CURVATURE BASED POINT STABILIZATION AND PATH FOLLOWING FOR COMPLIANT FRAMED WHEELED MODULAR MOBILE ROBOTS

by

Brian W. Albiston

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

The University of Utah

May 2003

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a thesis submitted by

Brian W. Albiston

This thesis has been read by each member of the supervisory committee and by majority vote has been found to be satisfactory.

Chair: Mark A. Minor

Sanford G. Meek

Samuel H. Drake

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the thesis of ____Brian W. Albiston________ in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the supervisory committee and is ready for submission to The Graduate School.

____
Date

____
Mark A. Minor
Chair: Supervisory Committee

Approved for the Major Department

____
Joseph C. Klewicki
Chair

Approved for the Graduate Council

____
David S. Chapman
Dean of The Graduate School

ABSTRACT

The compliant framed modular mobile robot is a new type of wheeled mobile robot for which the motion control problems of posture stabilization, path following, and trajectory tracking are studied. This robot has the advantages of a simple modular design that provides full suspension and steering capability without any additional components. Due to the flexible nature of the robot, the kinematics are much more complicated than typical mobile robot designs. It is shown that an equivalent curvature based model that improves mobility and reduces required traction forces can describe the kinematics.

A time invariant control law is developed for the posture stabilization problem and extended to compensate for nonideal initial conditions and system drift. It is shown through Lyapunov stability analysis that provided a special choice is made for the robot velocity and curvature, the system will stabilize to the origin. The trajectories generated are smooth and meet the desired requirements of the equivalent curvature based model. A similar control law is developed for the path following problem, and it is shown that this control law may also be utilized in the posture stabilization case. Hence, a single control law may be used for both the path following and posture stabilization problems.

The control laws developed are tested in simulation, and experimentally on the Partially Compliant Test Robot at the University of Utah. The results are presented and show the proposed control laws perform as expected in the point stabilization, path following, and trajectory tracking cases.

TABLE OF CONTENTS

Appendices

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Distance from axle midpoint to the wheel
c	Trajectory curvature, inverse of turning radius
c_1, c_2	Control variables
CFR	Compliant framed modular mobile robot
c_{path}	Instantaneous trajectory curvature of the path function
DSP	Digital signal processor
e	Polar distance coordinate
$F.V.$	Final signal value
F_m	Traction force due to moment
F_T	Net traction force
G_C	Wheel controller transfer function
G_F	Filter transfer function
G_P	Plant transfer function
I	Moment of inertia
k_1, k_2, k_3, k_4	Positive constant gains
K_C	Wheel controller gain
k_c	Positive curvature gain
k_u	Positive velocity gain
l	Length of the compliant frame element
M	Moment applied to axle
Max	Maximum signal value
O	Robot center point
$P.O.$	Percent overshoot
PCTR	University of Utah partially compliant test robot
R	Transverse reaction force
R_a	Transverse reaction force at each wheel
r_O	Trajectory radius of curvature at robot center, O
r_w	Wheel radius
s	Smooth path function
$\dot{s}_{des}$	Desired path velocity
$sign(\,)$	Signum function
T_P	Time to peak
u	Robot velocity
V	Lyapunov candidate function
x	Cartesian x coordinate of robot center, O
x_a, y_a, z_a	Actual robot Cartesian posture
x_{err}, y_{err}, z_{err}	Cartesian error coordinates
x_s, y_s, z_s	Desired robot Cartesian posture from path function

y	Cartesian y coordinate of robot center, O
Z	Smooth stabilizer
$\forall$	For all
ϕ	Robot orientation or heading with respect to the x-axis
ψ	Steering angle
α	Polar heading coordinate
Θ	Absolute wheel position
ε	Positive constant
ω	Robot center, O angular velocity or rate of change of ϕ
θ	Polar orientation coordinate
$\omega_{Desired}$	Desired wheel angular velocity
$\omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}$	Wheel angular velocities
$\mathbb{R}^n$	The n-dimensional Euclidean space
ϕ_s	Desired path following orientation
ω_{Sensed}	Wheel angular velocity sensed by encoders

CHAPTER I

INTRODUCTION

Point stabilization and path following control of a new breed of wheeled mobile robotic systems is the subject of this thesis. This new breed of robot is the compliant framed modular mobile robot (Figure 1). The concept of the Compliant Framed Robot (CFR) is unique in two ways. First, it uses a novel yet simple structure to provide suspension and highly controllable steering capability without adding any additional hardware to the system. This is accomplished by using flexible frame elements to couple rigid, differentially steered axles. In this study, the frame element provides compliant roll and yaw between the axles. Relative roll provides suspension capability in order to accommodate uneven terrain, and yaw allows the axles to independently change heading

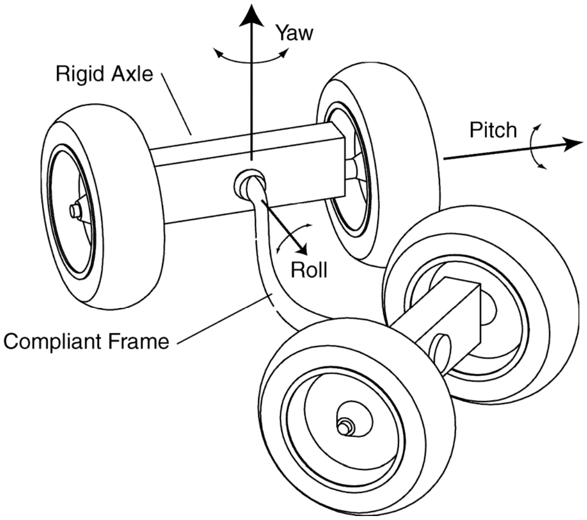


Figure 1: Compliant framed modular mobile robot.

for advanced steering capability. Steering and maneuvering of the system are thus accomplished via coordinated control of the axles. Since each axle can be steered independently, the system provides enhanced maneuverability in confined environments as well as the capability to control the shape of the frame.

A second unique aspect of the compliant framed mobile robot is its suitability for modular mobile robotics. Reconfigurable modular robotic systems have been of keen interest to researchers during the last decade due to their improved ability to overcome obstacles and perform multiple tasks using a single hardware platform. Towards this goal, numerous researchers have devoted their efforts to investigating minimalist homogenous robotic modules. These systems have examined reconfigurable manipulation [1, 2], mobility [3-5], or a combination therein [6-8]. Homogeneity of the modules is argued to reduce maintenance, offer increased robustness through redundancy, provide compact and ordered storage, and increase the adaptability of the systems [9, 10]. The compliant frame allows this concept to be extended to wheeled mobile robots by allowing a number of different vehicle configurations to be formed from a set of uniform frame and axle modules.

A limited number of compliant vehicles can be found in the literature, yet none possess a similar highly compliant frame whose deflection is controlled by coordinated actuation of the wheels. The earliest reference to compliant vehicles is a system proposed for planetary exploration that uses compliant members to provide roll and pitch degrees of freedom for suspension capability between the axles [11]. This concept was later extended [12] in a design where the frame of a vehicle was composed of at least one helical spring, but hydraulic cylinders were used to control the deflection. In each of these cases, compliance was introduced for accommodating terrain. More recent research has introduced compliance for accommodating measurement error and resulting wheel slip occurring between independently controlled axle units on a service robot [13]. This robot is similar in spirit to the compliant framed system in that it allows relative rotation between the axles, but this is provided by rotary joints connected to the ends of a frame with limited prismatic compliance. The system is intended for operation on flat surfaces in industrial service settings. As the author states, the system provides high levels of mobility, but since the axle units are coupled by a relatively nonconforming rigid frame, its ability to maneuver in confined environments will be limited [14]. Other flexible robots use actuated, articulated joints to provide similar relative motion between axles, as in the case of the Marsokhod rover [15] and other six wheeled research rovers with high relative DOF provided between axles modules [16].

The compliant frame mobile robotics system proposed here allows independent steering control of the axles with minimal slip and no additional hardware or actuators. For a more thorough presentation of the benefits of the CFR type platform see [17]. In this work, Schwenson studies the manufacturing, cost, and flexibility benefits of the CFR platform, along with two methodologies for steering the robot along a path. An initial study of CFR kinematics, steering, and point stabilization by Albiston et al. is presented in [18].

The subject of this thesis is point stabilization and path following control of the CFR concept. Discussion is limited to the configuration shown in Figure 1, consisting of two axle modules and one frame module. In Chapter II, point stabilization and path following control of the CFR is compared to existing point stabilization and path following control strategies. The unique challenges of the CFR type robot are also presented. In Chapter III an equivalent curvature based kinematic model is developed for the purpose of simplified motion planning, and the desired characteristics of a CFR control scheme are stated. In Chapter IV, a time invariant point stabilization control law is developed for this model based upon the work of Indiveri [19]. This control law is then extended to the path following case utilizing a signed polar representation similar in spirit to the work of Tayebi [20]. The experimental platform is described in Chapter V, including the methodology used to characterize the experimental platform. Chapter VI presents simulation and experimental results indicating the performance of the system, along with a discussion of those results. Concluding remarks and a discussion of future work are presented in Chapter VII.

