A General Framework for Cobot Control

R. Brent Gillespie J. Edward Colgate
Michael Peshkin

Northwestern University
2145 Sheridan Rd, B-230
Evanston, IL 60208
b-gillespie@nwu.edu colgate@nwu.edu peshkin@nwu.edu

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 nonlinear feedback controller is developed to steer the cobot so as to asymptotically approach and follow a pre-planned path. Generality across all cobot architectures is assured by basing the controller in end-effector configuration space and developing transformations between each of four spaces: end effector space, joint space, a set of coupling spaces and steering space.

1. Introduction

Unlike robots, cobots are intended for direct collaborative work with a human operator [1],[2]. The human and cobot both grasp a workpiece and cooperate to complete a manipulation task. To team up with a cobot in this manner presents minimal physical danger to the human since a cobot, by design, cannot move on its own. It is up to the human operator to produce all motions of the cobot and workpiece. The cobot contributes to the manipulation process by guiding the motion. For example, a cobot can restrict the set of reachable configurations or create virtual guideways along which the workpiece must follow. In effect, the cobot creates virtual fixtures to constrain workpiece configuration or motion. As demonstrated in the fields of haptic display and teleoperation, virtual fixtures are useful aids to a human operator performing manual tasks [3],[4].

Furthermore, a cobot can be designed to carry the weight of a workpiece, as do various counterbalance devices currently in use on automobile assembly lines. Indeed, the first applications envisioned for cobots are certain automobile assembly operations that have not been automated because they require the manual dexterity and adaptability of a human operator. These operations can be made safer, faster, and easier with virtual fixtures. For example, while an operator is hanging a car door, a cobot can support its weight and prevent collisions between the door and car body except at the hinge.

Like a physical fixture or barrier placed in the workspace, a viable virtual fixture must be able to produce reaction forces when contact between workpiece and fixture is detected. In cobots, continuously variable transmissions (CVTs) support these reaction forces. In contrast to the direct actuation (realized with motors) used in robots and haptic interfaces, the CVT is a passive mechanism: it can be used to resist applied forces but not to produce motive forces or output forces. Alternatively, a CVT may be considered a tunable nonholonomic constraint, setting a ratio between two joint speeds.

There are in fact two distinct types of CVT that have been used in cobots. The first is quite simple: a single steered wheel rolling on a planar surface. A steered wheel constrains a pair of translational speeds (i.e., x and y where x and y are Cartesian coordinates of the planar surface). The ratio of these speeds, and equivalently, the allowed direction of motion, are defined by the heading of the wheel. The second type of CVT relates two angular speeds [5],[6]. This CVT is composed of a sphere caged between two drive rollers and two steering rollers. The CVT constrains the drive roller speeds (ω_1, ω_2). The ratio of these speeds (equivalently, the allowed direction of motion in ω_1 ω_2 space) is defined by the heading of the steering rollers.

A General Framework for Cobot Control
R. Brent Gillespie, J. Edward Colgate, Michael Peshkin
1999 International Conference on Robotics and Automation. Detroit MI.

1.1 Apparent Degrees of Freedom

Each CVT furnishes one nonholonomic constraint and may be used to eliminate one instantaneous degree of freedom from the workpiece. If a minimal description of a non-redundant cobot's configuration uses n generalized coordinates, the number of CVTs extant in that cobot is n-1. Thus a non-redundant cobot inherently constrains the workpiece down to a single degree of freedom[1]. In other words, the available motions at each instant are restricted to a single direction.

By active tuning, the allowed direction of motion may be varied. By tuning in response to user applied forces, sensed configuration, and sensed motion, the cobot can be made to appear subject to arbitrary constraints. The number of *apparent* degrees of freedom (DOF) of a cobot, then, is a function of the control algorithm in effect at a given time, and is thus variable. At maximum, the cobot can allow motion in all directions, with effectively no relationships imposed among the joint speeds. This is called caster mode, since the cobot behaves as if each steering wheel (CVT) was a ball caster. Caster mode requires sensing of user-applied forces. Essentially, the controller steers so as to allow motion in whatever direction the cobot is being pushed.

By successive change of control algorithm, the number of degrees of freedom may be reduced incrementally down to one. The case of one degree of freedom is called path following mode. In path following mode, the operator may influence only the speed along a pre-defined path. Rather than force sensing, the path-following controller uses full state feedback and input-to-state linearization. Path following controllers will be developed and analyzed in detail in Section 3 of this paper. Controllers which realize caster mode and intermediate numbers of degrees of freedom will be treated in future papers.

