_d(t) \in \mathcal{X}$ as in

$$\begin{aligned} \tau = & -D\mathcal{F}(\theta)^{-1}\left(\textstyle\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta)\dot{\theta}_i\dot{\theta}\right) K(\theta, \dot{\theta}) \\ & +M(\theta)D\mathcal{F}(\theta)^{-1}v, \\ v = & \ \ddot{x}_d(t) - B^2(\dot{x} - \dot{x}_d(t)) - B^1(x - x_d(t)). \end{aligned} \qquad (6)$$

See [4].

Each of the above approaches has its advantages, but neither is entirely satisfactory for all robotic manipulator tasks. In the first class of controllers, if accuracy of end-effector pose is to be achieved, one must solve a great number of individual inverse-kinematic problems in order to find the corresponding sequence of points in joint-space. If a disturbance causes the end-effector to move substantially from its desired trajectory, a new sequence of workspace points may have to be inverted in order to fulfill desired error dynamics. The joint-space spline from one via point to the next may correspond to a workspace path that diverges substantially from the desired workspace trajectory for points midway between the via points. This can cause a lack of uniformity in the workspace tracking error. This approach has also necessitated a combined discrete-time, continuous-time approach to workspace tracking control of robotic manipulators. Intrinsically, it is not "real-time" since the next via point in the joint-space must be determined before the spline from the previous via point can be created, time-parameterized, and tracked. Joint-space control does have an advantage in that joint parameterizations are global. Thus one need not change coordinates in the middle of a control task.

In the second class of controllers control gains are posed in terms of the workspace errors $x - x_d$, and $\dot{x} - \dot{x}_d$. Using this approach one need not solve for an inverse-kinematic solution, though $D\mathcal{F}(\theta)$ must be inverted. This method too can be undesirable since the inputs to the manipulator are often joint torques. Avoidance of saturation of the joint torques, for instance, is made difficult. If $D\mathcal{F}(\theta)$ cannot be conveniently inverted symbolically, then, once again, a mixed discrete and continuous time control scheme is necessitated by the use of numerical matrix inversion. In addition, since the workspace is usually $SE(3)$, and since no global parameterization of $SE(3)$ exists, this approach can necessitate the overhead of coordinate changes in the controller implementation. However, specifying control gains in the workspace coordinates can be advantageous for certain combinations of manipulator and task.

This paper describes an alternative to the two classes of tracking controllers described above based on dynamic inversion [5, 6] and the implicit tracking results from [7]. Our method allows one to pose the controller and the feedback errors in the joint-space while continuously providing an estimate of $\theta_*(t)$ satisfying $x_d(t) = \mathcal{F}(\theta_*(t))$. This allows continuous-time control in the joint-space. The continuous-time approach also has the virtue of a degree of independence of choice of computational machinery. For realization of the control via digital computer one must choose an integrator in order to integrate the dynamic inverter. The issue of accuracy, however, is made solely a matter of the choice of integrator. Using our method, we also retain the advantage of global control coordinates.

4. Workspace Trajectory Tracking

We now apply dynamic inversion to the problem of tracking workspace trajectories. Given a desired end-effector trajectory $t \mapsto x_d(t)$, the inverse-kinematic solution $\theta_*(t)$ to $\mathcal{F}(\theta) = x_d(t)$ is defined implicitly as a continuous isolated solution of $F(\theta, t) = 0$ where

$$F(\theta, t) := \mathcal{F}(\theta) - x_d(t). \tag{7}$$

The use of dynamic inversion for the tracking of implicitly defined trajectories is described in [6, 7]. Those arguments will be specialized here to the case of robotic manipulator control.

