A general framework for cobot control

R. Brent Gillespie, J. Edward Colgate, Michael Peshkin

IEEE Transactions on Robotics and Automation, Volume: 17 Issue: 4 , Page(s): 391 -401, Aug. 2001

A General Framework for Cobot Control

R. Brent Gillespie, *Member, IEEE*, J. Edward Colgate, *Member, IEEE*, and Michael A. Peshkin, *Member, IEEE*

***Abstract*—A general framework is presented for the design and analysis of cobot controllers. Cobots are inherently passive robots intended for direct collaborative work with a human operator. While a human applies forces and moments, the controller guides motion by tuning the cobot's set of continuously variable transmissions. In this paper, a path-following controller is developed that steers the cobot so as to asymptotically approach and follow a preplanned path. The controller is based on feedback linearization. Generality across cobot architectures is assured by designing the controller in task space and developing transformations between each of four spaces: task space, joint space, a set of coupling spaces, and steering space.**

***Index Terms*—Cobot, feedback linearization, path following, virtual fixtures.**

I. Introduction

THE COBOT is a new type of robot intended for direct collaborative work with a human operator. To complete a manipulation task, the cobot and human grasp a workpiece together and share in the determination of its motion. The cobot, by design, cannot move on its own—it is inherently passive,[1] which confers a degree of safety to the operation. The human operator is responsible for producing the motion of the cobot and workpiece by applying forces and moments. The cobot contributes to the manipulation process by guiding that motion. For example, the cobot can make the workpiece behave as if it were constrained to move along a predefined path. Alternatively, the cobot can allow free motion of the workpiece within a certain region of the workspace and border this region with virtual walls. Most significantly, these walls and other constraint surfaces are defined in software. They are *virtual* fixtures, strategically placed in the shared workspace to assist the human operator in task completion.

A physical fixture or barrier makes its presence known by producing reaction forces when contacted by a workpiece. Likewise, a viable *virtual* fixture must be able to produce reaction forces to prevent workpiece penetration. In teleoperators and haptic interfaces, virtual fixtures are realized through direct actuation: electromagnetic forces act through a mechanical coupling. In contrast, a cobot uses continuously variable transmissions (CVTs) and fixed ground to support a reaction force. The cobot end-effector is coupled to ground through a network of CVTs. The inherent passivity of the CVT is the key to ensuring operator safety: the CVT network can be used to resist applied forces but not to produce output forces.

In actual cobot design, one fewer CVT is used than would be necessary to completely constrain the motion of the end-effector. Thus, there remains one allowed direction of motion, over which the operator has full control. The CVT network cannot resist (or produce) forces parallel to this allowed direction of motion. Instead, the CVT network determines the allowed direction of motion. Specifically, the instantaneous setting of transmission ratios of the CVT network determines an instantaneous allowed direction of motion.

There are two types of CVT used in the construction of cobots. The first is quite simple: a single steered wheel rolling on a planar surface. This *translational* CVT (simply called the wheel) constrains a pair of linear speeds (i.e., $\dot{x}$ and $\dot{y}$ where x and y are Cartesian coordinates of the planar surface). The ratio of these speeds is defined by the heading of the wheel; it is the allowed direction of motion on the rolling surface. Forces perpendicular to the rolling direction are supported by constraint forces. The second type of CVT relates two angular speeds. This *rotational* CVT (or simply CVT) is composed of a sphere caged between two drive rollers and two steering rollers. The CVT constrains the drive roller speeds $\dot{\theta}_1$ and $\dot{\theta}_2$ where θ_1 and θ_2 are the drive roller angular displacements. The ratio of these angular speeds is defined by the (common) angular displacement of the steering rollers; it is the allowed direction of motion in the Cartesian space spanned by θ_1 and θ_2. See the companion paper [1] for a detailed introduction to the CVT.

A. Apparent Degrees of Freedom

Before launching into a discussion of the kinematics and control of cobots, it will be helpful to define the dimension of a cobot's motion space and contrast it to the dimension of its configuration space. Whereas the configuration space[2] is spanned by the generalized coordinates that describe reachable configurations, the motion space is spanned by the generalized coordinate derivatives[3] that describe the allowed motions. The configuration space dimension P is regarded as the minimum number of generalized coordinates needed to uniquely describe configuration. A nonminimal set of N generalized coordinates will be accompanied by M holonomic constraints, where $P = N - M$. Similarly, the motion space dimension p is regarded as the minimum number of generalized coordinate derivatives needed to

Manuscript received March 26, 1999; revised December 13, 1999. This paper was recommended for publication by Associate Editor M. Buehler and Editor S. Salcudean upon evaluation of the reviewers' comments. This work was supported by General Motors Corporation and the National Science Foundation.

R. B. Gillespie is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: brentg@umich.edu).

J. E. Colgate and M. A. Peshkin are with the Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: colgate@nwu.edu; peshkin@nwu.edu).

Publisher Item Identifier S 1042-296X(01)08891-7.

[1]Recently, cobots with power assist have been developed. Although such cobots are not passive, safety is preserved by limiting the size of the power assist motor.

[2]In the sequel, we distinguish between two types of configuration space: the configuration of the cobot mechanism, called "joint space," and the configuration of the cobot end-effector, called "task space."

[3]Or generalized speeds, defined as linear combinations of the generalized coordinate derivatives.

describe motion. When a nonminimal set of n generalized coordinate derivatives is used, there exist m nonholonomic constraints that accompany such description, where $p = n - m$. Note that nonholonomic constraints may be expressed as nonintegrable relationships among the generalized coordinate derivatives.

In this paper, we use the term degree of freedom (DOF) to refer to the dimension of a cobot's *motion* space, i.e., DOF=p. Using this nomenclature we may state: a cobot is a single DOF device. Indeed, that is its essential feature. If a cobot's taskspace dimension is N (and nominal motion space dimension n), it has $n - 1$ CVTs, each providing one nonholonomic constraint that eliminates one DOF from the workpiece. (Cobots that use n CVTs in a parallel construction to make available an internal motion are discussed in [1].) To the user, the workpiece will feel as if it is constrained to move along a line in its configuration space. The speed of motion along that line is the only aspect of motion under control of the user. The orientation of that line is determined by the transmission ratio settings of the CVTs.

By placing the CVT transmission ratios under computer control, the allowed direction of motion may be varied. For example, a steering control algorithm that employs sensed displacement may be used to vary the allowed direction so that the cobot will follow a predefined, arbitrarily shaped path in its configuration space. Yet the user is free to determine the speed along that predefined path. If full configuration sensing is used, this path may be made asymptotically stable, and that is the subject of this paper. This is called a *path-following* controller. In this paper, a path-following controller based on input-to-state linearization [2], [3] is developed.

Perhaps more interesting are feedback control algorithms that use sensing of user-applied force and moment, for these can be used to vary the allowed direction of motion such that the cobot appears to have more than its inherent single DOF. When the cobot behaves (through control) as if it had extra DOF, we use the term "apparent DOF." For example, the cobot can be made to appear as if it had n apparent DOF if the controller steers so as to allow motion in whatever direction the user is pushing. When this controller is active, we say the cobot is in *free mode*. Controllers which realize free mode have been addressed in [4].

Intermediate cases, between a single DOF and n apparent DOF, require both configuration sensing and applied force and moment sensing. The cobot could be constrained to move in a submanifold embedded in its configuration space of any dimension between n and 1. Controllers that realize intermediate apparent DOF will be treated in future papers. By switching between various controllers, unilateral constraints may be realized. Switching between controllers as a function of sensed configuration is the basis for creating virtual fixtures.