CHAPTER II

BACKGROUND

Control of mobile robots and nonholonomic systems has received a great deal of attention in recent years, and many control alternatives have been proposed. Control of these systems is generally broken up into three types of problems: point stabilization, and path following, and trajectory tracking. The point stabilization problem consists of the robot starting from any initial condition and driving to a specified position and orientation, or “parking.” The path following problem consists of the robot tracking a memorized path. This path generally consists of a specified position and orientation in time. Trajectory tracking involves the robot tracking any desired velocity and curvature.

Point Stabilization

Point stabilization is generally viewed as the more difficult of the two problems. The challenge is that nonholonomic mobile robot systems have more degrees of freedom than controls. When represented in Cartesian space, they cannot be stabilized with a continuously differentiable, time-invariant feedback control law as pointed out in the famous paper by Brockett [21]. Various approaches have been undertaken to stabilize these systems such as time-varying [22], adaptive [23], discontinuous [24], and neural network based [25] strategies. For a thorough survey of nonholonomic control techniques see the review in [26].

The various strategies may be broken up in to three basic types: discontinuous time-invariant, continuous time-varying, and hybrid techniques that are some combination of the other two. All of the techniques make use of the fact that Brockett's Theorem shows that feedback stabilization is achievable if there is a discontinuity introduced in either the control law or time. Each of the techniques has advantages and disadvantages for various types of vehicle configurations.

Discontinuous time-invariant techniques are of two basic types: piecewise continuous and sliding mode controllers. Sliding mode controllers can provide good convergence rates by forcing the trajectory to slide on a manifold towards equilibrium, but often have problems with chatter as the controller switches control laws along the manifold. Piecewise continuous controllers are of several types, but most make use of a coordinate transformation introducing a discontinuity at the origin [19, 27, 28]. These controllers offer exponential convergence rates without the problem of chatter experienced by sliding mode control, and generally produce smooth natural looking paths [20].

The time-varying control laws that have been developed generally suffer from two problems. First, because time is discontinuous, exponential convergence usually cannot be guaranteed. Hence, these controllers normally suffer from slow convergence rates [26]. Second, the paths generated by this type of controller are generally not smooth or natural looking. They would require a robot with a high degree of maneuverability to follow the generated paths. Most of the hybrid techniques also suffer from these two problems.

Many of the proposed techniques make the simplifying assumption that the mobile robot is a simple unicycle type vehicle. These techniques are therefore well suited to vehicles with only one axle to control, the ability to perform a zero radius turn, and easily reverse direction. By its very nature, the CFR vehicle is limited in the minimum radius turn it can perform. The CFR also has independently steered axles with compliant coupling due to the flexible frame. This adds to the complexity of the CFR kinematics, and thus the simple assumptions of a unicycle type vehicle do not apply.

Path Following

Mobile robot path following is generally viewed as a less difficult problem than point stabilization because path movement introduces a discontinuity into the system and therefore the limitations of Brockett's Theorem do not apply. Thus, a continuously differentiable, time-invariant feedback control law is possible.

Like point stabilization, many approaches have been taken in the literature to accomplish path following. Most simply minimize the tangential, normal, and angular error between the current robot position and orientation and the current path position and orientation. An example of such an approach is given by Diaz del Rio [29].

In most path following techniques, the path is defined as a specified position and orientation in time. Hence, the path proceeds without any regard for the current states of the robot. Problems may arise when the path is defined by this method. Large perturbations may force the robot to remain at rest, yet the path will unavoidably continue its course [29]. Many recent techniques base the path velocity on the current states of the robot. The benefit to this approach is that the path velocity may be defined as to slow or stop altogether if the error grows beyond some specified bound. In the case of the path following algorithm developed for the CFR in section IV, the path velocity is based upon the states of the robot in such a way as to guarantee the asymptotic stability of the control algorithm.

Trajectory Tracking

Trajectory tracking is generally considered as the most basic of the three control problems. It simply consists of the robot tracking a specified velocity and curvature trajectory. This trajectory may be generated by a path following, point stabilization, or other control law, or it may be an input into the system by a human operator. It is considered as the most basic control problem, because generally a robot must be able to track a desired trajectory in order to achieve any of the previous control problems mentioned. Otherwise, no guarantee could be made that the robot would follow the trajectory output by the point stabilization or path following controller. For the CFR robot, the ability to track a trajectory shall flow from the controller dynamic extension performed in Chapter IV.

Mathematical Preliminaries

Nonlinear control theory is used extensively in the development of the point stabilization, path following, and trajectory tracking controllers in Chapter IV. Several mathematical theorems related to nonlinear control theory are repeated here for reader convenience.

Global Asymptotic Lyapunov Stability

From Khalil Theorem 4.2 [30], let a general nonlinear, time invariant system $\dot{x} = f(x)$, have an equilibrium point at $x = 0$. System equilibrium points not located at

$x = 0$ may be moved to the origin through a suitable coordinate transformation. Define a continuously differentiable Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$ such that

$$\begin{aligned} &V(0) = 0 \text{ and } V(x) > 0,\ \forall \text{ x} \neq 0 \\ &\|\text{x}\| \to \infty \ \Rightarrow\ V(x) \to \infty \\ &\dot{V}(x) < 0,\ \forall\ x \neq 0 \end{aligned} \tag{1}$$

then the origin $x = 0$ is globally asymptotically stable. This implies the system has only one equilibrium point.

<u>LaSalle's Theorem</u>

From Khalil Theorem 4.4 [30], let $\Omega \subset D$ be a compact set that is positively invariant with respect to $\dot{x} = f(x)$. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω. Let *E* be the set of all points in Ω where $\dot{V}(x) = 0$. Let *M* be the largest invariant set in *E*. Then every solution starting in Ω approaches *M* as $t \to \infty$.

This may be extended to asymptotic stability by Khalil Corollary 4.2 [30], let $x = 0$ be an equilibrium point for $\dot{x} = f(x)$. Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable, radially unbounded, positive definite function such that $\dot{V}(x) \leq 0$ for all $x \in \mathbb{R}^n$. Let $S = \left\{ x \in \mathbb{R}^n \mid \dot{V}(x) = 0 \right\}$ and suppose that no solution can stay identically in *S*, other than the trivial solution $x(t) \equiv 0$. Then, the origin is globally asymptotically stable. The benefit of LaSalle's theorem is that it does not require $\dot{V}$ to be negative definite for global asymptotic stability.

Barbalat’s Lemma

From Khalil Lemma 8.2 [30], let $\phi : \mathbb{R} \to \mathbb{R}$ be a uniformly continuous function on $[0, \infty)$. Suppose that

$$\lim_{t \to \infty} \int_0^t \phi(\tau) d\tau \tag{2}$$

exists and is finite. Then, $\phi(t) \to 0$ as $t \to \infty$.

Extension of Barbalat’s Lemma

Barbalat’s lemma is extended to show global asymptotic stability in Khalil Theorem 8.4 [30]. Let $D \subset \mathbb{R}^n$ be a domain containing $x = 0$ and suppose *f(t,x)* is piecewise continuous in *t* and locally Lipschitz in *x*. Let *V* be a continuously differentiable function such that

$$\begin{aligned} &W_1(x) \le V(x) \le W_2(x) \\ &\dot{V}(x) \le -W(x) \end{aligned} \tag{3}$$

for all $x \in D$, where *W₁(x)* and *W₂(x)* are continuous positive definite functions and *W(x)* is a continuous positive semidefinite function on *D*. Choose *r* > 0 such that $B_r \subset D$ and let $\rho < \min_{\|x\|=r} W_1(x)$. Then all solutions of $\dot{x} = f(x)$ with $x(0) \in \{x \in B_r \mid W_2(x) \le \rho\}$ are bounded and satisfy

$$W(x) \to 0 \text{ as } t \to \infty \tag{4}$$

If all of the above assumptions hold globally and $W_1(x)$ is radially unbounded, the statement is true for all $x(0) \in \mathbb{R}^n$.

Baccioti's Theorem

Bacciotti [31] defines a cascade system as

$$\begin{aligned} \dot{y} &= \phi(y,z) \\ \dot{z} &= \psi(z,u) \end{aligned} \tag{5}$$

where $y \in \mathbb{R}^l$, $z \in \mathbb{R}^{n-l}$, and $u \in \mathbb{R}^m$ and $\phi(0,0) = \psi(0,0) = 0$. From Bacciotti's Theorem 19.2 this system is smoothly stabilizable if both the y and z subsystems are stabilizable, $n - l = m$, $\psi(z,u) = u$, and the y subsystem introduces a smooth stabilizer $z = k(y)$. The proof of this is shown by performing a coordinate change

$$\begin{aligned} Y &= y \\ Z &= z - k(y). \end{aligned} \tag{6}$$

The cascade system then becomes

$$\begin{aligned} \dot{Y} &= \phi(Y, Z + k(Y)) = \Phi(Y,Z) \\ \dot{Z} &= \dot{z} - (Dk)(y)\dot{y} = u - (Dk)(Y)\Phi(Y,Z). \end{aligned} \tag{7}$$

If a stabilizing feedback $u = -Z + (Dk)(Y)\Phi(Y,Z)$ is now applied the closed loop system becomes

$$\begin{aligned} \dot{Y} &= \Phi(Y,Z) \\ \dot{Z} &= -Z. \end{aligned} \tag{8}$$

Hence, as $t \to \infty$, $Z \to 0$. When $Z = 0$, the equation for $\dot{Y}$ will become $\dot{Y} = \dot{y} = \Phi(Y,0) = \phi(y,k(y))$ which is asymptotically stable at the origin. In the original coordinates the feedback becomes $u = (Dk)(y)\phi(y,z) + k(y) - z$.