From the form of the manipulator dynamical equations (1), and the assumption that $M(\theta)$ is non-singular, it is clear by substitution into (1) that the feedback torque

$$\tau = -K(\theta, \dot{\theta}) + M(\theta)v \tag{8}$$

causes the controlled manipulator dynamics

$$\ddot{\theta} = v. \tag{9}$$

to be linear from input to state, as well as decoupled[3]. Let $e(t) := \theta - \theta_*(t)$ denote the tracking error between the manipulator configuration $\theta(t)$ and the inverse-kinematic solution $\theta_*(t)$ at time t. Let $\beta_i^2, \beta_i^1 \in \mathbb{R}$, $i \in \underline{n}$ be a choice of fixed numbers such that the roots of the polynomial in s,

$$s^2 + \beta_i^2 s + \beta_i^1, \quad i \in \underline{n} \tag{10}$$

have strictly negative real parts. Suppose we had explicit signals $\theta_*(t)$, $\dot{\theta}_*(t)$, and $\ddot{\theta}_*(t)$. Choosing v as

$$v_i := \ddot{\theta}_*(t)_i - \beta_i^2(\dot{\theta}_i - \dot{\theta}_*(t)_i) - \beta_i^1(\theta_i - \theta_*(t)_i) \tag{11}$$

[3] By the dynamics being *decoupled* we mean that for each $i \in \underline{p}$, $\ddot{\theta}_i = v_i$ where each v_i is distinct. Thus each θ_i may be made to track any $y_{di}(t)$ which is C^2.

results in controlled manipulator dynamics having exponentially stable tracking error.

If the trajectory $\theta_*(t)$ were given explicitly, our job would be over. However, we do not have explicit expressions for $\theta_*(t)$, $\dot{\theta}_*(t)$, and $\ddot{\theta}_*(t)$ since we do not have an explicit expression for $\theta_*(t)$. We will construct estimators for $\dot{\theta}_*$ and $\ddot{\theta}_*$ that will depend upon the state $\hat{\theta}$ of a dynamic inverter as well as the desired workspace trajectory $x_d(t)$. We will substitute those estimators for θ_* and $\dot{\theta}_*$ in 11.

We may construct estimators for the time derivatives of $\theta_*(t)$ as follows: Differentiate $\mathcal{F}(\theta_*) - x_d(t) = 0$ with respect to t to obtain

$$D\mathcal{F}(\theta_*)\dot{\theta}_* - \dot{x}_d(t) = 0. \tag{12}$$

Let Γ_* be the solution to

$$D\mathcal{F}(\theta)\Gamma - I = 0. \tag{13}$$

Solve (12) for $\dot{\theta}_*$, and substitute Γ for $D\mathcal{F}(\theta_*)^{-1}$, and $E^1(\Gamma, t)$ for $\dot{\theta}_*$ to get

$$E^1(\Gamma, t) := \Gamma \dot{x}_d(t). \tag{14}$$

To obtain an estimator for $\ddot{\theta}_*$, differentiate (12) with respect to t giving

$$\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta_*)\dot{\theta}_{*i}\dot{\theta}_* - D\mathcal{F}(\theta_*)\ddot{\theta}_* - \ddot{x}_d(t) = 0. \tag{15}$$

Solve for $\ddot{\theta}_*$ and replace θ by θ_*, $\dot{\theta}_*$ by $E^1(\Gamma, t)$, and $\ddot{\theta}_*$ by $E^2(\Gamma, \theta, t)$ to get

$$\begin{aligned} &E^2(\Gamma, \theta, t) = \\ &-\Gamma\left(\ddot{x}_d - \textstyle\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} D\mathcal{F}(\theta) E_i^1(\Gamma, t) E^1(\Gamma, t)\right). \end{aligned} \tag{16}$$

Note that $E_i^1(\Gamma_*, t) = \dot{\theta}_*$, and $E^2(\Gamma_*, \theta_*, t) = \ddot{\theta}_*$. Also, by Assumption 2.2, $E^1(\Gamma_*, t)$ and $E^2(\Gamma_*, t)$ are C^2 in their arguments.

Now let

$$\hat{v}_i := E_i^2(\hat{\Gamma}, \hat{\theta}, t) - \beta_i^2(\dot{\theta} - E_i^1(\hat{\Gamma}, t)) - \beta_i^1(\theta - \hat{\theta}) \tag{17}$$

where we denote the estimators for Γ and θ by $\hat{\Gamma}$ and $\hat{\theta}$ respectively.