In order to treat cobot controller design in a general framework, generic to all cobot architectures, we introduce four abstract spaces in Section II. Two of these spaces, task space and joint space, are familiar in robotics, but the next two, coupling space and steering space, are new. The geometry of curves is developed in each of these spaces and transformations between the spaces are derived. Controller design and analysis takes place in task space as described in Section III. The controller and the transformations are combined in the actual implementation of a cobot controller, as demonstrated by way of example in Section IV.

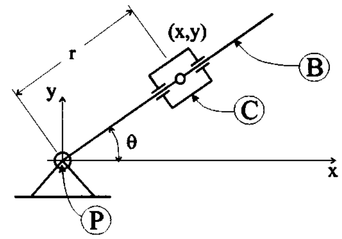

Fig. 1. Schematic of the jib cobot.

II. Cobot Kinematics

For the purpose of motion planning and the construction of virtual fixtures, the focus is on the body of the cobot to which the workpiece is fixed, the end-effector. Although the architecture of the CVT network that constrains the motion of the end-effector will eventually enter the analysis, to begin, we seek a setting in which the motion of the end-effector may be treated independently of its supporting architecture. For such purpose, the configuration space of the end-effector, or task space, denoted C_T, is employed. Cobot controllers are designed and analyzed in C_T-space. All control signals, both inputs and outputs, are expressed in C_T-space variables.

To interpret the control signal for a particular cobot and its network of CVTs, three additional classes of kinematic space are introduced: *joint configuration space*, denoted C_J; a set of *coupling spaces* (one for each CVT), denoted Σ_i; and steering space, denoted Φ. Each space is constructed taking certain portions of a cobot's architecture into account. The coordinate axes of joint space correspond to the joint generalized coordinates (joint angles for CVTs and wheel contact point coordinates for wheels). The coordinate axes of each coupling space correspond to the coordinates whose derivatives are related by the pertinent CVT or wheel. The coordinate axes of steering space correspond to the collection of CVT (or wheel) steering angles. Particularizing the controller design for a given cobot involves the application of transformations between C_T-space, C_J-space, the set of Σ_i-spaces, and Φ-space.

Two examples will help introduce each of the kinematic spaces. Our first example is the jib cobot, shown schematically in the plan view in Fig. 1. The boom B rotates about a vertical axis P while the cart C translates along B. The horizontal plane in which C moves is located overhead so that a load suspended from C may be manipulated at a convenient height above ground by a human operator. We assume here that the load is rigidly coupled to body C so that C may be considered the end-effector. In the typical jib crane, the motions of B and C are not motorized or coupled, so that C has free motion throughout its workspace. The jib cobot, however, features a CVT that couples the translational speed of C to the rotational speed of B. Thus, at any instant, C is only free to move in the direction determined by the setting of the CVT steering angle. Using various algorithms for control of the CVT steering angle, programmable constraints (virtual fixtures) may be placed

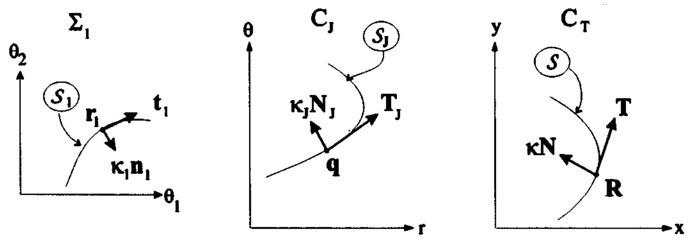

Fig. 2. Three example kinematic spaces: (a) one coupling space (Σ_1); (b) joint space (C_J); and (c) task space (C_T).

in the workspace to assist the operator in the completion of materials handling tasks.

Define (x, y) as the Cartesian coordinates of C and define r as the linear displacement of C from P and θ as the angular displacement of B from the x axis. Then the taskspace C_T of the jib cobot is two-dimensional (2-D), with axes x and y as shown in Fig. 2(c). The jointspace C_J is likewise 2-D with axes r and θ as shown in Fig. 2(b). There is a single coupling space Σ_1 (because there is only one CVT), spanned by the angular displacements of the CVT drive rollers θ_1 and θ_2 as shown in Fig. 2(a). The steering space (not shown) is 1-D.

Curves S, S_J, and S_1 are shown in each space, with position vectors $\mathbf{R}$, $\mathbf{q}$, and $\mathbf{r}_1$ locating points on each curve in C_T, C_J, and Σ_1-space, respectively. The tangent vectors $\mathbf{T}$, $\mathbf{T}_J$, and $\mathbf{t}_1$ and the normal vectors $\mathbf{N}$, $\mathbf{N}_J$, and $\mathbf{n}_1$ scaled by curvature κ, κ_J, and κ_1 are also shown. These vectors are fully defined in the sections below.

Our second example is a three-wheeled cobot known as "Scooter." Fig. 3 shows a plan view of a triangular body A supported by wheels W_1, W_2, and W_3 on a flat horizontal surface. A workpiece is fixed to A and its position and orientation in the horizontal plane are determined collaboratively by an operator and the controller that steers the wheels [4]. Let A_o be the center of body A. Each wheel W_i rolls freely about its horizontal axis but is independently steered with steering angle ϕ_i about a vertical axis P_i $(i = 1, 2, 3)$ fixed in A. Body A is considered the end-effector. The configuration of A may be established using three generalized coordinates: x, y, and θ as shown in Fig. 3. The taskspace C_T spanned by x, y and θ is shown in Fig. 4(c). The "joints" of this cobot are the three wheels. The variables whose derivatives are related by the associated nonholonomic constraints are the Cartesian coordinates of each wheel center x_i, y_i, $(i = 1, 2, 3)$. Thus, the jointspace of this cobot is 6-D, as shown schematically in Fig. 4(b). There are three coupling spaces Σ_i, $(i = 1, 2, 3)$, each spanned by x_i, y_i as shown in Fig. 4(a). The 3-D steering space (not shown) is spanned by the three steering angles ϕ_i, $(i = 1, 2, 3)$. Curves and vectors in Scooter's C_T, C_J, and Σ_1 space are defined in a manner analogous to the previous example and form the basis for the discussion in the following sections.

In the following sections, each kinematic space is introduced in turn, starting with C_T-space in Section II-A. With the introduction of C_J, Σ_i, and Φ-space in Sections II-B–II-D, transformations between each of these and the previously introduced space are developed. These are called the forward transformations. Finally, the inverse transformations are developed in Section II-E. As a final note before launching the development, the construction of all transformations relies on the existence of smooth curves with continuity through at least two differentiations.

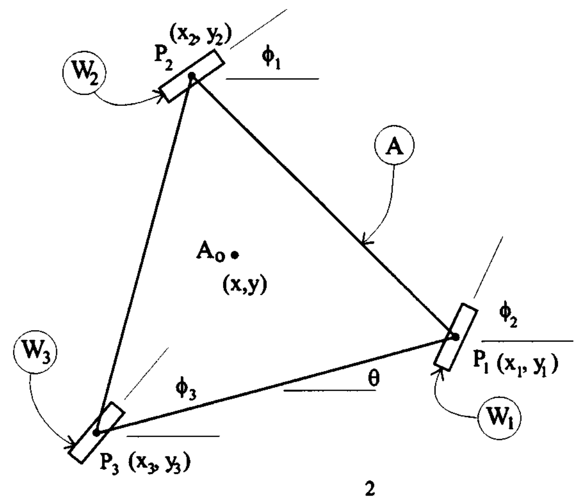

Fig. 3. Schematic of the Scooter: a three-wheel cobot with a three-dimensional task space.