In order to treat cobot controller design in a general framework, generic to all cobot architectures, we introduce in Section 2 four abstract spaces. Controller design takes place in what we call end-effector configuration (C_{EE}) space. Actual steering of the CVTs, however, takes place in steering (Φ) space. Transformations are introduced between C_{EE} space, joint (C_J) space, a set of coupling ($_i$) spaces, and Φ space. These transformations become part of the actual implementation of a cobot controller, as demonstrated by way of example in Section 4.

[1] The exception to this is that singularities may occur. This important topic will be addressed in a future paper.

2. Cobot Kinematics

2.1 Four Spaces

Associated with cobots are four classes of kinematic space: end effector configuration space, joint configuration space, a set of coupling spaces, and steering space. These are depicted in Fig. 1.

End effector configuration space (C_{EE} space, or "end effector space" for short) is of importance in planning cobot motions, because it does not rely on the details of the cobot structure. Each point in end effector space corresponds to a pose of the cobot end effector. Key variables associated with end effector space are end effector configuration ($\mathbf{R}$), path tangent ($\mathbf{T}$), path normal ($\mathbf{N}$), path curvature (κ), path length (s), and speed (u s).

Joint configuration space (C_J space, or "joint space" for short) has the same meaning as in robotics, but it does not take on the same level of importance because there are no actuators at the joints. The joints, however, are coupled by CVTs, and there may well be sensors at the joints. The key variable in joint space is cobot configuration ($\mathbf{q}$).

A two dimensional **coupling space** ($_i$ space) is associated with each pair of joints that are coupled by a CVT. There is, therefore, a set comprising one coupling space for each CVT. Coupling spaces prove to be quite useful in describing the steering behavior of a cobot, because they offer a geometric interpretation analogous to that associated with endpoint space. Moreover, coupling spaces are independent of CVT details (e.g., the coupling space behavior is the same whether a steered wheel, a cubic CVT, a tetrahedral CVT, or some other device is used). Key variables associated with coupling space $_i$ are configuration ($\mathbf{r}_i$), path tangent ($\mathbf{t}_i$), path normal ($\mathbf{n}_i$), path curvature (κ_i), and path length (s_i).

Steering space (Φ space) comprises the set of all CVT steering angles, ϕ_i, i $1,2,$ n $1.$ In

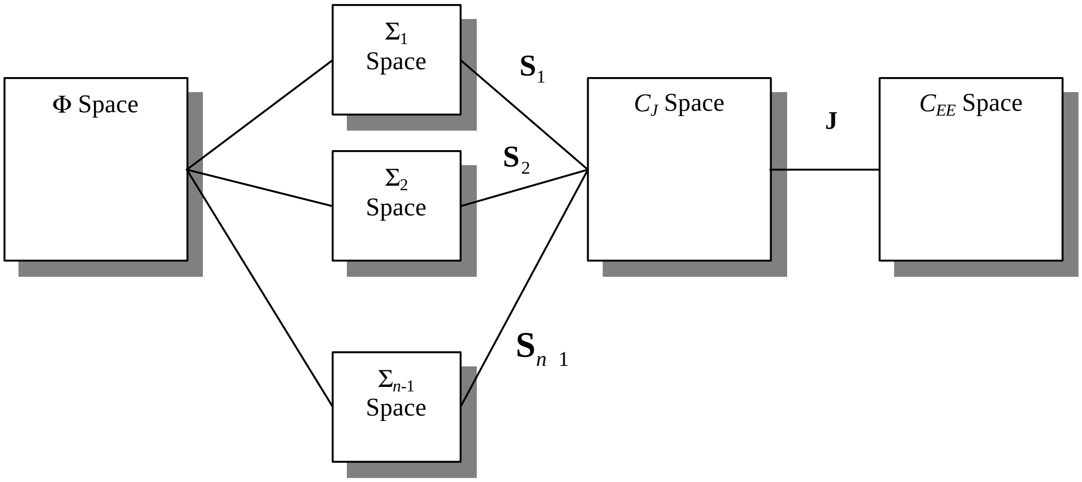

Figure 1. Four classes of kinematic space

addition to the steering angles, the steering angular speeds, $\dot{\phi}_i$, are important, and are generally taken to be the controller input variables.

2.2 Kinematics Overview

The distinguishing feature of a cobot is the network of CVTs that couple its joints. Each CVT establishes, under computer control, a transmission ratio relating the speeds of two joints. The types of CVTs used in cobots have the special character that they may establish *any* such ratio, including negative ratios. A non-redundant cobot is one that couples the n joints through a network of n-1 CVTs.

The following points may be made about a non-redundant cobot:

- For a given non-singular setting of the n-1 transmission ratios, the direction of $d\mathbf{q}$ is fixed. In other words, the cobot has one instantaneous degree-of-freedom.
- By adjusting the transmission ratios, *any* direction of $d\mathbf{q}$ may be obtained. This follows from the CVTs' infinite adjustability. When in nonsingular mechanism configurations, the same conclusion may be drawn for $d\mathbf{R}$.
- Controlling the transmission ratios has no direct effect on the cobot's speed (which, measured in joint space, is $|d\mathbf{q}/dt|$). Speed is under the control of the human operator.