In order to dynamically estimate a linear dynamic inverse for $F(\theta, t)$, we define

$$F^\Gamma(\Gamma, \theta) := D\mathcal{F}(\hat{\theta})\Gamma - I \tag{18}$$

with $F^\Gamma(\Gamma, \theta) \in \mathbb{R}^{n\times n}$. The dynamic inverse of $F^\Gamma(\Gamma, \theta)$ is $G : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined by

$$G^\Gamma(w, \Gamma) = \Gamma \cdot w. \tag{19}$$

We obtain an estimator for $\dot{\Gamma}_*$ by differentiating 13, solving for $\dot{\Gamma}_*$, and replacing $D\mathcal{F}(\theta_*)^{-1}$ to get

$$E^{\Gamma}(\Gamma, \theta, t) := -\Gamma\ (d/dt)D\mathcal{F}(\theta_*)|_{\dot{\theta}_* = E^1(\Gamma,t)}\ \Gamma. \tag{20}$$

Combining estimation, control, and manipulator dynamics we make the following claim.

Proposition 4.1 *Let $\mathcal{F}(\theta)$ and $x_d(t) \in \mathcal{X}$ be C^4. Let $\theta_*(t)$ be a continuous isolated solution of $\mathcal{F}(\theta) = x_d(t)$. Let $D\mathcal{F}(\theta_*(t))$ and its inverse be bounded for all t, and for all $z \in \mathcal{B}_r$, $r > 0$, let $D^2\mathcal{F}(z + \theta_*(t))$ be bounded. Let $B^j := \operatorname{diag}(\beta_1^j, \ldots, \beta_n^j)$, $j \in \{1, 2\}$. Then the control system*

$$M(\theta)\ddot{\theta} + K(\theta, \dot{\theta}) = \tau \tag{21}$$

$$\tau = -K(\theta, \dot{\theta}) + M(\theta)v(\hat{\Gamma}, \hat{\theta}, t) \tag{22}$$

$$\begin{aligned} v(\hat{\Gamma}, \hat{\theta}, t) \quad &= E^2(\hat{\Gamma}, \hat{\theta}, t) \\ &\quad -B^1(\dot{\theta} - E^1(\hat{\Gamma}, t)) - B^0(\theta - \hat{\theta}), \end{aligned} \tag{23}$$

$$\begin{aligned} \begin{bmatrix} \dot{\hat{\Gamma}} \\ \dot{\hat{\theta}} \end{bmatrix} &= -\mu \begin{bmatrix} \hat{\Gamma} \cdot \left(D\mathcal{F}(\hat{\theta})\hat{\Gamma} - I \right) \\ \hat{\Gamma} \cdot \left(\mathcal{F}(\hat{\theta}) - x_d(t) \right) \end{bmatrix} \\ &\quad + \begin{bmatrix} -\Gamma\ \frac{d}{dt} D\mathcal{F}(\theta_*)\big|_{\dot{\theta}_* = E^1(\hat{\Gamma},t)}\ \hat{\Gamma} \\ \hat{\Gamma}\dot{x}_d(t) \end{bmatrix} \end{aligned} \tag{24}$$

where $E^1(\hat{\Gamma}, t)$ and $E^2(\hat{\Gamma}, \hat{\theta}, t)$ are given by (14) and (16) respectively, causes $\theta(t)$ to converge to $\theta_*(t)$ exponentially. $\diamond$

Proof: This is a straightforward application of the implicit tracking theorem, Theorem 2 of [7]. See also [6]. $\square$

Remark 4.2 Equations (21) are the equations of motion for the manipulator. Equations (24) provide exponentially convergent estimates $\hat{\Gamma}(t)$ and $\hat{\theta}(t)$ of $\Gamma_*(t)$ and $\theta_*(t)$, Equations (22) and (23) determine the input τ as a function of θ, $\hat{\Gamma}$, and $\hat{\theta}$. $\triangle$

5. A Two-Link Example

In this section we work through an example of the application of Proposition 4.1 to the control of a simple model of a two-link robotic arm diagrammed in Figure 1. The links of the robot arm are assumed rigid and of length l_1 and l_2. The masses of each link are assumed, for simplicity, to be point masses m_1 and m_2 located at the distal ends of link 1 and link 2 respectively. The desired position of the end-effector at time t is $x_d(t)$. The actual position is $x(t)$. We wish to make the end-effector (end of the second link)