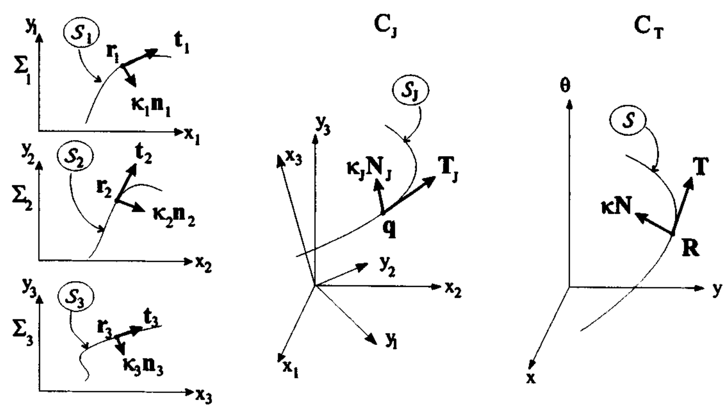

Fig. 4. Three example kinematic spaces: (a) three coupling spaces $(\Sigma_i,\ i = 1, 2, 3)$; (b) joint space (C_J); and (c) task space (C_T).

A. Task Space C_T

Each end-effector configuration (characterized by n generalized coordinate values) corresponds to a point in n-dimensional C_T-space. The vector $\mathbf{R}$ is defined to locate that point; its elements are the end-effector generalized coordinates. As the end-effector configuration evolves, $\mathbf{R}$ traces out a curve S in C_T-space.

By virtue of the underlying CVT network, the motion of the end-effector is subject to $n - 1$ nonholonomic constraints. In C_T-space, these nonholonomic constraints may be interpreted as $n - 1$ linearly independent relationships among the elements of the vector $d\mathbf{R}/dt$, the time-rate of change of $\mathbf{R}$. However, a more useful representation of the influence of the CVT network in C_T-space is available after defining s as the pathlength of S according to

$$ds = (d\mathbf{R}^T\, d\mathbf{R})^{1/2} \tag{1}$$

and expressing the vector $d\mathbf{R}/dt$ as the product of the unit tangent $\mathbf{T} \triangleq d\mathbf{R}/ds$ and the path speed $\dot{s} \triangleq ds/dt$

$$\frac{d\mathbf{R}}{dt} = \mathbf{T}\dot{s}. \tag{2}$$

The influence of the CVT network may now be encapsulated in the unit vector $\mathbf{T}$. The human operator determines the remaining (motion) degree of freedom $\dot{s}$.

Parameterization of C and its associated vectors by the pathlength s proves very useful for stating the cobot controller design problem. It facilitates the separation of the influence of the transmission ratio controller from the influence of the human operator over the end-effector motion. Although the direction $\mathbf{T}$ in C_T-space along which $\mathbf{R}$ may move at a given instant is determined by the instantaneous set of CVT transmission ratios, the speed $\dot{s}$ in the direction of $\mathbf{T}$ is determined by the human operator.

In actual controller implementation, the transmission ratios are not controlled directly; rather their time-derivatives (the steering angular speeds) are controlled. Thus, not only a representation of the influence of the set of transmission ratios, but also a representation of the influence of their derivatives in C_T-space is needed. To this end, $\mathbf{T}$ may be differentiated with respect to s to produce $\kappa\mathbf{N}$, which we call the *curvature vector*

$$\kappa\mathbf{N} = \frac{d\mathbf{T}}{ds}. \tag{3}$$

The vector $\mathbf{N}$ is normal to C at $\mathbf{R}$, of unit length, and orthogonal to $\mathbf{T}$. The scalar $\kappa(s)$ is known as the curvature, while $\mathbf{N}$ is called the unit normal.

The curve S embedded in C_T space, along with the above vectors that describe its geometry, form the basis for the statement of the controller design problem in Section III below. Basically, the controller is responsible for producing the curvature vector and thereby guiding the evolution of $\mathbf{R}$ (equivalently, the tracing of S). The expression of the curvature vector in transmission ratio derivatives, however, requires an accounting of the particular cobot architecture. The spaces C_J and Σ_i are introduced for this purpose, along with transformations between their respective tangent and curvature vectors.

B. Joint Space C_J

A configuration of the entire cobot (rather than simply the end-effector) is represented by a point in its joint configuration space C_J. The vector $\mathbf{q}$ is defined to locate this point, with the generalized coordinates q_i, $(i = 1, \ldots, N)$ of the cobot assembly as its elements. In the case of serial-architecture cobots, q_i are joint angles, whereas in parallel-architecture cobots, q_i are typically Cartesian coordinates of certain points on the cobot body. The number of generalized coordinates used for a cobot, and correspondingly the dimension N associated with C_J space, is oftentimes larger than the dimension n of C_T space. This is due to the presence of dependencies among the generalized coordinates, imposed by rigid body conditions and expressible as holonomic constraints. This is especially true of parallel architecture cobots.

Let s_J be the pathlength of the curve S_J traced in joint space by $\mathbf{q}$. In a manner analogous to their construction in C_T space above, the unit tangent $\mathbf{T}_J$ and the curvature vector $\kappa_J\mathbf{N_J}$ in C_J space may be constructed as follows:

$$\mathbf{T}_J = \frac{d\mathbf{q}}{ds_J}, \qquad \kappa_J\mathbf{N}_J = \frac{d\mathbf{T}_J}{ds_J}. \tag{4}$$

$\boxed{C_J \leftarrow C_T}$ For a nonredundant cobot, each point $\mathbf{q}$ in C_J space corresponds to a point $\mathbf{R}$ in C_T space. Let the inverse kinematics of the cobot relating the assembly generalized coordinates $\mathbf{q}$ to the end-effector generalized coordinates $\mathbf{R}$ be expressed as

$$\mathbf{q} = L(\mathbf{R}). \tag{5}$$

A mapping from the unit tangent $\mathbf{T}$ in C_T space to the unit tangent $\mathbf{T}_J$ in C_J space may be produced by differentiating (5) with respect to s_J:

$$\mathbf{T}_J = \frac{d\mathbf{q}}{ds_J} = \frac{\partial L}{\partial \mathbf{R}}\frac{d\mathbf{R}}{ds}\frac{ds}{ds_J}. \tag{6}$$

The term $\partial L/\partial\mathbf{R}$ is recognized as a Jacobian and denoted $J(\mathbf{R})$ while the term ds/ds_J is a scaling factor which ensures that $\mathbf{T}_J$ is a unit vector

$$\mathbf{T}_J = \frac{J\mathbf{T}}{|J\mathbf{T}|}. \tag{7}$$

The curvature transformation is similar in nature to the tangent transformation. To derive it, we make use of the following relation:

$$\begin{aligned} \text{if} \qquad & \mathbf{A} = \frac{\mathbf{X}}{|\mathbf{X}|} \\ \text{then} \qquad & \mathbf{A}' = \left[\mathbf{I} - \mathbf{A}\mathbf{A}^T\right]\frac{\mathbf{X}'}{|\mathbf{X}|} \end{aligned} \tag{8}$$

where $\mathbf{A}$ and $\mathbf{X}$ are vectors and the prime denotes differentiation. Equations (7) and (8) lead to

$$\kappa_J\mathbf{N}_J = \frac{d\mathbf{T}_J}{ds_j} = \frac{\left[\mathbf{I} - \mathbf{T}_J\mathbf{T}_J^T\right]}{|J\mathbf{T}|^2}\left[\mathbf{T}^T\frac{\partial J}{\partial\mathbf{R}}\mathbf{T} + J\kappa\mathbf{N}\right]. \tag{9}$$

The term $\mathbf{T}^T(\partial J/\partial\mathbf{R})\mathbf{T}$ is shorthand for a column matrix whose ith element is defined as

$$\left\{\mathbf{T}^T\frac{\partial J}{\partial\mathbf{R}}\mathbf{T}\right\}_i = \sum_{j=1}^{n}\left[\sum_{k=1}^{n}\frac{\partial J_{(ij)}}{\partial\mathbf{R}_{(k)}}\mathbf{T}_{(k)}\right]\mathbf{T}_{(j)} \tag{10}$$

where $\mathbf{R}_{(k)}$ denotes the kth element of $\mathbf{R}$, $J_{(ij)}$ denotes the ijth element of J, and so on. The matrix whose kjth element is $\partial J_{(ij)}/\partial\mathbf{R}_{(k)}$ may also be recognized as a Hessian. It is often convenient to express (9) in terms of Hessians, as demonstrated in Section IV.