CHAPTER III

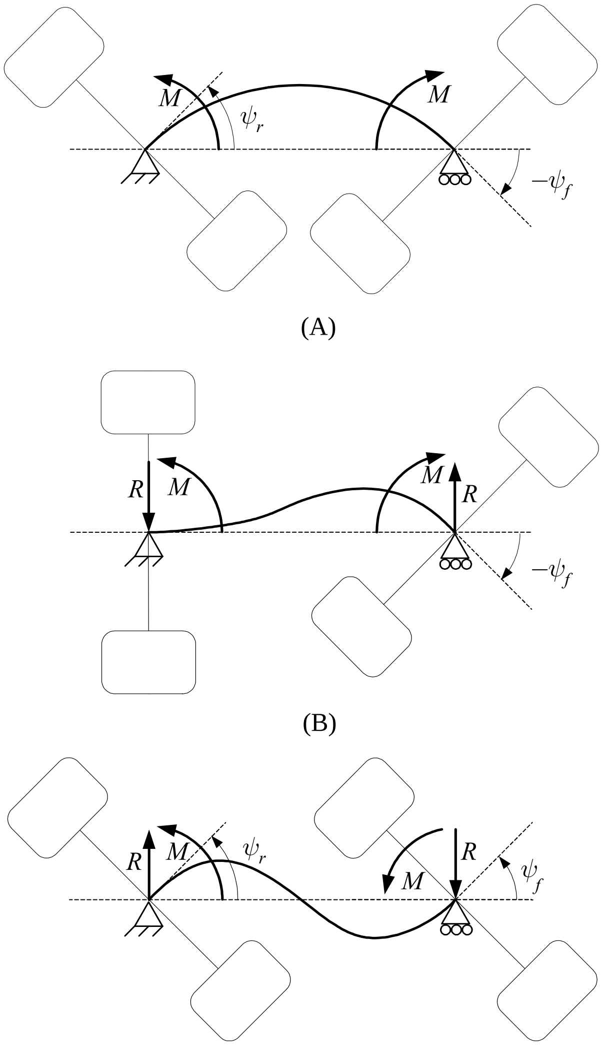
KINEMATIC MODEL

As stated in Chapter II, the compliant framed mobile robot has much more complex steering kinematics than unicycle type vehicles since it possess independently steered axles with compliant coupling. Each of the axles must apply a particular force and moment on the frame module to maintain a desired frame deflection for vehicle steering. The force and moment required will vary depending on the stiffness of the frame element and the frame deflection magnitude and shape. These required forces and moments will in turn be transferred to the wheels where they will affect the traction force of each of the wheels.

For a particular surface the robot is driving across, there will be maximum traction force possible at each of the wheels depending on the coefficient of friction between the wheels and the driving surface. This traction force will be split between the amount available for useful propulsion, and the amount necessary to deflect the frame element for steering. It is strongly desired to maintain the percentage of the maximum traction force used for robot propulsion as high as possible for two reasons. First, the positional tracking accuracy of the robot may be greatly affected as wheel slip occurs. The position of the robot is determined by integrating the angular velocity of each of the wheels from encoder feedback. If one of the wheels slips, then an error will be introduced into the positional tracking of the amount of the slip. The second reason to maintain the percentage of wheel traction force used for propulsion as high as possible is to increase the capability of the robot to do useful work. The greater the percentage of useful traction force available, the greater the amount of payload that may be carried or towed by the robot. Correspondingly, increased frame deflection moments and forces decrease the amount of payload that may be carried or towed.

The variations in axle forces and moments imposed by the deflection of the frame can be modeled by considering the frame module as a simply supported beam [32]. Each axle is viewed as imposing a different end condition on each end of the beam (Figure 2). The figure shows three fundamental shapes of the frame imposed by three axle end condition cases. All axle configurations can be classified as some combination of these fundamental shapes.

Each of the axle end condition cases could potentially result from driving the robot by a particular method. Case 1 would result when the axles are coordinated to maintain a pure bending, constant curvature condition, which is the equivalent of $\psi_f = -\psi_r$. This would be equivalent to the first mode of bending for a simply supported beam. This case would result in an end condition of an equal and opposite moment being applied to each end of the beam, with zero transverse force. Cases 2 and 3 result in very similar end conditions being applied to the beam, but are the result of different steering methods. Both cases result when the axles apply end conditions of a moment and transverse force to the beam. Case 2 is realized when the robot is steered in a car-like fashion, with only the front axle being directed towards the target, and the rear axle maintaining ψ_r =0. This case results in nonequal similarly directed moments, and equal opposing transverse forces. Case 3 may be produced when each axle is steered to the



(A)

(B)

(C)

Figure 2: Euler-Bernoulli beam cases. (A) Case 1 (B) Case 2 (C) Case 3

goal independently, i.e., $\psi_f = -\psi_r$. Each axle is directed towards the target resulting in end conditions of equal similarly directed moments and equal opposing transverse forces being applied, by the axles, to each end of the beam. Cases 2 and 3 are equivalent to the Single Independent Axle (SIA) and Dual Independent Axle (DIA) cases studied by Schwenson [17].

In order to evaluate which of these three fundamental cases will require the least power and traction, the axle moments and reaction forces for a given steering angle ψ, (Figure 2) can be described for each axle by the equations,

$$\begin{aligned} M_f &= \frac{2EI}{l}\left(2\psi_f - \psi_r\right) \\ R_f &= \frac{M_f - M_r}{l} \\ M_r &= \frac{2EI}{l}\left(2\psi_r - \psi_f\right) \\ R_r &= -R_f \end{aligned} \tag{9}$$

where *l* is the robot length, *E* is the modulus of elasticity, *I* the beam moment of inertia, subscript *f* for the front axle and subscript *r* for the rear axle. Using *a*, the axle half width as shown in Figure 3, the traction forces required to produce these moments, *M* and

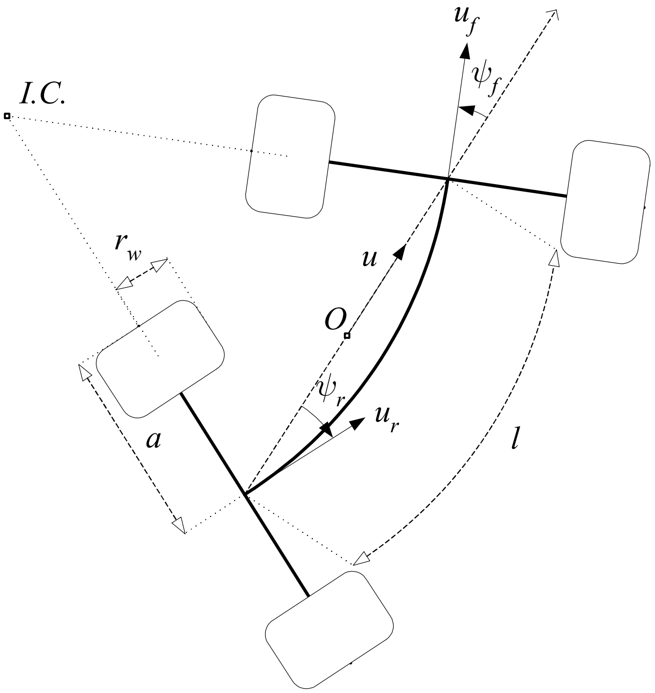


Figure 3: Compliant framed mobile robot turning kinematics.

lateral reaction forces, *R* at each wheel, may be found from the equations

$$\begin{aligned} F_m &= \frac{M}{2a} \quad R_a = \frac{R}{2} \\ F_T^2 &= F_m^2 + R_a^2 \end{aligned} \tag{10}$$

where F_m is the longitudinal force at each wheel to produce *M*, R_a is the lateral reaction at each wheel to produce *R*, and F_T is the norm of F_m and R_a, or the net traction force required to produce *M* and *R*. These moments and forces are shown for each of the three cases in Table 1. As inspection of the equations reveals, the minimum traction force required for steering the robot is realized when the transverse reaction force is kept at a minimum, or zero. This occurs when the system is deflected in the first mode of beam bending, Case 1. Maintaining the robot in this mode while steering will minimize the required traction force. This will allow the robot to exert larger forces for towing a load or accelerating the robot, and reduce the probability of wheel slip. As it turns out, Case 1 also results in the smallest turning radius and correspondingly maximum maneuverability

Table 1: Robot traction and steering performance. *K*=*a*/*l*.