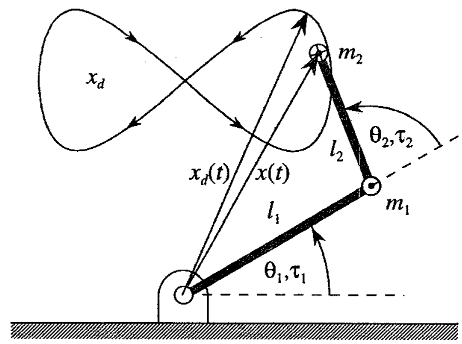

Figure 1: A two-link robot arm with joint angles $\theta = (\theta_1, \theta_2)$, joint torques $\tau = (\tau_1, \tau_2)$, end-effector position x, desired end-effector position x_d, link lengths l_1 and l_2, and link masses m_1 and m_2.

track a prescribed trajectory $x_d(t)$ in the Euclidean plane. The joint-space of the arm is parameterized by $\theta \in \mathbb{T}^2$ where $\mathbb{T}^2$ is the two-torus. For our purposes we may view $\mathbb{T}^2$ through a single chart from $\mathbb{R}^2$. We will assume that we may exert a control torque at each joint and will denote the vector of input torques by $\tau \in \mathbb{R}^2$. In this case $\mathcal{F} : \mathbb{R}^2 \to \mathbb{R}^2; \theta \mapsto \mathcal{F}(\theta)$ maps the configuration space to the Euclidean plane. Let $c_i := \cos(\theta_i)$, $c_{ij} := \cos(\theta_i + \theta_j)$, $s_i := \sin(\theta_i)$, and $s_{ij} := \sin(\theta_i + \theta_j)$, with $i, j \in \{1, 2\}$. For the two-link arm, the forward-kinematics map is

$$\mathcal{F}(\theta) = \begin{bmatrix} l_1c_1 + l_2c_{12} \\ l_1s_1 + l_2s_{12} \end{bmatrix}. \tag{25}$$

The workspace of the two-link robot arm is the range of $\mathcal{F}$, namely $\{x \in \mathbb{R}^2 : x = \mathcal{F}(\theta), \theta \in \mathbb{T}^2\}$. We wish to determine a τ such that the end-effector position $x(t) = \mathcal{F}(\theta(t))$ converges to the desired end-effector position $x_d(t)$.

The equations of motion for the two link manipulator (see [2], Section 6.8) are

$$M(\theta)\ddot{\theta} = -\left(V(\theta, \dot{\theta}) + W(\theta) \right) + \tau \tag{26}$$

where

$$\begin{aligned} &M_{11}(\theta) = l_2^2m_2 + 2l_1l_2m_2c_2 + l_1^2(m_1 + m_2) \\ &M_{12} = M_{21} = l_2^2m_2 + l_1l_2m_2c_2 \\ &M_{22} = l_2^2m_2, \end{aligned} \tag{27}$$

$$V(\theta, \dot{\theta}) = \begin{bmatrix} -m_2l_1l_2s_2\dot{\theta}_2^2 - 2m_2l_1l_2s_2\dot{\theta}_1\dot{\theta}_2 \\ m_2l_1l_2s_2\dot{\theta}_1^2 \end{bmatrix}, \tag{28}$$

and

$$W(\theta) = \begin{bmatrix} m_2l_2gc_{12} + (m_1 + m_2)l_1gc_1 \\ m_2l_2gc_{12} \end{bmatrix}. \tag{29}$$

The matrix $M(\theta)$ is a positive definite symmetric mass matrix, $V(\theta,\dot{\theta})$ is the vector of centrifugal and Coriolis forces on the manipulator, and $W(\theta)$ is the gravitational force on the arm. Let $\theta_*(t)$ be the solution of $F(\theta,t)=0$, where $F(\theta,t) := \mathcal{F}(\theta) - x_d(t)$. If we knew $\theta_*(t)$, $\dot{\theta}_*(t)$, and $\ddot{\theta}_*(t)$ we could set

$$\begin{aligned}\tau &= V(\theta,\dot{\theta}) + W(\theta)\\ &\quad + M(\theta)\left(\ddot{\theta}_* - B^2(\dot{\theta}-\dot{\theta}_*) - B^1(\theta-\theta_*)\right)\end{aligned} \tag{30}$$

where $B^1 = \mathrm{diag}(\beta_1^1,\beta_2^1)$ and $B^2 = \mathrm{diag}(\beta_1^2,\beta_2^2)$, But, for generality, we will assume that we don't know θ_* or its derivatives. We will use dynamic inversion to obtain estimators for these quantities.g