C. A Set of Coupling Spaces Σ_i

A 2-D coupling space is associated with each CVT or wheel of the cobot. In a nonredundant cobot, there exist $n-1$ such coupling spaces Σ_i, $(i = 1, \ldots, n-1)$ where $n-1$ is the number of CVTs and/or wheels. Each point in Σ_i-space corresponds to a configuration of the ith CVT or wheel. Configuration in this context describes the values of the pair of coordinates whose derivatives are related by the associated tunable nonholonomic constraint. In the case of a rotational CVT, the coordinates in question are the angular displacements of the CVT's drive rollers θ_1, θ_2. In the case of a wheel, the coordinates are the Cartesian coordinates of the wheel's contact point with respect to a frame fixed in its rolling surface x_i, y_i.

The position vector $\mathbf{r}_i$ is defined in Σ_i space to locate the current configuration point. The curve S_i is traced by $\mathbf{r}_i$ with pathlength s_i. The unit tangent $\mathbf{t}_i$ and the curvature vector $\kappa_i\mathbf{n}_i$ in Σ_i space are produced as follows:

$$\mathbf{t}_i = \frac{d\mathbf{r}_i}{ds_i}, \qquad \kappa_i\mathbf{n}_i = \frac{d\mathbf{t}_i}{ds_i}. \tag{11}$$

$\boxed{\Sigma_i \leftarrow C_J}$ By design, each of the two coordinates in a particular Σ_i space is coupled to a certain coordinate in C_J space. The coupling may include a mechanical advantage in the case of a CVT. Let the selection and the coupling be expressed in the relationship

$$\mathbf{r}_i = M_i(\mathbf{q}). \tag{12}$$

As an example, consider a four joint serial-architecture cobot in which CVT 2 couples joint coordinates 1 and 3. Further, assume that the first drive roller of CVT 2 is coupled to joint coordinate q_1 with a mechanical advantage of ρ_1 and that the second drive roller is coupled to joint coordinate q_3 with a mechanical advantage of ρ_2. Then

$$\mathbf{M}_2 = \begin{bmatrix} q_1/\rho_1 & 0 & 0 & 0 \\ 0 & 0 & q_3/\rho_2 & 0 \end{bmatrix}. \tag{13}$$

A transformation may be constructed between $\mathbf{t}_i$ and $\mathbf{T}_J$ by differentiating (12) with respect to pathlength s_i

$$\mathbf{t}_i = \frac{d\mathbf{r}_i}{ds_i} = \frac{D_i\mathbf{T}_J}{|D_i\mathbf{T}_J|} \tag{14}$$

where the factor $D_i \triangleq \partial M_i/\partial \mathbf{q}$ is a Jacobian-type matrix, called the *transmission matrix*. The transmission matrix consists of zeros and mechanical coupling factors. The curvature transformation follows by differentiation of (14) with respect to s_i:

$$\kappa_i\mathbf{n}_i = \frac{d\mathbf{t}_i}{ds_i} = \frac{\left[\mathbf{I} - \mathbf{t}_i\mathbf{t}_i^T\right]}{|D_i\mathbf{T}_J|^2}\left[D_i\kappa_J\mathbf{N}_J\right]. \tag{15}$$

$\boxed{\Sigma_i \leftarrow C_T}$ It is not strictly necessary to define any new transformations relating end-effector space to the coupling spaces, since these may be obtained by concatenating the two sets introduced above. It is often the case, however, that joint space holds little geometric interest. This is especially true in the case of wheeled and parallel-architecture cobots such as Scooter [4], for which the concept of a joint is somewhat abstract.

Fortunately, the transformations from end-effector space directly to coupling space are entirely analogous to those already derived. It is only necessary to define a Jacobian J_i relating incremental end-effector displacements to incremental Σ_i-space displacements. When a well-defined joint space exists, this Jacobian is simply

$$J_i = D_iJ. \tag{16}$$

The essential kinematic transformations are then

$$\mathbf{t}_i = \frac{J_i\mathbf{T}}{|J_i\mathbf{T}|} \tag{17}$$

$$\kappa_i\mathbf{n}_i = \frac{\left[\mathbf{I} - \mathbf{t}_i\mathbf{t}_i^T\right]}{|J_i\mathbf{T}|^2}\left[\mathbf{T}^T\frac{\partial J_i}{\partial \mathbf{R}}\mathbf{T} + J_i\kappa\mathbf{N}\right]. \tag{18}$$

D. Steering Space Φ

To implement a desired curvature in C_T space, it is necessary, ultimately, to compute the steering speeds $\dot{\phi}_i$. We seek a velocity-level forward kinematics relationship between the coordinates of coupling space and the coordinates of steering space. The final set of transformations necessary to compute $\dot{\phi}_i$ are CVT-specific—they depend on the kinematics of the CVTs. We will illustrate two cases: the wheel and the tetrahedral CVT.

Wheel: If the ith coupling space is for a wheel, we define β_i as the angle that the tangent vector $\mathbf{t}_i$ makes with the x_i direction. Then the curvature of the path traced in Σ_i space is

$$\kappa_i = \frac{d\beta_i}{ds_i} \tag{19}$$

where s_i is the pathlength. Now, so long as the rolling wheel does not suffer transverse slip, the wheel heading, given by its steering angle ϕ_i, determines the ratio of linear speeds $\dot{y}_i/\dot{x}_i$ (called the transmission ratio) according to

$$\frac{\dot{y}_i}{\dot{x}_i} = \tan(\phi_i). \tag{20}$$

But since the axes associated with the coupling space of a wheel are x_i and y_i, the tangent $\mathbf{t}_i$ and the wheel heading are one and the same. We have

$$\frac{\dot{y}_i}{\dot{x}_i} = \tan(\phi_i) = \tan(\beta_i) \tag{21}$$

or $\beta_i = \phi_i$. Thus, by (19), we have

$$\dot{\phi}_i = u_i\kappa_i \tag{22}$$

where u_i is the wheel speed, a signed scalar taking on positive values when the inner product of wheel velocity and $\mathbf{t}_i$ is positive. Typically, u_i is a sensed quantity.

Tetrahedral CVT: If, on the other hand, the ith coupling space is for a CVT, we use the θ_{1i} axis and the tangent vector $\mathbf{t}_i$ to define the angle β_i. The curvature in such a coupling space is, as before, $\kappa_i = d\beta_i/ds_i$. The relationship between the transmission ratio $M \triangleq \omega_{2i}/\omega_{1i}$ and the CVT steering angle ϕ_i, however, is significantly more complicated than that for the wheel, owing to the kinematics of the tetrahedral CVT

$$M(\phi) = \frac{\omega_1}{\omega_2} = \frac{\sin(\phi) - \sqrt{2}\cos(\phi)}{\sin(\phi) + \sqrt{2}\cos(\phi)}. \tag{23}$$

For a treatment of the CVT kinematics, see [5]. By the construction of Σ_i-space for the CVT and the definition of β_i, and using (23),

$$\beta = \tan^{-1}(\omega_{2i}/\omega_{1i}) = \tan^{-1}(M(\phi_i)). \tag{24}$$

Thus, from (23), the following forward kinematic relation may be derived:

$$\mathbf{t}_i = \begin{bmatrix} \dfrac{\sin(\phi) - \sqrt{2}\cos(\phi)}{(2 + 2\cos^2(\phi))^{1/2}} \\ \dfrac{\sin(\phi) + \sqrt{2}\cos(\phi)}{(2 + 2\cos^2(\phi))^{1/2}} \end{bmatrix}. \tag{25}$$

By differentiation of (25) and use of (23), we have

$$\dot{\phi}_i = u_i \kappa_i \frac{d\phi_i}{dM} \frac{dM}{d\beta_i} = u_i \kappa_i \frac{1 + \cos^2(\phi_i)}{\sqrt{2}} \tag{26}$$

where u_i is a signed scalar taking on positive values when the inner product between the direction defined by ϕ_i and $\mathbf{t}_i$ is positive. The speed u_i may be constructed by forming $\mathbf{V}_J = [\dot{q}_1 \cdots \dot{q}_N]^T$ as a measured vector of joint speeds and appealing to $u_i = \mathbf{t}_i^T D_i \mathbf{V}_J$.