Because of this last point, it is often useful to think of cobots in terms of spatial derivatives rather than temporal derivatives. In this paper, spatial derivatives will be taken with respect to the cobot's C_{EE} space path length variable, s:

$$ds = \left(d\mathbf{R}^T d\mathbf{R}\right)^{1/2} \tag{2.1}$$

Key derivatives of interest are:

$$\mathbf{T} = \mathbf{R}' = \frac{d\mathbf{R}}{ds} \tag{2.2}$$

$$\kappa\mathbf{N} = \mathbf{T}' = \frac{d\mathbf{T}}{ds} \tag{2.3}$$

As the notation suggests, it is useful to think of the cobot's trajectory through C_{EE} space as a curve C with a configuration vector **R,** a tangent (first derivative with respect to path length) **T** and curvature vector $\kappa\mathbf{N}$. The former is a unit vector (which follows from the definition of s) and the latter is denoted as the product of a scalar curvature, κ, and a unit vector, **N**, normal to **T**.

The behavior of a cobot is most appropriately understood in terms of **T** and $\kappa\mathbf{N}$. The instantaneously allowed direction of motion, viewed in C_{EE} space, is **T**. Thus, in a nonsingular configuration, a given setting of the n-1 transmission ratios is equivalent to a particular tangent direction.

Even more important from the standpoint of control is the role of the curvature vector. Analogous to the relationship between the tangent vector and transmission ratio, there is a relationship between the curvature vector and rate-of-change of the transmission ratio. The latter is the primary control input to a cobot. Thus, in developing C_{EE} space controllers, the curvature vector is naturally treated as the primary control input. Section 3 develops these ideas more fully and applies them to the design of a path following controller.

2.3 Transformations

In order to develop useful controllers, a number of important transformations between the four spaces must first be understood. In this section, we present these transformations, with reference to Fig. 1. We begin with inverse transformations.

2.3.1 Inverse Transformations

$\boxed{C_J \quad C_{EE}}$ The standard Jacobian plays a key role in transformations between end effector and joint space. In this paper, the Jacobian will be taken to relate small displacements in the end effector space ($\delta\mathbf{R}$) to small displacements in the joint space ($\delta\mathbf{q}$) according to:

$$\delta\mathbf{q} = \mathbf{J}(\mathbf{R})\delta\mathbf{R} \tag{2.4}$$

Where $\mathbf{J}(\mathbf{R})$ is a jacobian matrix. Note that, implicit in this definition of $\mathbf{J}$ is the notion that inverse kinematics is the "easy" direction. This stems from the fact that most cobots to date have exhibited parallel rather than serial architectures.

Two more important relations between end effector space and joint space are *the tangent and curvature transformations*. As the name implies, the tangent transformation relates a unit tangent in C_{EE} space to a unit tangent in C_J space. The relation follows easily from equations 2.2 and 2.4:

$$\mathbf{T}_J = \frac{d\mathbf{q}}{ds_J} = \frac{\mathbf{JT}}{|\mathbf{JT}|} \tag{2.5}$$

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

$$\text{if } \mathbf{A} = \frac{\mathbf{X}}{|\mathbf{X}|}$$

$$\text{then } \mathbf{A}' = \left[\mathbf{I} - \mathbf{A}\mathbf{A}^T\right]\frac{\mathbf{X}'}{|\mathbf{X}|} \tag{2.6}$$

where $\mathbf{A}$ and $\mathbf{X}$ are vectors and the prime denotes differentiation. Equations 2.5 and 2.6 lead to:

$$\kappa_J\mathbf{N}_J = \frac{d\mathbf{T}_J}{ds_J} = \frac{\left[\mathbf{I} - \mathbf{T}_J\mathbf{T}_J^T\right]}{|\mathbf{JT}_J|^2}\left(\mathbf{T}^T\frac{\partial\mathbf{J}}{\partial\mathbf{R}}\mathbf{T} + \mathbf{J}\kappa\mathbf{N}\right) \tag{2.7}$$

where $\mathbf{T}^T\frac{\partial\mathbf{J}}{\partial\mathbf{R}}\mathbf{T}$ is shorthand for a column vector whose i^{th} element is defined as:

$$\left\{\mathbf{T}^T\frac{\partial\mathbf{J}}{\partial\mathbf{R}}\mathbf{T}\right\}_{(i)} = \sum_{j=1}^{n}\left[\sum_{k=1}^{n}\frac{\partial\mathbf{J}_{(ij)}}{\partial\mathbf{R}_{(k)}}\mathbf{T}_{(k)}\right]\mathbf{T}_{(j)} \tag{2.8}$$

and where $\mathbf{R}_{(k)}$ denotes the k^{th} component of $\mathbf{R}$, $\mathbf{J}_{(ij)}$ denotes the ij^{th} component of $\mathbf{J}$, and so on. The matrix whose kj^{th} element is $\frac{\partial\mathbf{J}_{(ij)}}{\partial\mathbf{R}_{(k)}}$ may also be recognized as a Hessian. It is often convenient to express Eq. 2.7 in terms of Hessians, as will be demonstrated in Section 4.