In the case of our two-link robotic arm, closed form solutions for the inverse-kinematics exist (see Craig [2], p.122) and are known. For demonstration purposes we will use dynamic inversion to invert the kinematics and we will use the closed form of the inverse-kinematic to check our results. As long as x_d is kept away from the boundary of the workspace, the two possible inverse-kinematic solutions of $F(\theta,t)=0$ never intersect. We will choose one inverse-kinematic solution, by our choice of initial conditions for dynamic inversion, and track it.

For the two-link arm we have as an estimator for $\dot{\theta}_*(t)$,

$$E^1(\Gamma,t) := \Gamma \cdot \dot{x}_d(t). \tag{31}$$

An estimator for $\ddot{\theta}(t)$ is

$$\begin{aligned}&E^2(\Gamma,\theta,t) :=\\ &\Gamma\left(\ddot{x}_d - \left(\tfrac{d}{dt}D\mathcal{F}(\theta)\right)\big|_{\dot{\theta}=E^1(\Gamma,t)} E^1(\Gamma,t)\right)\end{aligned} \tag{32}$$

where

$$\begin{aligned}&\left(\tfrac{d}{dt}D\mathcal{F}(\theta)\right)\big|_{\dot{\theta}=E^1(\Gamma,t)} =\\ &\begin{bmatrix} -l_1c_1 - l_2c_{12} & -l_2c_{12}\\ -l_1s_1 - l_2s_{12} & -l_2s_{12}\end{bmatrix} E_1^1(\Gamma,t)\\ &+\begin{bmatrix} -l_2c_{12} & -l_2c_{12}\\ -l_2s_{12} & -l_2s_{12}\end{bmatrix} E_2^1(\Gamma,t)\end{aligned} \tag{33}$$

Let $\hat{c}_i = \cos(\hat{\theta}_i)$, $\hat{s}_i = \sin(\hat{\theta}_i)$, $\hat{c}_{12} = \cos(\hat{\theta}_1+\hat{\theta}_2)$, and $\hat{s}_{12} = \sin(\hat{\theta}_1+\hat{\theta}_2)$. A dynamic inverter for this problem is

$$\left\{\begin{aligned}\dot{\hat{\Gamma}} &= -\mu\hat{\Gamma}\left(\begin{bmatrix} -l_1\hat{s}_1 - l_2\hat{s}_{12} & -l_2\hat{s}_{12}\\ l_1\hat{c}_1 + l_2\hat{c}_{12} & l_2\hat{c}_{12}\end{bmatrix}\hat{\Gamma} - I\right)\\ &\quad -\hat{\Gamma}\left(\tfrac{d}{dt}D\mathcal{F}(\theta)\right)\big|_{\dot{\theta}=E^1(\hat{\Gamma},t)} E^1(\hat{\Gamma},t)\hat{\Gamma}\\ \dot{\hat{\theta}} &= -\mu\hat{\Gamma}\left(\begin{bmatrix} l_1\hat{c}_1 + l_2\hat{c}_{12}\\ l_1\hat{s}_1 + l_2\hat{s}_{12}\end{bmatrix} - x_d(t)\right)\\ &\quad +\hat{\Gamma}\dot{x}_d(t)\end{aligned}\right. \tag{34}$$

Combining (34) with the control law

$$\begin{aligned}&\tilde{\tau}(\theta,\hat{\theta},t) = V(\theta,\dot{\theta}) + W(\theta) + M(\theta)\cdot\\ &\left(E^2(\hat{\Gamma},\hat{\theta},t) - B^2(\dot{\theta} - E^1(\hat{\Gamma},t)) - B^1(\theta-\hat{\theta})\right)\end{aligned} \tag{35}$$

gives our controller.