E. The Forward Transformations

$\boxed{\Sigma_i \to C_J}$ It is generally necessary to compute the cobot's instantaneously available motion in C_J (or C_T) space based on the measured steering angles. This involves, as a first step, the forward kinematic computation of each $\mathbf{t}_i$, which was covered above [see (25)]. In this section, we show how to compute $\mathbf{T}_J$ from the set of coupling space tangents $\mathbf{t}_i$, $(i = 1, \ldots p)$. A key to this computation is the fact that $D_i \mathbf{T}_J$ is a 2 × 1 vector parallel to $\mathbf{t}_i$. If we introduce the following 90° rotation matrix:

$$W = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{27}$$

then we can write

$$[W\mathbf{t}_i]^T D_i \mathbf{T}_J = 0 \tag{28}$$

and, by concatenation,

$$\begin{bmatrix} [W\mathbf{t}_1]D_1 \\ \vdots \\ [W\mathbf{t}_{n-1}]D_{n-1} \end{bmatrix} \mathbf{T}_J = 0. \tag{29}$$

By adding a row of zeros to the matrix on the left, (29) takes on the appearance of an eigenvalue/eigenvector problem in which the eigenvalue is known to be zero, and $\mathbf{T}_J$ is the eigenvector. The solution to this problem is well known. If we define Λ as

$$\Lambda = \left[\Lambda_1 \ \cdots \ (-1)^{k+1}\Lambda_k \ \cdots \ (-1)^n \Lambda_{n-1}\right]^T \tag{30}$$

where Λ_k is the determinant of the matrix formed by removing the kth column of the matrix in (29), then

$$\mathbf{T}_J = \frac{\Lambda}{|\Lambda|}. \tag{31}$$

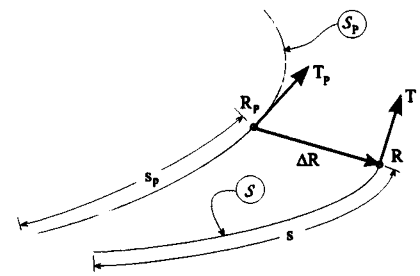

Fig. 5. The configuration error in task space (C_T).

$\boxed{C_J \to C_T}$ The forward kinematics are used to compute the cobot configuration $\mathbf{R}$ in terms of the joint space coordinates

$$\mathbf{R} = L^{-1}(\mathbf{q}). \tag{32}$$

The forward tangent transformation may be expressed using the inverse Jacobian

$$\mathbf{T} = \frac{J^{-1}\mathbf{T}_j}{|J^{-1}\mathbf{T}_J|}. \tag{33}$$

III. Path-Following Controller Design

We turn now to the design of a steering controller that causes a cobot's end-effector, as a human operator pushes, to follow a predetermined path through its workspace. Viewed in C_T space, the predetermined path is a curve or 1-D manifold, which we label S_P as shown in Fig. 5. The controller to be designed is called a *path-following controller*—it controls the CVT transmission ratios so that curve S [traced by the end-effector configuration point $\mathbf{R}(s)$] follows curve S_P. Moreover, feedback control is employed so that $\mathbf{R}(s)$ asymptotically approaches S_P from any initial position in C_T space as s increases under the influence of the human operator.

To ensure that the path-following controller design is generic to all cobot architectures, it is developed in C_T space. To guide curve S toward S_P, the controller produces the curvature vector $\kappa\mathbf{N}$. To adapt $\kappa\mathbf{N}$ for a particular cobot, the curvature transformations introduced in the previous section [see (9) and (15)] are applied, generating the steering speeds $\dot{\phi}_i$ $(i = 1, \ldots, p)$. The measured steering angles ϕ_i $(i = 1, \ldots, p)$ are processed in turn through the inverse tangent transformations, (33) and (30), to produce $\mathbf{T}$, which is required by the controller.

In the path-following design problem (unlike the trajectory tracking problem), a reference configuration $\mathbf{R}_p(s)$ is not available for comparison to $\mathbf{R}(s)$. Instead, the entire curve S_P is given. Nevertheless, a reference point $\mathbf{R}_p$ may be chosen from S_P so long as that choice is made (and maintained) by the controller. Let s_p be a pathlength parameterization of curve S_P. Then s_p may be used by the controller to select a reference point $\mathbf{R}_p(s_p)$ and correspondingly, a reference tangent $\mathbf{T}_p(s_p)$ and reference curvature vector $\kappa_p\mathbf{N}_p(s_p)$. The controller is held responsible for maintaining s_p as a function of the pathlength s.

One algorithm that has proven useful in practice is to choose s_p such that $\mathbf{R}_p(s_p)$ is always the closest point on S_P to $\mathbf{R}_p(s)$. With this algorithm in place, however, a stability analysis is not tractable. The present design controls s_p dynamically, achieves good performance, and most importantly, is accompanied by a stability guarantee.

The present design is based on a system representation that includes both the control of the cobot configuration $\mathbf{R}(s)$ and the control of s_p in a single problem statement. The state variables are taken to be the configuration error $\Delta\mathbf{R}$ and its derivative $\Delta\mathbf{R}'$. Fig. 5 shows the actual configuration $\mathbf{R}$ lying at path-length s on curve S in C_T space. Also shown is the reference configuration $\mathbf{R}_p$, located on the preplanned path S_P by path-length s_p. The configuration error $\Delta\mathbf{R}$ is the vector difference of $\mathbf{R}(s)$ and $\mathbf{R}_p(s_p)$. The path tangents $\mathbf{T}$ and $\mathbf{T_P}$ at $\mathbf{R}(s)$ and $\mathbf{R}_p(s_p)$ are also shown.

After defining a state $\mathbf{x} \triangleq [\Delta\mathbf{R}\ \Delta\mathbf{R}']^T$, where $(\cdot)'$ indicates differentiation with respect to s, the system equations may be written

$$\dot{\mathbf{x}} \triangleq \begin{bmatrix} \Delta\mathbf{R}' \\ \Delta\mathbf{R}'' \end{bmatrix} = \begin{bmatrix} \mathbf{T} - \mathbf{T}_p s_p' \\ \kappa\mathbf{N} - s_p''\mathbf{T}_p - (s_p')^2\kappa_p\mathbf{N}_p \end{bmatrix} \tag{34}$$

where the identities $\mathbf{R}' = \mathbf{T}$, $\mathbf{T}' = \kappa\mathbf{N}$ have been used. Note that these system equations are nonlinear in the states $\Delta\mathbf{R}$ and $\Delta\mathbf{R}'$. The system output $\mathbf{y}$ is defined as

$$\mathbf{y} = \Delta\mathbf{R} \tag{35}$$

and the objective of the controller is to drive $\mathbf{y}$ to zero.