$\boxed{\Sigma_i \quad C_J}$ Recall that the i^{th} coupling space (Σ_i space) comprises two coordinates corresponding to the two joints coupled by the i^{th} CVT. Because these same coordinates appear in the joint space, the basic kinematic relationship is extremely simple:

$$\delta\mathbf{r}_i = \mathbf{S}_i\delta\mathbf{q} \tag{2.9}$$

where $\mathbf{S}_i$ is a matrix which serves both to select the relevant two joints and incorporate any fixed transmission ratios relating the joint displacements to the CVT displacements. As an example, consider an RRPR cobot in which CVT 2 couples to joint 1 directly and to joint 3 via a transmission ratio of $1/r$. Then:

$$\mathbf{S}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/r & 0 \end{bmatrix}$$

It is again possible to define tangent and curvature transformations analogous to equations 2.5 and 2.7:

$$\mathbf{t}_i = \frac{\mathbf{S}_i\mathbf{T}_J}{|\mathbf{S}_i\mathbf{T}_J|} \tag{2.10}$$

$$\kappa_i\mathbf{n}_i = \frac{\left[\mathbf{I} - \mathbf{t}_i\mathbf{t}_i^T\right]}{|\mathbf{S}_i\mathbf{T}_J|^2}\left[\mathbf{S}_i\kappa_J\mathbf{N}_J\right] \tag{2.11}$$

$\boxed{\Sigma_i \quad C_{EE}}$ 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 cobots such as the Unicycle and Scooter [1], for which the concept of a joint is somewhat abstract, and arguably, counterproductive.

Fortunately, the transformations from end effector space to joint space are entirely analogous to those already derived. It is necessary to define a 2 n jacobian, $\mathbf{J}_i$, relating incremental end effector displacements to incremental $_i$ space displacements. When a well-defined joint space exists, this jacobian is simply:

$$\mathbf{J}_i \quad \mathbf{S}_i\mathbf{J} \tag{2.12}$$

For wheeled cobots, $\mathbf{J}_i$ can be directly computed.

The essential kinematic transformations are then:

$$\mathbf{t}_i \quad \frac{\mathbf{J}_i\mathbf{T}}{|\mathbf{J}_i\mathbf{T}|} \tag{2.13}$$

$$\kappa_i\mathbf{n}_i \quad \frac{[\mathbf{I} \quad \mathbf{t}_i\mathbf{t}_i^T]}{|\mathbf{J}_i\mathbf{T}|^2}\left(\mathbf{T}^T \frac{\mathbf{J}_i}{\mathbf{R}}\mathbf{T} \quad \mathbf{J}_i\kappa \quad \mathbf{N}\right) \tag{2.14}$$

$\boxed{\quad_i}$ To implement a desired curvature in C_{EE} space, it is necessary, ultimately, to compute the steering speeds, ϕ_i. The final set of transformations necessary to compute ϕ_i are CVT-specific. In other words, these transformations depend on the kinematics of the CVTs. We will illustrate two cases: the wheel and the tetrahedral CVT.

Wheel — The steering speed of a wheel is found almost trivially:

$$\phi_i \quad u_i\kappa_i \tag{2.15}$$

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 — In [1], it is shown that the transmission ratio of the tetrahedral CVT is:

$$M\ \phi \quad \frac{\omega_1}{\omega_2} \quad \frac{\sin(\phi) \quad \sqrt{2}\cos(\phi)}{\sin(\phi) \quad \sqrt{2}\cos(\phi)} \tag{2.16}$$

From Eq. 2.16, the following forward kinematic relation is easily derived:

$$\mathbf{t}_i \quad \begin{pmatrix} \dfrac{\sin(\phi_i) \quad \sqrt{2}\cos(\phi_i)}{\left(2 \quad 2\cos^2(\phi_i)\right)^{1/2}} \\ \dfrac{\sin(\phi_i) \quad \sqrt{2}\cos(\phi_i)}{\left(2 \quad 2\cos^2(\phi_i)\right)^{1/2}} \end{pmatrix} \tag{2.17}$$

differentiation and straightforward algebraic manipulation lead to:

$$\phi_i \quad [\mathbf{t}_i^T\mathbf{S}_i\mathbf{V}_J]\kappa_i \frac{1 \quad \cos^2(\phi_i)}{\sqrt{2}} \tag{2.18}$$

where $\mathbf{V}_J$ is a measured vector of joint speeds.