Case	Max. Moment, M	Reaction Force, *R*	Traction Force, F_T	Turning Radius (approx. for smallψ_f)
1	$\frac{2EI\psi_f}{l}$	0	$\frac{EI\psi_f}{Kl^2}$	$\approx \frac{l}{2\psi_f} = \frac{0.183}{\psi_f}$
2	$\frac{4EI\psi_f}{l}$	$\frac{6EI\psi_f}{l^2}$	$2\frac{EI\psi_f}{Kl^2}\sqrt{1+9K^2}$	$\approx \frac{l}{2}\sqrt{1+\frac{4}{\tan^2(\psi_f)}}$ $\approx \frac{0.366}{\psi_f}$
3	$\frac{6EI\psi_f}{l}$	$\frac{12EI\psi_f}{l^2}$	$3\frac{EI\psi_f}{Kl^2}\sqrt{1+16K^2}$	∞

for a given ψ_f, Table 1. Radius of curvature can be realized by considering the intersection of velocity normals relative to the center of the frame, or the instantaneous center as denoted by *I.C.* in Figure 3.

It is therefore desired to utilize the fact that Case 1 requires minimum traction and provides maximum steering capability for the robot to develop an equivalent kinematic model. Further illustration of the system in the Case 1 configuration is shown in Figure 4. Using this scenario to derive the kinematics, the net system position and orientation of the robot may be described by an equivalent posture attached to point, *O*, located at the center of a line drawn between the axle midpoints. Assuming that the front and rear axles

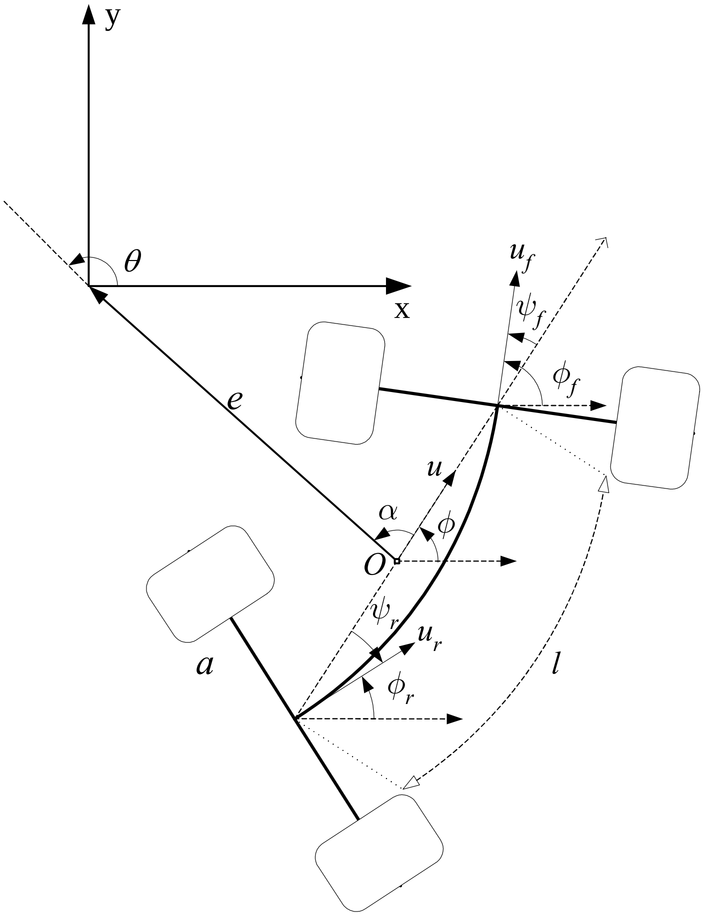


Figure 4: Equivalent model kinematics.

steering angle ψ have equal absolute values and opposite signs, the familiar unicycle type Cartesian kinematics for the center point are defined as,

$$\begin{aligned}\dot{x} &= u\cos\phi \\ \dot{y} &= u\sin\phi \\ \dot{\phi} &= \omega\end{aligned} \tag{11}$$

where x and y are the Cartesian coordinates of a moving coordinate frame attached to point O, that describes the robot equivalent posture. The variable u represents the velocity of the coordinate frame moving in a heading ϕ relative to the global frame, and ω is the rate of change of ϕ, or the equivalent robot angular velocity. Equation (11) can then be derived in terms of the individual axle kinematics commonly found in the literature.

Before deriving the equivalent kinematic relationships, several desirable steering conditions must be described:

1. As opposed to the case of a unicycle type robot, the compliant framed mobile robot cannot turn with a zero radius of curvature. Thus, the robot should proceed only on paths of bounded curvature.
2. For simplicity, it is desired that the robot's motion proceed only in the forward direction. It is not desired to have cusps in the robots motion.
3. The pure bending condition, $\psi_f = -\psi_r$, should be maintained throughout all turning maneuvers to ensure Case 1, thereby increasing capability and also preventing wheel slip.

The point O can also be represented in polar-like coordinates where

$$\begin{aligned} e &= \sqrt{x^2 + y^2} \\ \theta &= ATAN2(-y, -x) \\ \alpha &= \theta - \phi. \end{aligned} \tag{12}$$

The system state equations then may be defined as,

$$\begin{aligned} \dot{e} &= -u\cos\alpha \\ \dot{\alpha} &= -\omega + u\frac{\sin\alpha}{e} \\ \dot{\theta} &= u\frac{\sin\alpha}{e}. \end{aligned} \tag{13}$$

The advantage of this polar-like representation is that the state itself is not defined for *e* = 0, and therefore Brockett's Theorem does not hold and a smooth time invariant state feedback control law for global asymptotic stability is possible [27], where several controllers in the literature have been similarly implemented [19, 27, 28].

Using this transformation, the angular velocity of the robot center point, *O*, can be described as a function of the bounded curvature, or inverse turning radius, *c*.

$$\omega = \dot{\phi} = uc \tag{14}$$

Substitution into the polar state equations (13) yields

$$\begin{aligned} \dot{e} &= -u\cos\alpha \\ \dot{\alpha} &= -u\left(c - \frac{\sin\alpha}{e}\right) \\ \dot{\theta} &= u\frac{\sin\alpha}{e} \end{aligned} \tag{15}$$

The angular velocity of the robot equivalent posture can be described as a function of the steering angles ψ_f and ψ_r (Figure 4). Once again, it is assumed for simplicity and to ensure the pure bending condition that $\psi_f = -\psi_r$. From Figure 3 expressions for the radius and curvature of the robot center point as a function of the steering angle ψ and frame element length *l* can be shown to be

$$r_O = \frac{l}{2\psi}\cos\psi \qquad c = \frac{1}{r_O} = \frac{2\psi}{l\cos\psi} \tag{16}$$

Substituting (16) into (14), the angular velocity of *O* then becomes

$$\dot{\phi} = uc = u\frac{2\psi}{l\cos\psi}. \tag{17}$$

The steering angle ψ and its time derivative $\dot{\psi}$, may be solved for numerically by rearranging (16) to form the equation,

$$\frac{\psi}{\cos\psi} = \frac{c\,l}{2}. \tag{18}$$

Referring again to Figure 4, the velocities u_f and u_r are shown to be

$$u_f = u_r = \frac{u}{\cos\psi} \tag{19}$$

and it can be deduced that the angular velocities are,

$$\begin{aligned} \phi_f &= \phi + \psi \quad \phi_r = \phi - \psi \\ \dot{\phi}_f &= \dot{\phi} + \dot{\psi} \quad \dot{\phi}_r = \dot{\phi} - \dot{\psi}. \end{aligned} \tag{20}$$

Thus, using this equivalent kinematic formulation, the complicated kinematics of the CFR may be controlled by admitting familiar steering algorithms that utilize common kinematics. Given *u* and *c*, the respective linear velocity and path curvature of the center point *O,* the steering angle ψ may be found from equation (18). The robot angular velocity may then be found from (17), which may be used to find the angular velocity of each axle using (20). The linear velocity of each axle may then be found from equation (19). The angular velocity of each wheel may then be determined, assuming no wheel slip and the pure bending condition, using the velocity Jacobian

$$\begin{bmatrix} \omega_{f_l} \\ \omega_{f_r} \\ \omega_{r_l} \\ \omega_{r_r} \end{bmatrix} = \begin{bmatrix} \frac{1}{r_w} & -\frac{a}{r_w} & 0 & 0 \\ \frac{1}{r_w} & \frac{a}{r_w} & 0 & 0 \\ 0 & 0 & \frac{1}{r_w} & -\frac{a}{r_w} \\ 0 & 0 & \frac{1}{r_w} & \frac{a}{r_w} \end{bmatrix} \begin{bmatrix} u_f \\ \dot{\phi}_f \\ u_r \\ \dot{\phi}_r \end{bmatrix} \tag{21}$$

where r_w is the wheel radius as shown in Figure 4, and ω is the angular velocity of each wheel.

CHAPTER IV

TIME INVARIANT CONTROL LAWS

It is desired to develop smooth, time invariant control laws for point stabilization and path following of the compliant framed mobile robot utilizing the equivalent kinematic model developed in Chapter III. A control law for point stabilization will be developed first utilizing Lyapunov techniques, and then this control law shall be extended to the path following case.

Point Stabilization

As discussed in Chapter II, many of the nonholonomic point stabilization techniques common in the literature are not suitable for the CFR. The control algorithm must meet the desired characteristics set forth in Chapter III: 1. Generated paths should be of bounded curvature. 2. Paths should not have cusps. 3. Algorithm should accommodate the pure bending condition, $\psi_f = -\psi_r$. In addition to these three characteristics, the control law must also have no discontinuous changes in curvature or velocity, including the instant from time 0 to time greater than zero. In other words, the initial conditions of the controller and the robot must match. This last requirement shall also prove critical in path following.