The appearance of s_p' and s_p'' in (34) reveals the manner in which s_p is controlled: through "dynamic extension" [3]. The quantity s_p'' is defined as a new system input, while s_p' and s_p are computed by integration. In simulation, the state $\mathbf{x}$ may be augmented with the scalar variables s_p and s_p' to perform the integration inside the system model. In practice, the controller itself carries out the integration of s_p''.

A. Feedback Linearization

Our path following controller is based on an input–output linearization of the system (34). Its development follows that in [6]. Computed torque control, familiar in robotics, is based on feedback linearization. The central idea of the feedback linearization approach is to algebraically transform a nonlinear system into a linear system so that linear control techniques may be applied. An outer loop linear controller then completes the controller design. In the absence of internal dynamics in the input–output system, asymptotic stability of the closed-loop system follows from a linear analysis.

Note that the control inputs $\kappa\mathbf{N}$ and s_p'' appear after taking two derivatives of the output $\mathbf{y}$, thus the relative degree is 2 [3], [2]. Since the system order is also 2, there are no internal dynamics associated with this input–output system and an input–output linearization leads to an input-state linearization.

The inputs we have identified, $\kappa\mathbf{N}$ and s_p'', may in fact be taken as two projections of a single n-dimensional control input $\mathbf{U}$. Although $\kappa\mathbf{N}$ is an n-dimensional vector, its direction is not arbitrary: it must lie in a plane perpendicular to $\mathbf{T}$. Thus $\kappa\mathbf{N}$ has only $n-1$ free parameters, leaving one linear combination of the elements of $\mathbf{U}$ available for defining the scalar s_p''. Specifically, s_p'' is defined as the magnitude of the projection of $\mathbf{U}$ onto $\mathbf{T}$

$$s_p'' = \mathbf{T}^T\mathbf{U}. \tag{36}$$

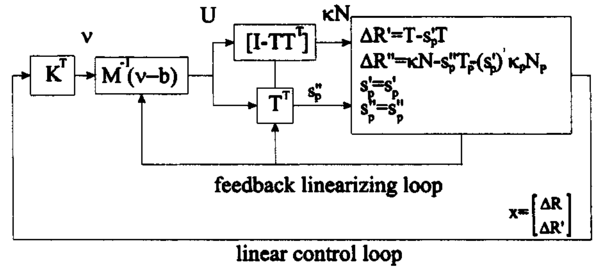

Fig. 6. Block diagram showing the inner linearizing loop and the outer linear loop.

The projection of $\mathbf{U}$ onto a plane perpendicular to $\mathbf{T}$ produces the term $\kappa\mathbf{N}$

$$\kappa\mathbf{N} = [\mathbf{I} - \mathbf{T}\mathbf{T}^T]\mathbf{U} \tag{37}$$

where $[\mathbf{I} - \mathbf{T}\mathbf{T}^T]$ is a projection matrix of rank $n-1$. Substituting (36) and (37) into the output equation yields

$$\mathbf{y}'' = \mathbf{M}\mathbf{U} + \mathbf{b} \tag{38}$$

where $\mathbf{M} \triangleq [\mathbf{I} - \mathbf{T}\mathbf{T}^T - \mathbf{T}_p\mathbf{T}^T]$ and $\mathbf{b} \triangleq -(s_p')^2\kappa_p\mathbf{N}_p$. A feedback linearizing controller may now be designed by defining $\mathbf{U}$ in terms of a new input $\boldsymbol{\nu}$ as

$$\mathbf{U} \triangleq \mathbf{M}^{-1}[\boldsymbol{\nu} - \mathbf{b}] \tag{39}$$

where $\mathbf{M}$ and $\mathbf{b}$ are continually updated as functions of the state $\mathbf{T}$ and s_p' and s_p. Then the input–output system is transformed into an equivalent linear system

$$\mathbf{y}'' = \boldsymbol{\nu}. \tag{40}$$

Fig. 6 shows the inner linearization loop which renders the cascade system comprising linearizing controller, projections, cobot model, and output equation as the simple decoupled second-order system $\mathbf{y}'' = \boldsymbol{\nu}$. Finally, the outer loop linear controller is designed using linear techniques, such as pole placement. The linear controller shown in Fig. 6 uses a full state feedback gain matrix $\mathbf{K}$.

The path-following controller based on feedback linearization enjoys asymptotic stability. Though valid in a large region of the state space, it is not global: the controller is not well defined when $\mathbf{T}$ is perpendicular to $\mathbf{T}_p$. However, given reasonable starting configurations and allowing for reversals in rolling direction, this situation is not troublesome for cobots. Also note that the linearizing controller relies on a system model. Uncertainty in the model will cause error in computation of the control input $\mathbf{U}$. Future papers will introduce alternative nonlinear controllers designed for robustness to modeling errors.

Fig. 7 shows a block diagram that includes the outer loop linear controller, the inner linearizing loop, and the tangent and curvature transformations. The cobot is shown here as a composition of CVT models and the error vector is formed by differencing the monitored cobot position and tangent with the position and tangent chosen by the controller from the preplanned path.

In simulation, a model of the cobot is used, where the current heading of each CVT ϕ_i is maintained by integration of the

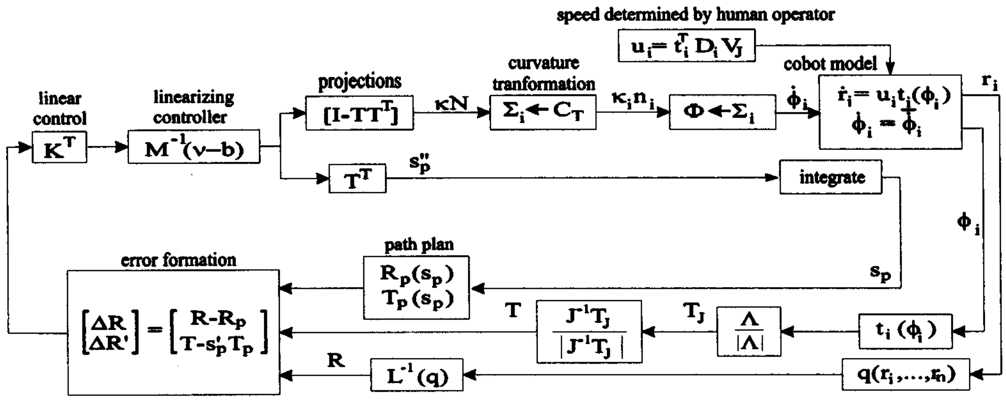

Fig. 7. Block diagram showing the controller in taskspace and the cobot model in steering space. The curvature and tangent transformations link the controller and cobot model.

input, and the coordinates in coupling space $\mathbf{r}_i$ are maintained by integration of the current speed (determined by the human operator) in the direction of the current heading. In practice, the cobot itself replaces these differential equations and $\mathbf{r}_i$ and ϕ_i are measured quantities. The speed of the cobot is shown as an input to the model and set by the human operator. The quantity u_i is the speed in the direction of allowed motion and is the one degree of freedom always in the control of the human operator. For the purposes of informing the cobot controller about the current speed, u_i may be formed by computing $u_i = \mathbf{t}_i^T D_i \mathbf{V}_J$, where $\mathbf{V}_J$ is a vector of measured joint speeds.

IV. Examples

In this section, we will develop path following controllers for the same jib cobot and three-wheeled cobot called "Scooter" that were briefly used in Section II to introduce the kinematic spaces.