2.3.2 Forward Transformations

$\boxed{\quad_i}$ The key transformation from steering space to a particular coupling space has already been presented in Eq. 2.17. This relation specifies the available direction in $_i$ space given a steering angle ϕ_i.

$\boxed{_i \quad C_J}$ It is generally necessary to compute the cobot's instantaneously available motion in C_J (or C_{EE}) space based on the measured steering angles. This involves, as a first step, the forward kinematic computation of each $\mathbf{t}_i$

In this section, we show how to compute $\mathbf{T}$ from the set of coupling space tangents $_i$. A key to this $\mathbf{S}_i\mathbf{T}_J$ is a 2 1vector parallel to $\mathbf{t}_i$. If we introduce the following 90 rotation matrix:

$$\mathbf{W} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.19}$$

then we can write:

$$[\mathbf{W}\mathbf{t}_i]^T\mathbf{S}_i\mathbf{T}_J \quad 0 \tag{2.20}$$

and, by concatenation:

$$\begin{pmatrix} [\mathbf{W}\mathbf{t}_1]^T\mathbf{S}_1 \\ [\mathbf{W}\mathbf{t}_{n \ 1}]^T\mathbf{S}_{n \ 1} \end{pmatrix}\mathbf{T}_J \quad 0 \tag{2.21}$$

By adding a row of zeros to the matrix on the left, Eq. 2.21 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:

$$\Lambda = \left[\Delta_1 \quad \cdots \quad (-1)^{k+1}\Delta_k \quad \cdots \quad (-1)^{n}\Delta_{n+1}\right]^T \tag{2.22}$$

where Δ_k is the determinant of the matrix formed by removing the k^{th} column of the matrix in Eq. 2.21, then:

$$\mathbf{T}_J = \frac{\Lambda}{|\Lambda|} \tag{2.23}$$

$\boxed{C_J \rightarrow C_{EE}}$ The tangent vector in end effector space can be found via the inverse Jacobian:

$$\mathbf{T} = \frac{\mathbf{J}^{-1}\mathbf{T}_J}{\left|\mathbf{J}^{-1}\mathbf{T}_J\right|} \tag{2.24}$$

It is also important to compute the cobot configuration $\mathbf{R}$ in end effector space. $\mathbf{R}$ is, in general, a nonlinear function of the joint space coordinates:

$$\mathbf{R} = \mathbf{L}^{-1}(\mathbf{q}) \tag{2.25}$$

3. Path Following Control for Cobots

Our path following cobot controller is developed using the input-to-state feedback linearization approach [7], [8]. Similar to computed torque controllers often used in robotics, feedback linearizing controllers use model inversion to transform a nonlinear control problem into a linear control problem. An outer loop linear controller then completes the design procedure. The important advantage to these methods in our view is that they provide stability guarantees. Although this model-based approach may be sensitive to modeling errors, it has worked well in practice. Future papers will introduce alternative nonlinear controllers designed for robustness to modeling errors.

The path-following controller is developed below in four steps. First, n-dimensional kinematic error equations and error equation derivatives are developed in C_{EE} space. The error equations express the actual cobot configuration relative to a reference configuration on the desired path. Second, two terms appearing in the error equation are recognized as projections of an n-dimensional control input. Third, an input transformation renders the nonlinear cobot kinematic error equations in an equivalent linear form. Finally, a stabilizing outer loop controller is designed using linear control techniques.

3.1 Kinematic error equations in C_{EE} space

The appropriate space in which to address the design of cobot controllers is C_{EE} space since it is general, covering all cobot architectures. Once a controller is developed in C_{EE} space, it is implemented for a particular cobot by applying the transformations relevant to that cobot's architecture.

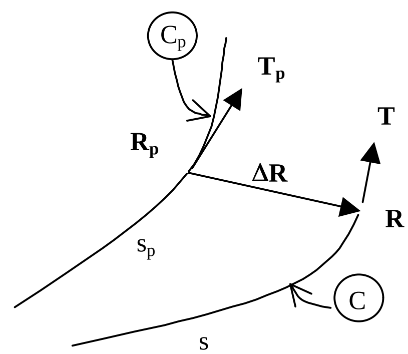

Figure 2. Configuration error in C_{EE} space

Figure 2 shows a cobot's current configuration $\mathbf{R}$ lying at pathlength s on a path C (indicating the configuration history) in C_{EE} space. Also shown in Figure 2 is a pre-planned desired path C_p through C_{EE} space with a special current reference configuration $\mathbf{R}_p(s_p)$ located on C_p by s_p. An effective path-following controller will cause path C to converge to and follow the pre-planned desired path C_p. This is done by driving the configuration error $\Delta\mathbf{R} = \mathbf{R} - \mathbf{R}_p$ to zero. Note that the controller is also responsible for maintaining the choice of $\mathbf{R}_p$ by controlling the evolution of s_p as discussed in further detail below.