Indiveri [19] suggests a control law for a bicycle that is of simple yet effective design that meets conditions 1, 2, and 3, and may be modified to meet the continuous curvature requirement. The control law is piecewise continuous and makes use of the polar coordinate transformation performed in Chapter III equation (12). As mentioned in Chapter III the states are not defined for $e = 0$, and therefore Brockett's Theorem does not hold and a smooth time invariant state feedback control law for global asymptotic stability is possible. The controller offers exponential convergence rates and produces smooth natural looking paths.

Based upon the positive definite and radially unbounded Lyapunov function,

$$V \equiv \frac{1}{2}\left(\alpha^2 + k_2\theta^2\right) \tag{22}$$

Indiveri suggests defining the forward velocity as

$$u = k_1 e \tag{23}$$

where k_1 and k_2 are positive constant gains. This ensures the velocity will always proceed in the forward direction, since e will always be positive, and hence prevent cusps. Using this definition, the polar state equations become

$$\begin{aligned} \dot{e} &= -k_1 e \cos\alpha \\ \dot{\alpha} &= -k_1 e\left(c - \frac{\sin\alpha}{e}\right) \\ \dot{\theta} &= k_1 \sin\alpha. \end{aligned} \tag{24}$$

The time derivative of the quadratic Lyapunov function (22) upon substitution of the polar state equations (24) becomes

$$\dot{V} = \alpha\dot{\alpha} + k_2\theta\dot{\theta} = k_1\left(\alpha\sin\alpha + k_2\theta\sin\alpha - \alpha e c\right). \tag{25}$$

It is desired to show that (25) is negative definite for asymptotic stability, which suggests the choice of *c* as

$$c = \frac{\sin\alpha}{e} + k_2\frac{\theta}{e}\frac{\sin\alpha}{\alpha} + k_3\frac{\alpha}{e}. \tag{26}$$

The time derivative of the Lyapunov function now becomes

$$\dot{V} = -k_1k_3\alpha^2 \leq 0 \tag{27}$$

which is negative semidefinite for all k_1 and $k_3 > 0$. Because (27) is not negative definite, the requirements for Lyapunov global asymptotic stability are not met.

Indiveri shows that the system is globally asymptotically stable through the use of the following logic. Equation (27) and the fact that V is positive definite and radially unbounded implies by Lyapunov's stability theorem ([30], Theorems 4.2 and 4.4) that V tends towards a non-negative finite limit. As a result, we have,

$$\lim_{t\to\infty}\alpha = \overline{\alpha} \tag{28}$$

The fact that $\ddot{V} = -2k_1k_3\alpha\dot{\alpha}$ is bounded denotes that $\dot{V}$ is uniformly continuous. This implies by Barbalat's Lemma ([30], Lemma 8.2 and Theorem 8.4) that $\dot{V}$ tends to zero. This necessarily indicates that $\overline{\alpha} = 0$. From this result and (24) we see that

$$\lim_{t\to\infty}\theta = \overline{\theta} \tag{29}$$

From the choice of *u* and *c* in (23) and (26) the polar state equations become

$$\begin{aligned}\dot{e} &= -k_1 e\cos\alpha\\ \dot{\alpha} &= -k_1\left(k_3\alpha + k_2\theta\frac{\sin\alpha}{\alpha}\right)\\ \dot{\theta} &= k_1\sin\alpha.\end{aligned} \tag{30}$$

From above we know $\alpha \to 0$, $\theta \to \overline{\theta}$ and the state equation for $\dot{\alpha}$ is uniformly continuous. Substituting $\alpha \to 0$, $\theta \to \overline{\theta}$ into the state equation (30) for $\dot{\alpha}$, we can show from Barbalat's Lemma that the limit

$$\lim_{t\to\infty}\dot{\alpha} = -k_1k_2\overline{\theta} = 0. \tag{31}$$

From this limit it can be ascertained that the limit value $\overline{\theta}$ of θ must be zero. Given this result and substituting $\alpha \to 0$ into the state equation (30) for $\dot{\theta}$, it can be concluded that $\dot{\theta}$ tends asymptotically towards zero. Thus we now know that

$$\begin{aligned}\alpha \to 0; \quad \dot{\alpha} \to 0\\ \theta \to 0; \quad \dot{\theta} \to 0\end{aligned} \tag{32}$$

as $t \to \infty$. From (32) we know that the state equation for $\dot{e} = -k_1 e\cos\alpha$ will become

$$\dot{e} \to -k_1 e \quad \Rightarrow \quad e \to 0 \tag{33}$$

as $t \to \infty$, and therefore e asymptotically converges to zero. Hence, provided the special choice for velocity u and curvature c in (23) and (26), the states (e,α,θ) must necessarily globally asymptotically converge to the origin [19].

Global asymptotic stability may also be proven through use of the extension to Barbalat's lemma, Theorem 8.4 in Khalil [30]. It may easily be seen that the Lyapunov function (22) may be bounded from above and below by two continuous positive definite functions $W_1(x)$ and $W_2(x)$ as in (3). In fact, $W_1(x)$ and $W_2(x)$ may simply be chosen as being equal to V. It may also be seen that the derivative of the Lyapunov function (27) is less than or equal to the negative of some continuous positive semidefinite function $W(x)$. The domain may be chosen as $D = x \in \mathbb{R}^n$, and therefore r may be chosen on $(0,\infty)$. Hence any $x(0)$ shall satisfy $x(0) \in \{x \in B_r \mid W_2(x) \le \rho\}$. All of the assumptions hold globally and $W_1(x)$ is radially unbounded; therefore $W(x) \to 0$ as $t \to \infty$ for all $x \in \mathbb{R}^n$. This implies that the derivative of the Lyapunov function (27) is negative definite, and hence the system is globally asymptotically stable by an invariance approach and Lyapunov's stability theorem.

Implementation Issues

Several issues arise when one tries to implement the above control on an actual mobile robot. First, the choice of u as $u = k_1 e$ implies that u could be very large if the robot initial position is too distant from the origin and velocity saturation may occur. Indiveri shows that if the choice of u is changed to

$$u = k_1 e : \mathrm{sat}\left(k_1 \mathrm{e}, \overline{u}\right), \tag{34}$$

where $\mathrm{sat}(\mathrm{k}_1\mathrm{e},\overline{u})$ is a positive continuous saturation function that prevents the proportional control input *u* to grow larger than some upper bound $\overline{u}$, the convergence of the system is still maintained.

The behavior of the system will also depend on the choice of the constant gains k_1, k_2, and k_3. The velocity gain k_1 may simply be chosen by how aggressively it is desired that the system minimize the polar error *e*. By linearization of the system state equations and calculation of the eigenvalues, Indiveri suggests the choice of k_2, and k_3 as

$$k_2 > 1 \qquad 2 < k_3 < k_2 + 1. \tag{35}$$

The proof of the above is not repeated here and may be found by the interested reader in [19].

<u>Controller Dynamic Extension</u>

The major implementation issue, especially in the case of the compliant framed mobile robot, is that the robot initial conditions will rarely match the initial conditions for *u* and *c* output by the controller in (34) and (26). For instance, if the initial states were $e = 1, \quad \alpha = {}^{\pi}\!/_{4}, \quad \theta = {}^{3\pi}\!/_{4}$ with k_1=1, k_2=2, and k_3=2.9, the initial conditions output by the controller would be a velocity of 1 m/s and a curvature of 7.2 m^{-1}. In general, the initial robot conditions will be a velocity of 0 m/s and curvature of 0 m^{-1}. This discrepancy between the desired and actual initial robot velocity and curvature dictate that the robot will not stabilize to the desired stabilization point. In addition, perturbations throughout the robot path resulting in error between the desired *u* and *c* and the actual *u* and *c* will result in error of the final stabilization point.