A. The Jib Cobot: Configuration of a Point in the Plane

In this example, the CVT steering angle is controlled such that C approaches and follows a predefined path from any starting position within the workspace. Referring to Fig. 1, the C_J-space position vector $\mathbf{q} = [r\ \theta]^T$ is expressed in terms of the elements of the C_T-space position vector $\mathbf{R} = [x\ y]^T$ in

$$\mathbf{q}(\mathbf{R}) = \begin{bmatrix} \sqrt{x^2+y^2} \\ \tan^{-1}(y/x) \end{bmatrix}. \tag{41}$$

Define θ_1 and θ_2 as the angular displacements of the CVT drive rollers. Let the first drive roller be coupled directly to θ and the second drive roller be coupled to r through a cable drive with transmission ratio ρ. The Σ_1-space position vector $\mathbf{r}_1 = [\theta_1\ \theta_2]^T$ is expressed in terms of the elements of $\mathbf{q}$ in

$$\mathbf{r}_1(\mathbf{q}) = \begin{bmatrix} \frac{1}{\rho} r \\ \theta \end{bmatrix}. \tag{42}$$

Expressed in terms of the elements of $\mathbf{R}$, the Jacobian relating speeds in C_J space to speeds in C_T space reads

$$J(\mathbf{R}) = \frac{\partial \mathbf{q}}{\partial \mathbf{R}} = \frac{1}{r^2}\begin{bmatrix} rx & ry \\ -y & x \end{bmatrix}. \tag{43}$$

The transmission matrix relating directions in Σ_1 space to directions in C_J space is

$$D_1 = \frac{\partial \mathbf{r}_1}{\partial \mathbf{q}} = \begin{bmatrix} 1/\rho & 0 \\ 0 & 1 \end{bmatrix}. \tag{44}$$

The column matrix appearing in the curvature transformation [see (9)] can be expressed using two Hessian matrices as follows:

$$\mathbf{T}^T \frac{\partial \mathbf{J}}{\partial \mathbf{R}} \mathbf{T} = \begin{bmatrix} \mathbf{T}^T H_{11} \mathbf{T} \\ \mathbf{T}^T H_{12} \mathbf{T} \end{bmatrix} \tag{45}$$

where

$$H_{11} = \frac{1}{r^3}\begin{bmatrix} y^2 & -xy \\ -xy & x^2 \end{bmatrix}$$

and

$$H_{12} = \frac{1}{r^4}\begin{bmatrix} 2xy & (y^2-x^2) \\ (y^2-x^2) & -2xy \end{bmatrix}. \tag{46}$$

Since there is only one coupling space for the jib cobot, these formulas are all that is needed to implement the above path-following controller.

Fig. 8 shows simulation results in C_T space for the jib cobot under path-following control starting at position (1.5, 0) and headed in the y direction. The preplanned path is a line oriented at 45 degrees. The cart approaches, then stays on the path. The locations of the reference cobot on the planned path chosen by the controller as simulation proceeds are indicated with circles while the corresponding positions of the cobot are asterisks. The speed along the path (determined by an exogenous agent representing the human operator) starts at 0.5 units per second, ramps linearly up to 2.5 units per second at $t = 4$ seconds and returns to 0.5 at $t = 8$ seconds. Note that this speed input influences the spacing of the points on the path (both on the path taken and the reference points chosen from the pre-planned path,) but does not influence the shape of the asymptotic path.

Fig. 9 shows simulation results in C_T space for the jib cobot under path following control starting at the same initial conditions but following a circular pre planned path.

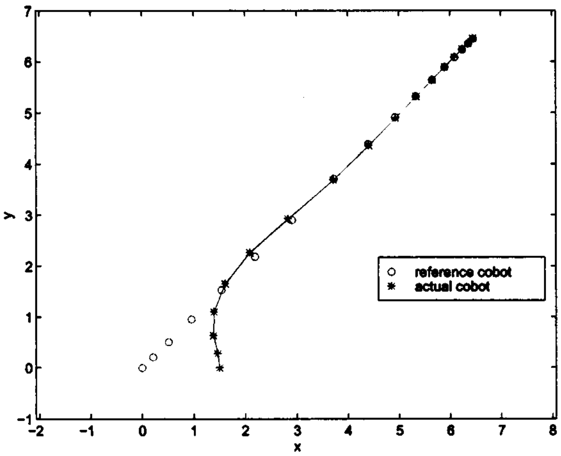

Fig. 8. The Jib cobot following a linear path oriented at 45 degrees in C_T space.

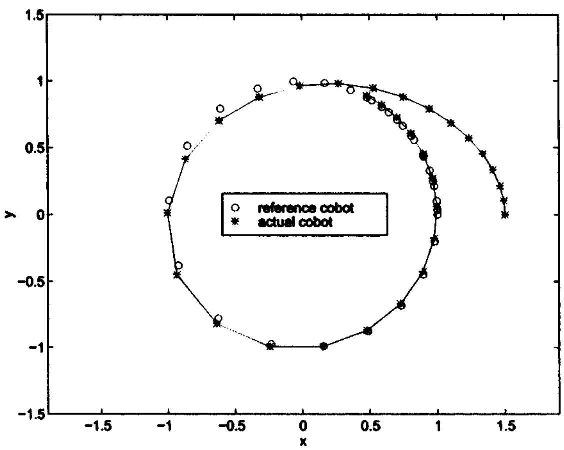

Fig. 9. The Jib cobot following a unit circular path centered at the origin in C_T space.

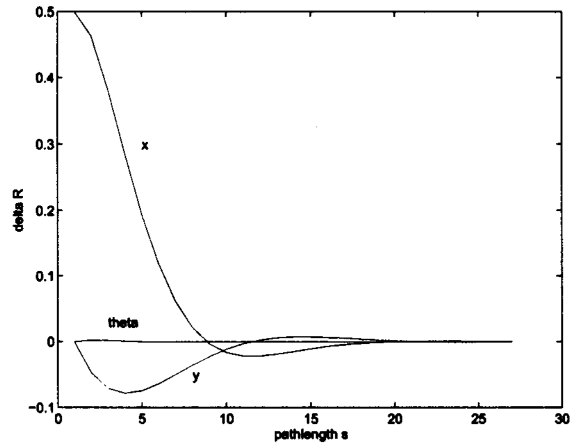

Fig. 10. The configuration error $\Delta\mathbf{R}$.

B. Scooter: Configuration of a Body in the Plane

Referring to Fig. 3, the C_T-space position vector $\mathbf{R}$ is $[x\ \ y\ \ \theta]^T$. Each wheel W_i corresponds to a "joint," with the Cartesian coordinates x_i and y_i of P_i as joint variables. The C_J-space position vector $\mathbf{q}$ is $[x_1\ \ y_1\ \ x_2\ \ y_2\ \ x_3\ \ y_3]^T$. This cobot has three coupling spaces Σ_i $(i = 1, 2, 3)$, with position vectors $\mathbf{r}_i = [x_i\ y_i]^T$ $(i = 1, 2, 3)$.

The taskspace C_T has dimension $n = 3$, yet this cobot features three wheels (linear CVTs), so there is a redundancy. Two wheels would be sufficient to constrain the motion of $\mathbf{R}$ to a 1-D manifold in C_T space. A third wheel is used, however, to avoid a singularity that occurs when the axes of two wheels are parallel. The singularity may best be understood using the center of rotation (COR) of A to characterize the instantaneous rate-of-change of configuration. The COR is located at the intersection of the axes of the three wheels. The third axis defines the single point of intersection when two axes are parallel. All three wheels must agree on a single COR at all times and this is assured in practice using a low-level controller.