The next step is to take derivatives of the configuration error expression until terms appear which are related to the actuated variables, the steering angular speeds. We take derivatives with respect to the pathlength *s* rather than time in order to realize a very important distinction between the influence available to the controller and the influence available to the human operator. As discussed in section 2.2 above, the controller influences the rate of change of heading while the human operator may influence only the speed along the heading.

In nonsingular configurations, the steering angles determine the current configuration heading (the allowed motion direction) and the steering angular speeds control the rate of change of the configuration heading. The heading is the derivative of **R** with respect to *s* and is the unit tangent vector **T** in Figure 2. Likewise, the rate of change of heading can be interpreted as the spatial derivative of **T**, or k**N** where k is the curvature and **N** is the normal to the curve *C* at **R**.

We take two derivatives of the configuration error to arrive at an expression which contains k**N** as follows**.**

$$\mathbf{R} \quad \mathbf{R} \quad \mathbf{R}_p \tag{3.1}$$

$$\mathbf{R} \quad \mathbf{R} \quad \mathbf{R}_p \quad \mathbf{T} \quad s_p \mathbf{T}_p \tag{3.2}$$

$$\begin{aligned} \mathbf{R} \quad & \mathbf{T} \quad s_p \mathbf{T}_p \quad s_p \mathbf{T}_p \\ & \kappa\mathbf{N} \quad s_p \mathbf{T}_p \quad \left(s_p\right)^2 \kappa_p \mathbf{N}_p \end{aligned} \tag{3.3}$$

3.2 Recognizing the control input

We now recognize the term k**N** as the portion of the control input available as steering angular speeds. It is not, however, the only control input. The remaining terms contain $s_p{}'$ and $s_p{}''$ which are not part of the pre-planned path and must therefore be maintained by the controller. Note that k**N** is a vector in *n*-dimensional C_{EE} space, yet its direction is not arbitrary. It must remain perpendicular to the tangent **T**. That is, there exists a scalar constraint equation on k**N.** The *n*-dimensional control input vector must be projected onto the plane perpendicular to **T** to produce the steering control k**N**. The projection parallel to **T** shall be given the responsibility of maintaining s_p**.**

Let us designate an *n*-dimensional vector **U** as the control input. Note that the projection of **U** in the **T** direction can have no effect on the cobot's heading or rate of change of heading. Also, since **U** shall eventually be expressed as steering, it has nothing to do with the speed s of the cobot along its path *C*. Instead, the magnitude of the projection of **U** in the **T** direction, $\mathbf{T}^T\mathbf{U}$, shall govern the motion of the reference cobot along C_p through

$$s_p \quad \mathbf{T}^T\mathbf{U}. \tag{3.4}$$

The state or error equations are augmented with another state s_p . As expected, the term k**N** is the projection of **U** onto a plane perpendicular to **T**:

$$\kappa\mathbf{N} \quad \left[\mathbf{I} \quad \mathbf{T}\mathbf{T}^T\right]\mathbf{U} \tag{3.5}$$

where $\mathbf{I} \quad \mathbf{T}\mathbf{T}^T$ is a projection matrix of rank *n*-1 which removes the component of **U** in the **T** direction.

3.3 Feedback Linearization

Substituting these two laws into equation (3.3), we arrive at

$$\Delta\mathbf{R}'' = \mathbf{M}\mathbf{U} - \mathbf{b} \tag{3.6}$$

where $\mathbf{M} \quad [\mathbf{I} \quad \mathbf{T}\mathbf{T}^T \quad \mathbf{T}_p\mathbf{T}^T]$ and $\mathbf{b} \quad (s_p \ )^2 \kappa_p \mathbf{N}_p$.