This issue may be resolved with a change in the feedback where the controller dynamics are extended. The extended system may be written as a cascade system of the form

$$\begin{aligned}
&\dot{y} = f_1(y,z)\\
&\dot{z} = f_2(z,u)\\
&y = (e,\alpha,\theta) \in \mathbb{R}^l = \mathbb{R}^3\\
&u = (u_d, c_d) \in \mathbb{R}^m = \mathbb{R}^2\\
&z \in \mathbb{R}^m = \mathbb{R}^2
\end{aligned} \tag{36}$$

where $f_1(0,0) = f_2(0,0) = 0$, f_1 is the system in (30), f_2 is the extended system, and u_d and c_d are the desired velocity and curvature output from the controller. The smooth stabilizers

$$\begin{aligned}
z_1 &= e_u\\
z_2 &= e_c
\end{aligned} \tag{37}$$

are admitted where

$$\begin{aligned}
e_u &= u_d - u_a\\
e_c &= c_d - c_a
\end{aligned} \tag{38}$$

and u_a and c_a are the actual measured feedback velocity and curvature. Differentiating (38) with respect to time

$$\begin{aligned}
\dot{e}_u &= \dot{u}_d - \dot{u}_a\\
\dot{e}_c &= \dot{c}_d - \dot{c}_a
\end{aligned} \tag{39}$$

which can be rearranged such that,

$$\begin{aligned}\dot{u}_a &= \dot{u}_d - \dot{e}_u \\ \dot{c}_a &= \dot{c}_d - \dot{e}_c.\end{aligned} \tag{40}$$

Now let the time derivatives of the smooth stabilizers become

$$\begin{aligned}\dot{z}_1 &= \dot{e}_u = -k_u e_u \\ \dot{z}_2 &= \dot{e}_c = -k_c e_c\end{aligned} \tag{41}$$

and,

$$\begin{aligned}\dot{u}_a &= \dot{u}_d + k_u\left(u_d - u_a\right) \\ \dot{c}_a &= \dot{c}_d + k_c\left(c_d - c_a\right)\end{aligned} \tag{42}$$

where k_u and k_c are positive gains. Upon substitution of (41) into (39) and utilizing the definition for e_u and e_c the equations for the time derivative of the velocity and curvature become

$$\begin{aligned}\dot{u}_a &= \dot{u}_d + k_u e_u \\ \dot{c}_a &= \dot{c}_d + k_c e_c\end{aligned} \tag{43}$$

which matches the definition for the input *u* in Bacciotti's Theorem 19.2 [31]. Hence, by Bacciotti's Theorem 19.2 the overall system is smoothly stabilizable. An alternate formulation of this proof, that more closely follows the proof of Bacciotti, may be found in Appendix A.

Essentially, the addition of the smooth stabilizers represents the actual velocity and curvature of the robot as additional states of the system. The stabilizing control (43) minimizes the error between the actual robot velocity and curvature and the desired velocity and curvature output by the point stabilization controller in (35) and (26). Therefore, it is no longer required that the robot initial conditions and the controller outputs match. The state equations of the extended system with the control inputs inserted now become

$$\begin{aligned}
\dot{u}_a &= k_u(k_1 e - u_a) + \dot{u}_d \\
\dot{c}_a &= k_c\left(\frac{\sin\alpha}{e} + k_2\frac{\theta}{e}\frac{\sin\alpha}{\alpha} + k_3\frac{\alpha}{e} - c_a\right) + \dot{c}_d \\
\dot{e} &= -u_a\cos\alpha \\
\dot{\alpha} &= -u_a\left(c_a - \frac{\sin\alpha}{e}\right) \\
\dot{\theta} &= u_a\frac{\sin\alpha}{e}.
\end{aligned} \tag{44}$$

One benefit of the formulation of u_a and c_a in (43) is that the system will now track any desired velocity and curvature. Generally this trajectory will be generated by the control law in (34) and (26), but suppose u_d and c_d in (43) are replaced with u_t and c_t, which could be any desired velocity and curvature trajectory. This trajectory could be input by a human operator or other means. The system state equations would then become

$$\begin{aligned}
\dot{u}_a &= k_u(u_t - u_a) + \dot{u}_t \\
\dot{c}_a &= k_c(c_t - c_a) + \dot{c}_t \\
\dot{e} &= -u_a \cos\alpha \\
\dot{\alpha} &= -u_a \left(c_a - \frac{\sin\alpha}{e} \right) \\
\dot{\theta} &= u_a \frac{\sin\alpha}{e}.
\end{aligned} \tag{45}$$

and the system would still be guaranteed to stabilize to the desired trajectory. This would prove useful for many types of situations when point stabilization or path following is not feasible and a "human in the loop" is preferred.

Path Following

Ideally it would be desired to extend the point stabilization controller developed above to the path following case. The point stabilization controller already meets the requirements set forth in Chapter III, and it would be convenient to use the same controller for both cases. In fact, the point stabilization controller may be utilized as a path following controller provided a special coordinate transformation is performed.

The kinematics of the path is developed in Figure 5, where the path is a smooth directed function *s,* and *O* represents the equivalent center posture of the robot. The desired robot posture *s* is represented as a moving coordinate system tangent to the path. Thus, the desired robot velocity now becomes the vector $\dot{s}$, which is also oriented tangent to the path. The desired orientation becomes ϕ_s, which is the angle from the global x-axis to the path tangent. The polar coordinates now symbolize the polar error between the desired robot posture *s* and the actual robot location.

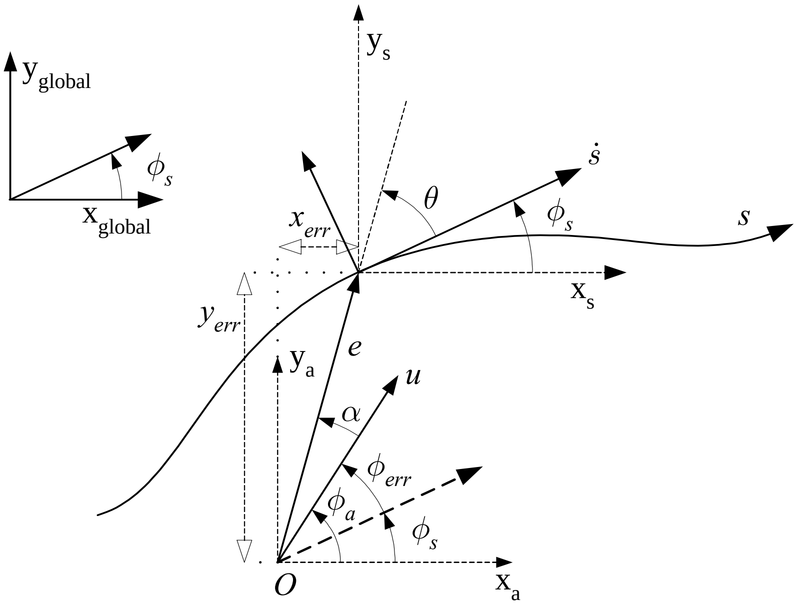


Figure 5: Path error coordinates.

The Cartesian path error coordinates, x_{err}, y_{err} and ϕ_{err}, may be developed according to the equations

$$\begin{aligned} x_{err} &= x_a - x_s \\ y_{err} &= y_a - y_s \\ \phi_{err} &= \phi_a - \phi_s \end{aligned} \tag{46}$$

where x_a, y_a and ϕ_a are the actual robot equivalent center point posture, and x_s, y_s and ϕ_s are the desired center point posture of s as shown in Figure 5. The polar error coordinates may then be found from the Cartesian error coordinates by the following equations

$$\begin{aligned} &e = \sqrt{x_{err}{}^2 + y_{err}{}^2} \\ &\theta = ATAN2(-y_{err}, -x_{err}) - \phi_s \\ &\alpha = \theta - \phi_{err}. \end{aligned} \tag{47}$$

From Figure 5 it can be seen that the polar state equations now become

$$\begin{aligned}\dot{e} &= -u\cos\alpha + \dot{s}\cos\theta \\ \dot{\theta} &= u\frac{\sin\alpha}{e} - \dot{s}\frac{\sin\theta}{e} \\ \dot{\alpha} &= u\frac{\sin\alpha}{e} - \dot{s}\frac{\sin\theta}{e} - uc.\end{aligned} \tag{48}$$

Again it is desired to use Lyapunov stability techniques to show the convergence of the path following system. For path following, the positive definite and radially unbounded candidate Lyapunov function

$$V = \frac{1}{2}k_1 e^2 + \frac{1}{2}\left(\alpha^2 + k_3\theta^2\right) \tag{49}$$

is chosen. In the point stabilization case, the state *e* was not included in the Lyapunov function, yet it was shown that it would converge to zero. In the path following case the logic used to show the convergence of *e* would not apply because of the $\dot{s}$ terms in (48), therefore all three states are included in the candidate Lyapunov function.

The time derivative of the candidate Lyapunov function becomes much more complex in the path following case because of the extra terms in the state equations

$$\begin{aligned}\dot{V} = k_1 e(-u\cos\alpha + \dot{s}\cos\theta) - \frac{\alpha}{e}(uce - u\sin\alpha + \dot{s}\sin\theta) \\ - \frac{k_3\theta}{e}(-u\sin\alpha + \dot{s}\sin\theta).\end{aligned} \tag{50}$$

If the control inputs *u* and *c* are chosen to be

$$
\begin{aligned}
u &= \frac{k_1 e}{\cos\alpha} + \frac{\dot{s}\cos\theta}{\cos\alpha} \\
c &= \frac{\sin\alpha}{e} + \frac{k_2\theta\sin\alpha}{e\alpha} + \frac{k_1 k_3 \alpha}{u_d} - \frac{\dot{s}\sin\theta}{u_d e} - \frac{k_2 \dot{s}\theta\sin\theta}{u_d e\alpha}
\end{aligned} \tag{51}
$$

then the time derivative of the candidate Lyapunov function reduces to

$$
-k_1^2 e^2 - k_1 k_3 \alpha^2 \le 0 \tag{52}
$$

which is negative semidefinite. Although this choice for the velocity and curvature control laws looks promising, it has two major flaws: the *u* term in (51) will become unbounded as α approaches $\pm\pi/2$, and the *c* term will become unbounded as α approaches zero. A saturation function may again be imposed on *u* to inhibit the control variable from becoming unbounded when α approaches $\pm\pi/2$. The unboundedness of the *c* control variable is not as easily overcome since this variable will become unbounded as α approaches zero, the value that the controller is trying to stabilize to.