Referring again to Fig. 3, let l_{i1} and l_{i2} be the coordinates of P_i in A, where the A-fixed coordinate frame is aligned with the C_T coordinate frame when $\theta = 0$. Then the joint coordinates (elements of $\mathbf{q}$) are given by

$$\begin{aligned} x_i &= x + l_{i1}c_\theta - l_{i2}s_\theta \\ y_i &= y + l_{i1}s_\theta + l_{i2}c_\theta, \qquad i = 1, 2, 3 \end{aligned} \tag{47}$$

where s_θ denotes $\sin(\theta)$ and c_θ denotes $\cos(\theta)$. The Jacobian matrix relating $\mathbf{T}_J$ to $\mathbf{T}$ may be written

$$J = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} \tag{48}$$

where

$$J_i = \begin{bmatrix} 1 & 0 & -l_{i1}s_\theta - l_{i2}c_\theta \\ 0 & 1 & l_{i1}c_\theta - l_{i2}s_\theta \end{bmatrix}, \qquad i = 1, 2, 3. \tag{49}$$

The transmission matrix D_i is a (2×6) matrix with the (2×2) identity in the ith block and zeros elsewhere.

Finally, the Hessian matrices may be found by differentiating the Jacobian matrices

$$\begin{aligned} H_{i1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -l_{i1}c_\theta + l_{i2}s_\theta \end{bmatrix} \\ H_{i2} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -l_{i1}s_\theta - l_{i2}c_\theta \end{bmatrix}. \end{aligned} \tag{50}$$

The path-following controller has been tested in simulation on Scooter. Fig. 10 shows the configuration error variables $\Delta\mathbf{R}$ approaching the origin as pathlength s grows. The initial condition was $\mathbf{R} = [1.5\ 0\ \pi/2]^T$ and $\mathbf{R_p} = [1\ 0\ \pi/2]^T$. The initial steering angles were $\pi/2$ for all three wheels.

The preplanned path used in simulation was a helix centered at the origin with unity radius and pitch 2π radians per unit length along the z-axis. Fig. 11 shows the path traced by $\mathbf{R}$ approaching the pre-planned path in C_T-space. The reference

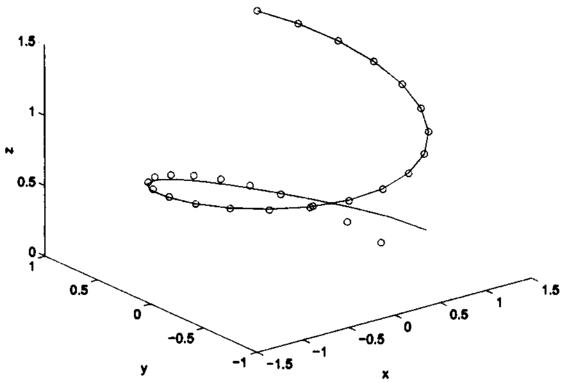

Fig. 11. Path following in C_T space.

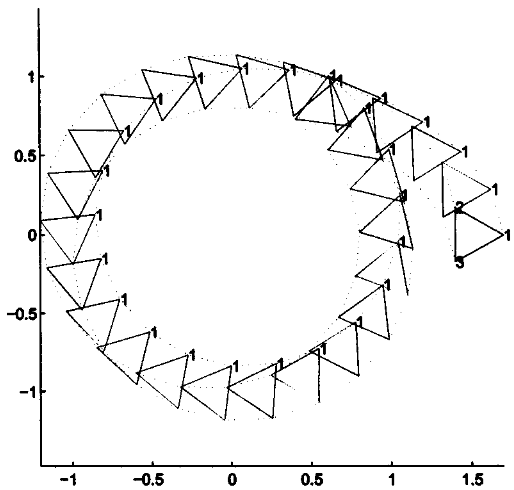

Fig. 12. Path following in Σ_i space $(i = 1, 2, 3)$.

points on the pre-planned path are shown as circles. A constant speed was used for the human input.

Fig. 12 shows the performance of the path-following controller in the joint space (or equivalently, the coupling spaces) of Scooter. The approach to the circular helix can be recognized as well as the rotation about its center.

V. Summary

An asymptotically stable path-following controller has been developed for cobots. From any starting configuration, the cobot will choose a heading that converges to and then follows the path as the human operator chooses the speed of motion. The transformations between task space, joint space, the coupling spaces, and steering space are essential for the development and implementation of general and extensible controllers. Path-following controllers have been developed for other cobots using the present framework, including the unicycle and the rail cobot.

The robustness of the present path-following controllers to modeling errors has not been analyzed, although some simulation studies indicate that certain modeling errors can be tolerated without significant loss in performance. Certainly the sensitivity to sideslip in the CVT or wheel rolling contacts merits further study. The real test is implementation of these path following controllers on various cobots in the laboratory, an activity which is currently underway.

Acknowledgment

The authors gratefully acknowledge the many insights provided by W. Wannasuphoprasit and P. Akella. They also thank the anonymous reviewers for their thoughtful comments and suggestions.

References

[1] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. Moore, R. B. Gillespie, J. Santos-Munne, A. Lorenz, and P. Akella, "Cobot architecture," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 377–390, Aug. 2001.

[2] A. Isidori, *Nonlinear Control Systems: An Introduction*. Berlin, Germany: Springer-Verlag, 1985, Lecture Notes in Control and Information Sciences, vol. 72.

[3] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[4] W. Wannasuphoprasit, R. B. Gillespie, J. E. Colgate, and M. A. Peshkin, "Cobot control," in *IEEE Int. Conf. Robotics and Automation*, 1997, pp. 3571–3576.

[5] B. Gillespie, C. Moore, M. Peshkin, and J. E. Colgate, "Kinematic creep in continuously variable transmissions: Traction drive mechanics for cobots," *J. Mech. Syst. Design*, 2001, to be published.

[6] N. Sarkar, X. Yun, and V. Kumar, "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots," *Int. J. Robotics Res.*, vol. 13, no. 1, pp. 55–69, 1994.

R. Brent Gillespie (M'96) received the B.S. degree in mechanical engineering from the University of California, Davis, in 1986, the Master's degree in music (piano performance) from the San Francisco Conservatory of Music in 1989, and the M.S. and Ph.D. degrees in mechanical engineering from Stanford University, Stanford, CA, in 1992 and 1996, respectively.

Prior to beginning his graduate studies, he spent four years with Hewlett Packard, San Jose, CA. From 1996 through 1999, he was a Postdoctoral Researcher with Northwestern University, Evanston, IL. Since 1999, he has been an Assistant Professor of mechanical engineering at the University of Michigan, Ann Arbor. His research interests include haptic interface and cobots.

Dr. Gillespie has received a National Science Foundation CAREER Award.

J. Edward Colgate (M'88) received the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, in 1988.

He subsequently joined Northwestern University, Evanston, IL, where he is currently an Associate Professor in the Department of Mechanical Engineering. His principal research interests are cobots—collaborative robots—and haptic interface. In addition to his academic pursuits, he is a founder of Cobotics, Inc., a leading developer of human interface technologies for the industrial marketplace.

Dr. Colgate has served as an Associate Editor for the IEEE Transactions on Robotics and Automation. He has served as U.S. Editor of *Robotics and Computer Integrated Manufacturing*, and as an Associate Editor of the *Journal of Dynamic Systems, Measurement and Control*.

Michael A. Peshkin (S'86–M'86) received the Ph.D. degree in physics from Carnegie-Mellon University, Pittsburgh, PA, in 1987.

He is currently an Associate Professor in the Department of Mechanical Engineering at Northwestern University, Evanston, IL. With colleague J. E. Colgate, he is an inventor of cobots and a founder of Cobotics, Inc. He is also a founder of Z-KAT, Inc., in the image-guided surgery field. He holds patents on cobots, image-guided surgery, and electromagnetic sensors. Among current projects, he is working with the Robotics Industries Association, creating a national safety standard for intelligent assist devices.

Dr. Peshkin has served as Associate Editor of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and was for several years the Conference Publications Editor of IEEE Control Systems Society (ACC and CDC conferences).