We are now ready to design a feedback linearizing controller by model inversion. Let ν be the control input derived from the outer-loop full-state feedback controller. We then produce **U** using

$$\mathbf{U} \quad \mathbf{M}^{\ 1}(\nu \quad \mathbf{b}). \tag{3.7}$$

3.4 The outer loop linear controller

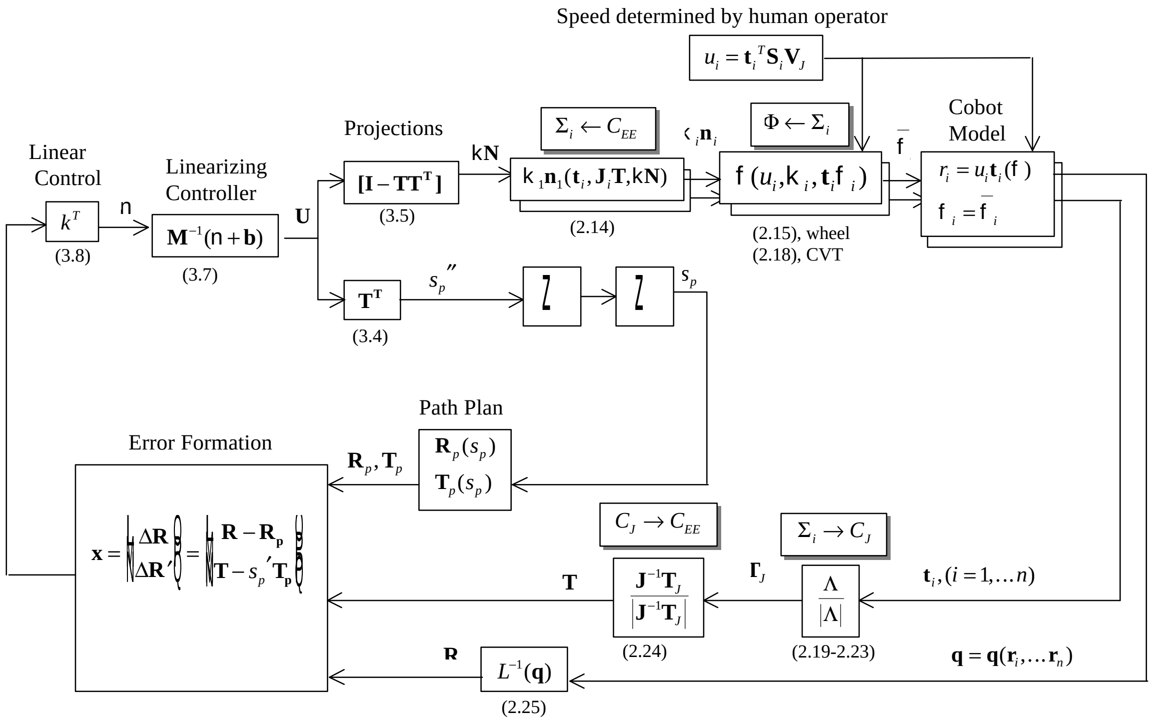

Figure 3. Block diagram of a path following controller implementation

The linearizing controller in cascade with the second order kinematic error equations appears to the outer loop controller as a linear system with the dynamics $x \quad \nu$. We implement a full-state linear controller of the form

$$\nu \quad \mathbf{k}^T\mathbf{x} \quad \mathbf{k}^T \begin{bmatrix} \mathbf{R} \\ \mathbf{R} \end{bmatrix}, \tag{3.8}$$

where $\mathbf{k}$ is a vector of feedback gains. The design of the outer loop controller is then quite simple. It may be accomplished using, for example, pole placement methods.

3.5 The whole picture

Figure 3 shows a block diagram that includes the outer loop linear controller, the inner linearizing loop, and the transformations into coupling space and steering space. The Cobot model is shown here as a composition of CVT models and the error vector is formed by differencing the monitored cobot position and heading with the position and heading chosen by the controller from the path plan. Equation numbers below each block indicate the pertinent equation in the text.

4. Example: The r-theta cobot

Figure 4 shows a revolute-prismatic cobot that we call the "r-theta cobot". Joint coordinate θ measures the angular displacement of a boom B around the fixed pivot P while joint coordinate r measures the linear displacement of a cart C from P. The r-theta cobot is cousin to the jib, a load-carrying device often used in automobile assembly. The boom swings overhead such that a load may be cable-suspended from the cart. The jib typically lacks motors or joint-coupling of any kind. The r-theta cobot, on the other hand, features a CVT to couple the speed of translation to the speed of rotation. One drive roller of a tetrahedral CVT is coupled to q while the other is coupled to r through a cable drive with transmission ratio $1/r_o$. By CVT steering, the r-theta cobot may define programmable constraints in the horizontal plane. For example, pre-defined paths can be laid out for parts handling.