This problem may be solved by using an approach similar to that suggested by Tayebi [20] for a related system. First, the polar representation of the robot kinematics shown in Figure 4 may be changed into a signed polar representation as shown in Figure 6 (A). By a signed polar representation it is meant that the polar error distance variable *e* is represented by both positive and negative values. The value of e is defined as positive in the right hand half plane and negative in the left hand half. The angle θ is also changed so that it is always measured from the abscissa and therefore is always contained in the set $\left[-\pi/2, \pi/2\right]$. The polar error coordinates are now found from the Cartesian error coordinates by

$$\begin{aligned} e &= \sqrt{{x_{err}}^2 + {y_{err}}^2} \\ \theta &= ATAN2(y_{err}, x_{err}) - \phi_s \end{aligned} \tag{53}$$

where the angle α is now defined using equation (46) in the Left Hand Plane (LHP) as

$$\begin{aligned} \alpha &= \theta + \phi_{err} \\ &= \theta + \phi_a - \phi_s \end{aligned} \tag{54}$$

where,

$$\dot{\alpha} = \dot{\theta} + \dot{\phi}_a - \dot{\phi}_s \tag{55}$$

and in the Right Hand Plane (RHP) as

$$\begin{aligned} \alpha &= \theta + \phi_{err} \\ &= \theta - \phi_a + \phi_s \end{aligned} \tag{56}$$

where,

$$\dot{\alpha} = \dot{\theta} - \dot{\phi}_a + \dot{\phi}_s. \tag{57}$$

An example of the signed polar representation for a posture located in the third quadrant is shown in Figure 6 (B). The path following state equations of the system using the signed polar approach become

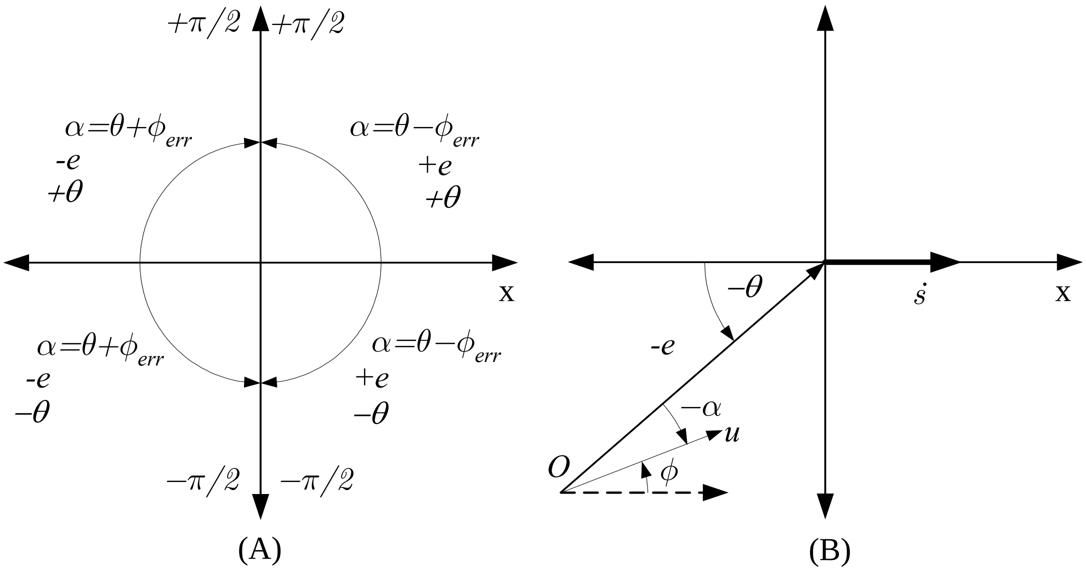


Figure 6: Signed polar representation of robot kinematics. (A) Polar variable definitions. (B) Example posture description.

$$\begin{aligned}
\dot{e} &= u\cos\alpha - \dot{s}\cos\theta \\
\dot{\theta} &= -u\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} \\
\dot{\alpha} &= -u\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} + \dot{\phi}_a - \dot{\phi}_s
\end{aligned} \tag{58}$$

for the LHP and

$$\begin{aligned}
\dot{e} &= u\cos\alpha - \dot{s}\cos\theta \\
\dot{\theta} &= -u\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} \\
\dot{\alpha} &= -u\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} - \dot{\phi}_a + \dot{\phi}_s
\end{aligned} \tag{59}$$

for the RHP.

A coordinate transformation is now introduced to place the state equations in a form more suitable to Lyapunov analysis. The control variables *c₁* and *c₂* are introduced where

$$\begin{aligned} c_1 &= u - \dot{s} \\ c_2 &= \dot{\phi}_a - \dot{\phi}_s \end{aligned} \tag{60}$$

Upon substitution of (60) into (58) and (59) the system state equations become

$$\text{LHP} \qquad \begin{aligned} \dot{e} &= c_1 \cos\alpha - \dot{s}\cos\theta + \dot{s}\cos\alpha \\ \dot{\theta} &= -\left(c_1 + \dot{s}\right)\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} \\ \dot{\alpha} &= -\left(c_1 + \dot{s}\right)\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} + c_2 \end{aligned} \tag{61}$$

$$\text{RHP} \qquad \begin{aligned} \dot{e} &= c_1 \cos\alpha - \dot{s}\cos\theta + \dot{s}\cos\alpha \\ \dot{\theta} &= -\left(c_1 + \dot{s}\right)\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} \\ \dot{\alpha} &= -\left(c_1 + \dot{s}\right)\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} - c_2. \end{aligned} \tag{62}$$

Now with the state equations in this form we can use the positive definite, radially unbounded quadratic candidate Lyapunov function

$$V = \frac{1}{2}\left(e^2 + \alpha^2 + k_3\theta^2\right). \tag{63}$$

The time derivative of (63) is

$$\text{LHP}\qquad \begin{aligned}\dot{V} &= ec_1\cos\alpha - e\dot{s}\cos\theta + e\dot{s}\cos\alpha - \frac{\alpha c_1\sin\alpha}{e} - \frac{\alpha\dot{s}\sin\alpha}{e} \\ &+ \frac{\alpha\dot{s}\sin\theta}{e} + \alpha c_2 - \frac{k_3 c_1\theta\sin\alpha}{e} - \frac{k_3\dot{s}\theta\sin\alpha}{e} + \frac{k_3\dot{s}\theta\sin\theta}{e}\end{aligned} \tag{64}$$

$$\text{RHP}\qquad \begin{aligned}\dot{V} &= ec_1\cos\alpha - e\dot{s}\cos\theta + e\dot{s}\cos\alpha - \frac{\alpha c_1\sin\alpha}{e} - \frac{\alpha\dot{s}\sin\alpha}{e} \\ &+ \frac{\alpha\dot{s}\sin\theta}{e} - \alpha c_2 - \frac{k_3 c_1\theta\sin\alpha}{e} - \frac{k_3\dot{s}\theta\sin\alpha}{e} + \frac{k_3\dot{s}\theta\sin\theta}{e}\end{aligned} \tag{65}$$

Although (64) and (65) look quite formidable, provided the control variables are chosen as

$$c_1 = -\frac{k_1 e}{\cos\alpha} + \frac{\dot{s}\cos\theta}{\cos\alpha} - \dot{s} \tag{66}$$

$$\text{LHP}\qquad c_2 = -k_2\alpha + \frac{k_3 c_1\theta\sin\alpha}{e\alpha} + \frac{k_3\dot{s}\theta\sin\alpha}{e\alpha} + \frac{c_1\sin\alpha}{e} + \frac{\dot{s}\sin\alpha}{e} - \frac{\dot{s}\sin\theta}{e} \tag{67}$$

$$\text{RHP}\qquad c_2 = k_2\alpha - \frac{k_3 c_1\theta\sin\alpha}{e\alpha} - \frac{k_3\dot{s}\theta\sin\alpha}{e\alpha} - \frac{c_1\sin\alpha}{e} - \frac{\dot{s}\sin\alpha}{e} + \frac{\dot{s}\sin\theta}{e} \tag{68}$$

the time derivative of (63) reduces to

$$\dot{V} = -e^2 k_1 - k_2\alpha^2 + \frac{k_3\theta\dot{s}\sin\theta}{e} \tag{69}$$

for the entire plane. Again, it is desired to show that (69) is negative definite for asymptotic stability. It is easily seen that the first two terms in (69) are negative definite, but the last term is inconclusive. The path velocity $\dot{s}$ may be chosen as

$$\dot{s} = \frac{\theta\left(\dot{s}_{des} - k_4 e\right)}{\sin\theta} \tag{70}$$

where $\dot{s}_{des}$ represents the desired velocity of the path, $\dot{s}$ is the actual velocity of the path, and k_4 is a positive constant. Hence, the desired robot posture *s,* and velocity $\dot{s}$ are now dependent on the system error states. Utilizing the definition in (70) the time derivative of the Lyapunov equation (69) now becomes

$$\dot{V} = -k_1 e^2 - k_2\alpha^2 - k_3 k_4\theta^2 + \frac{k_3\theta^2\dot{s}_{des}}{e}. \tag{71}$$

In the path following case the $\dot{s}_{des}$ term may be a function of time. Consequently, the system is no longer autonomous as in the point stabilization case. Hence, stability techniques for nonautonomous systems must be used.