Simulation. We choose $x_d(t)$ to be a time-parameterized figure-eight in the workspace,

$$x_d(t) = [3.75\cos(\pi t), 2 + 1.5\sin(2\pi t)]^T. \tag{36}$$

Figures 2 through 4 show the results of a simulation. The integration was performed in Matlab [8] using the adaptive step size Runge-Kutta integrator `ode45`. The parameters used in the simulation are $B_1 = B_2 = I$, $\mu = 10$, $l_1 = 3$[m], $l_2 = 2$[m], $m_1 = m_2 = 1$[kg], $g = 9.8$[m/s^2]. The initial conditions were $\dot{\theta}(0) = [0,\pi/2]^T$, $\theta(0) = [\pi,-\pi/2]^T$, $\dot{\theta}(0) = [0,0]^T$, and

$$\hat{\Gamma}(0) = \begin{bmatrix} 0 & 1/3\\ -1/2 & 1/3\end{bmatrix}. \tag{37}$$

Figure 2 shows the resulting end-effector path $\mathcal{F}(\theta)$

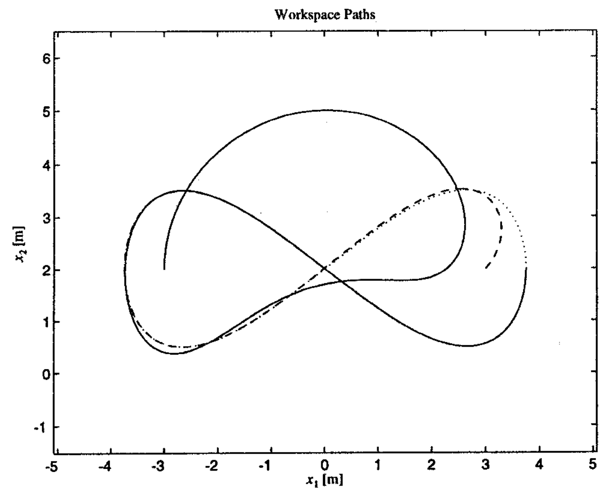

Figure 2: Workspace paths: $\mathcal{F}(\theta)$ (solid), $\mathcal{F}(\hat{\theta})$ (dashed), and $\mathcal{F}(\theta_*)$ (dotted).

(solid), desired path $x_d = \mathcal{F}(\theta_*)$ (dotted), and the image $\mathcal{F}(\hat{\theta})$ of the estimator $\hat{\theta}$ for θ_* through the forward-kinematics map $\mathcal{F}$ (dashed). Both $\mathcal{F}(\hat{\theta})$ and the path of the end-effector $x(t)$ can be seen to converge to the desired path. Figure 3 shows a similar picture, but in joint space. Again, the convergence of both the estimator $\hat{\theta}$ for the inverse-kinematic solution, and the actual joint-angles θ to the inverse-kinematic solution corresponding to the desired trajectory can be seen. Figure 4 shows the norm of the estimation error $\hat{\theta}(t) - \theta(t)$, (top) and of the tracking error $[\theta(t),\dot{\theta}(t)]^T - [\theta_*(t),\dot{\theta}_*(t)]^T$ (bottom) graphed versus time.

The particular inverse-kinematic solution chosen was due to the choice of $\hat{\Gamma}(0)$. We may cause the arm to track the other inverse-kinematic solution simply by choosing a different set of initial conditions for the dynamic inverter (See [6], Chapter 5).

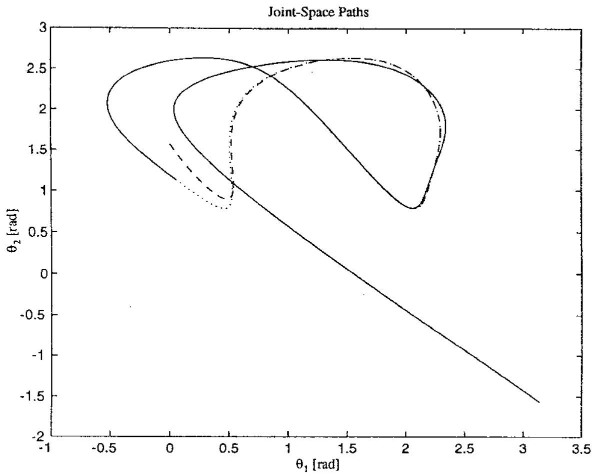

Figure 3: Joint space paths: θ (solid), $\hat{\theta}$ (dashed), and θ_* (dotted).