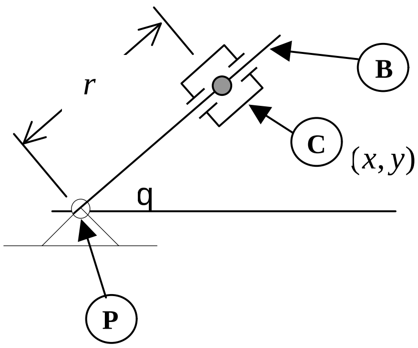

Figure 4. Schematic of the r-theta cobot

The jointspace vector $\boldsymbol{q}$ is $[r \quad \theta]^T$. The end-effector configuration space position vector $\mathbf{R}$ is $[x \quad y]^T$. The Jacobian relating jointspace velocity to C_{EE} space motion is as follows.

$$J(\mathbf{R}) \quad \frac{1}{r^2}\begin{bmatrix} rx & ry \\ y & x \end{bmatrix} \tag{4.1}$$

The transmission ratio $1/r_o$ appears in the matrix relating directions in jointspace to the single coupling space:

$$S_1 = \begin{bmatrix} 1/r_o & 0 \\ 0 & 1 \end{bmatrix} \tag{4.2}$$

The elements in Eq. 2.8 can be arranged into a Hessian style matrix if desired where

$$H_{11} \quad \frac{1}{r^3}\begin{bmatrix} y^2 & xy \\ xy & x^2 \end{bmatrix} \tag{4.3}$$

and $$H_{12} \quad \frac{1}{r^4}\begin{bmatrix} 2xy & (y^2 \quad x^2) \\ (y^2 \quad x^2) & 2xy \end{bmatrix} \tag{4.4}$$

and $$T^T \frac{J}{R} T \quad \begin{bmatrix} T^T H_{11} T \\ T^T H_{12} T \end{bmatrix}. \tag{4.5}$$

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

Figure 5 shows simulation results for the r-theta cobot under path following control starting at position *(1.2, 0)* and headed in the *y*-direction. The pre-planned path is a unit circle centered at the origin. The cart approaches then stays on the circular path. The locations of the reference cobot on the planned path chosen by the controller as simulation proceeded are indicated with circles while the corresponding positions of the cobot are asterisks.

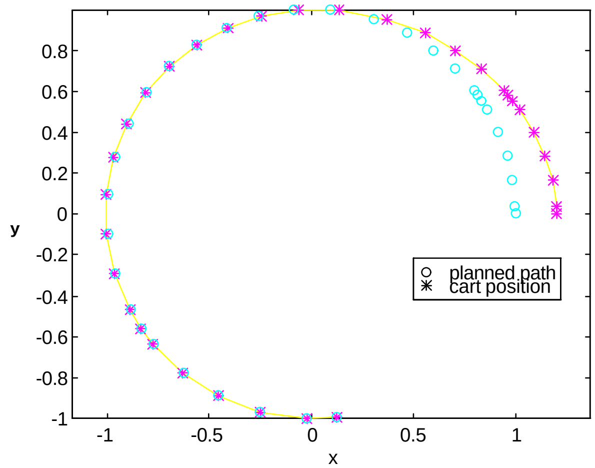

Figure 5. Following a circular path in simulation

5. Conclusion

An asymptotically stable path following controller has been developed for cobots. From a starting configuration away from the pre-planned path, 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 the end-effector and joint configuration space and the coupling spaces and steering space are essential for the development and implementation of general and extensible controllers.

Acknowledgements

The authors gratefully acknowledge the support of General Motors Corporation and the National Science Foundation, as well as the many insights provided by Witaya Wannasuphoprasit and Prasad Akella.

References

[1] J. Edward Colgate, W. Wannasuphoprasit and M.A. Peshkin. Cobots: Robots for Collaboration with Human Operators. Proceedings of the ASME Dynamic Systems and Control Division

DSC-Vol. 58, 433-440, 1996.

[2] W. Wannasuphoprasit, R.Brent Gillespie, J.Edward Colgate and Michael Peshkin, Cobot Control Proceedings of the IEEE International Conference on Robotics and Automation, 3571-3576, 1997.

[3] L. B. Rosenberg, Virtual Fixtures: Perceptual Overlays Enhance Operator Performance in Telepresence Tasks. PhD thesis, Stanford University, 1994.

[4] C. Sayers and R. Paul. An Operator Interface for Teleprogramming Empolying Synthetic Fixtures, Presence, Vol. 3, No. 4, 1994.

[5] Haugwitz, Otto. U.S. Patent 2,727,396, Dec. 20, 1955.

[6] Sordalen, O.J. Nakamura, Y., and Chung, W.J. Design of a Nonholonomic Manipulator. Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 8-13.

[7] C. Samson and K. Ait-Abderrahim, Feedback Control of a Nonholonomic Wheeled Cart in Cartesian Space. Proceedings of the 1991 international Conference on Robotics and Automation, 1136-1141, 1991.

[8] A. Isidori. Nonlinear Control systems: An Introduction, Lecture Notes in Control and Information Sciences, 72, Springer-Verlag, 1985.