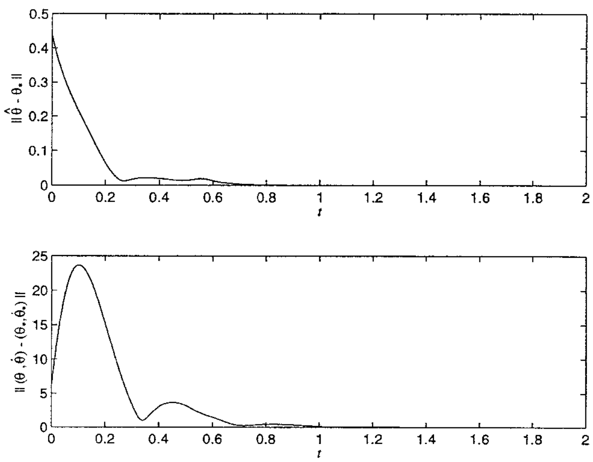

Figure 4: Error norms, $||\hat{\theta}(t) - \theta_*(t)||_2$ (top), and $||(\theta(t), \dot{\theta}(t)) - (\theta_*(t), \dot{\theta}_*(t))||_2$ (bottom).

6. Summary

The implicit tracking controller of [7, 6] has been applied to the robot control problem of tracking of workspace trajectories using joint-space control. This approach provides exponentially convergent tracking of the inverse-kinematic solution corresponding to a continuous end-effector trajectory in the workspace. This results in exponential tracking of the desired end-effector path in the workspace. The controller has been posed in continuous time, using a dynamic inverter to produce approximations of the joint-space signals necessary for control.

A similar continuous-time dynamical handling of inverse-kinematics has been presented by Nicosia et al. in [9]. Their approach fits well into the framework of dynamic inversion. Both $G(w, \theta) = D\mathcal{F}(\theta)^{-1} \cdot w$ and $G(w, \theta) = D\mathcal{F}(\theta)^T$ are used by those authors as dynamic inverses. Derivative estimation similar to that used above is also used by Nicosia et al. [9], though rather than assuming knowledge of the derivatives of the desired end-effector trajectory as we have done, they use an observer to estimate those derivatives. They also rely upon the availability of $D\mathcal{F}(\theta)^{-1}$. Though such reliance is often feasible in practice, we have avoided it, relying instead upon dynamic estimation of a dynamic inverse. This has allowed us to keep all inverse computation in continuous-time rather than having to rely upon discrete matrix inversion routines, letting us avoid a mixed discrete and continuous time control approach.

Though the two-link robot arm of Section 5 had simple rotary joints, it should be kept in mind that dynamic inversion may be used for inverse-kinematics of manipulators with more complex joint geometries. As long as our assumption on the rank and smoothness of $\mathcal{F}(\theta)$ hold, our approach will work for such manipulators.

Acknowledgements. The authors are grateful to C.A. Desoer for his comments and advice.

7. References

[1] N. H. Getz and J. E. Marsden, "A dynamic inverse for nonlinear maps," in *34th IEEE Conference on Decision and Control*, (New Orleans), IEEE, 13-15 December 1995.

[2] J. Craig, *Introduction to Robotics, Mechanics and Control.* New York: Addison Wesley, second ed., 1989.

[3] M. Spong and M. Vidyasagar, *Robot Dynamics and Control.* Wiley, 1989.

[4] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[5] N. H. Getz and J. E. Marsden, "Dynamic inversion of nonlinear maps," Tech. Rep. 621, Center for Pure and Applied Mathematics, Berkeley, California, 19 December 1994.

[6] N. H. Getz, *Dynamic Inversion of Nonlinear Maps with Applications to Nonlinear Control.* PhD thesis, University of California at Berkeley, October 1995.

[7] N. H. Getz and J. E. Marsden, "Tracking implicit trajectories," in *IFAC Symposium on Nonlinear Control Systems Design*, (Tahoe City), International Federation of Automatic Control, June 1995.

[8] *Matlab.* Natick, Mass.: The MathWorks, Inc., 1992.

[9] S. Nicosia, A. Tornambè, and P. Valigi, "A solution to the generalized problem of nonlinear map inversion," *Systems and Control Letters*, vol. 17, no. 5, 1991.