If the control variable c_1 in (66) is substituted into (61) the state equation for *e* is obtained as $\dot{e} = -k_1 e$, which will decay exponentially to zero without crossing zero. This implies that the $sign(d) = sign(d(0)) \quad \forall \quad t \geq 0$. If the condition is imposed that in order to track a path in the forward direction $e(0) < 0$, and to track a path in reverse motion $e(0) > 0$, then $sign(\dot{s}_{des}) = -sign(d(0))$. It now may be determined that there is a β, such that

$$\dot{V}(e,\theta,\alpha) \le -k_1 e^2 - k_2 \alpha^2 - k_3 k_4 \theta^2 \le -\beta V(e,\theta,\alpha) \tag{72}$$

where β can be chosen as

$$2\inf(k_1, k_2, k_4) \tag{73}$$

where inf denotes the infimum, or the greatest lower bound. Hence, the origin of the path following system is exponentially stable. From (66) and (70) the limits

$$\begin{aligned} &\lim_{e,\theta,\alpha\to 0} c_1 = 0 \\ &\lim_{e,\theta\to 0} \dot{s} = \dot{s}_{des} \end{aligned} \tag{74}$$

may be ascertained.

In the path following case, the control variable c_2 may grow unbounded as e tends to zero. This may be overcome by letting the control variable c_1 be

$$\begin{aligned} &c_1 = -\frac{k_1 e}{\cos\alpha} + \frac{\dot{s}\cos\theta}{\cos\alpha} - \dot{s} \quad if \quad |e| > \varepsilon \\ &c_1 = 0 \qquad\qquad\qquad\qquad\quad if \quad |e| \le \varepsilon \end{aligned} \tag{75}$$

where ε is a positive constant that may be chosen based upon the particular robot system. Systems with large perturbations or drift will require a larger value of ε to ensure the control variable c_2 does not become unbounded.

The velocity and curvature of the robot may be found from the control variable c_1 and c_2 from the definition for c_1 and c_2 in (60)

$$\begin{aligned} u &= c_1 + \dot{s} \\ c &= \frac{c_2 + \dot{s}c_{path}}{u} \end{aligned} \tag{76}$$

where c_{path} is the curvature of the path at s. Thus it is necessary to know c_{path} along with x_s, y_s and ϕ_s at each s shown in Figure 6.

As in the point stabilization case nonideal initial conditions will require the extension of u and c in (76). The stability of the extension performed in the point stabilization case holds, and the state equations for the actual velocity and curvature may be taken as

$$\begin{aligned} \dot{u}_a &= \dot{u}_d + k_u\left(u_d - u_a\right) \\ \dot{c}_a &= \dot{c}_d + k_c\left(c_d - c_a\right) \end{aligned} \tag{77}$$

where u_d and c_d are the desired velocity and curvature output from the controller in (76) and u_a and c_a are the actual measured feedback velocity and curvature.

Point Stabilization Using the Signed Polar Approach

One benefit of the signed polar formulation is that it may be used for path following and point stabilization. In the point stabilization case, $\dot{s}_{des}(t) = 0,\ \dot{\phi}_s = 0$, and we set k_4=0. The system now reduces to the autonomous or time-invariant case. Given that $\dot{s}_{des}(t) = 0$, the time derivative of the Lyapunov function in (69) reduces to

$$\dot{V} = -e^2k_1 - k_2\alpha^2 \tag{78}$$

and it can be seen from LaSalle's theorem [30] that *e*, α, and $\dot{\alpha}$ tend to zero as time tends to infinity. The control variables c_1 and c_2 in (66) become

$$\begin{aligned} &c_1 = -\frac{k_1 e}{\cos\alpha} + \frac{\dot{s}\cos\theta}{\cos\alpha} - \dot{s} \\ &c_1 \to 0 \end{aligned} \tag{79}$$

LHP
$$\begin{aligned} &c_2 = -k_2\alpha + \frac{k_3 c_1 \theta \sin\alpha}{e\alpha} + \frac{k_3 \dot{s}\theta \sin\alpha}{e\alpha} + \frac{c_1 \sin\alpha}{e} + \frac{\dot{s}\sin\alpha}{e} - \frac{\dot{s}\sin\theta}{e} \\ &c_2 \to \frac{k_3 c_1 \theta}{e} \end{aligned} \tag{80}$$

RHP
$$\begin{aligned} &c_2 = k_2\alpha - \frac{k_3 c_1 \theta \sin\alpha}{e\alpha} - \frac{k_3 \dot{s}\theta \sin\alpha}{e\alpha} - \frac{c_1 \sin\alpha}{e} - \frac{\dot{s}\sin\alpha}{e} + \frac{\dot{s}\sin\theta}{e} \\ &c_2 \to -\frac{k_3 c_1 \theta}{e} \end{aligned} \tag{81}$$

and utilizing the fact that the state equation for α in (61) is,

LHP
$$\begin{aligned} &\dot{\alpha} = -\left(c_1 + \dot{s}\right)\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} + c_2 \\ &c_2 \to c_1 \frac{\sin\alpha}{e} \end{aligned} \tag{82}$$

RHP
$$\begin{aligned} &\dot{\alpha} = -\left(c_1 + \dot{s}\right)\frac{\sin\alpha}{e} + \dot{s}\frac{\sin\theta}{e} - c_2 \\ &c_2 \to -c_1 \frac{\sin\alpha}{e} \end{aligned} \tag{83}$$

it may be seen that

$$\text{LHP} \qquad c_2 \to c_1 \frac{\sin\alpha}{e} \to \frac{c_1 k_3 \theta}{e} \tag{84}$$

$$\text{RHP} \qquad c_2 \to -c_1 \frac{\sin\alpha}{e} \to -\frac{c_1 k_3 \theta}{e} \tag{85}$$

and therefore $k_3\theta \to \sin\theta \to 0$ as $t \to \infty$. Thus, θ tends asymptotically towards zero.

It is now desired to show the boundedness and convergence of the control variable c_2. From (84) it may be seen that the boundedness and convergence of c_2 are dependent on the ratios $\frac{\theta}{e}$ and $\frac{\alpha}{e}$. Again, we know from substituting the control variable c_1 in (66) into (61) that the state equation for e is obtained as $\dot{e} = -k_1 e$, which decays exponentially to zero. Hence, the convergence rates of θ and α must be greater than that of e to guarantee the boundedness of $\frac{\theta}{e}$ and $\frac{\alpha}{e}$.

The rates of convergence of the system states may be studied by linearizing (using small angle assumptions) the system state equations in (61) which become

$$\begin{aligned} \dot{e} &= -k_1 e \\ \dot{\theta} &= k_1 \alpha \\ \dot{\alpha} &= -k_2 \alpha - k_1 k_3 \theta \end{aligned} \tag{86}$$

for the entire plane. From the linearized state equation for *e* in (86), it can be determined that *e* converges to zero as the exponential of $-k_1 t$. Therefore, if the real part of the dominant pole of the subsystem

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & k_1 \\ -k_1 k_3 & -k_2 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \end{bmatrix} \tag{87}$$

is $-\lambda$, θ and α will converge to zero at a rate equal or greater than $\exp(-\lambda t)$. If the values for the gains k_1, k_2, k_3, are chosen as

$$\begin{aligned} & k_1 > 0, k_3 > 0 \; and \; 2k_1\sqrt{k_3} \le k_2 < k_1(1+k_3), \\ & or \\ & k_1 > 0, k_3 > 1 \; and \; 2k_1 < k_2 < 2k_1\sqrt{k_3} \end{aligned} \tag{88}$$

then the values of θ and α will converge faster than *e*, and the control variable c_2 will asymptotically converge towards zero.

The velocity and curvature may be found from (76) as in the path following case. The extension for nonideal initial conditions performed in the previous cases again holds, and the state equations for the actual velocity and curvature may be taken as

$$\begin{aligned} \dot{u}_a &= \dot{u}_d + k_u\left(u_d - u_a\right) \\ \dot{c}_a &= \dot{c}_d + k_c\left(c_d - c_a\right) \end{aligned} \tag{89}$$

where u_d and c_d are the desired velocity and curvature output from the controller in (76) and u_a and c_a are the actual measured feedback velocity and curvature.

CHAPTER V

EXPERIMENTAL PLATFORM

The point stabilization and path following controllers developed in Chapter IV were tested on the Partially Compliant Test Robot at the University of Utah, Figure 7. The Partially Compliant Test Robot (PCTR) consists of two differentially driven axle modules connected by a flexible frame module. The PCTR is constructed of commonly available components and illustrates how the CFR concept lends itself to simple, easy to manufacture designs [17]. The axles are constructed of commercially available structural aluminum extrusions. The flexible frame module connecting the axles is a single piece of flat spring steel, which is bolted to each axle forming a clamped joint. Each of the axles has a machined bearing block with two miniature ball bearings to support the wheel assemblies. A 24-volt dc gear motor independently drives each of the wheels with wheel positional feedback being provided by optical encoders (Figure 8). Power can be

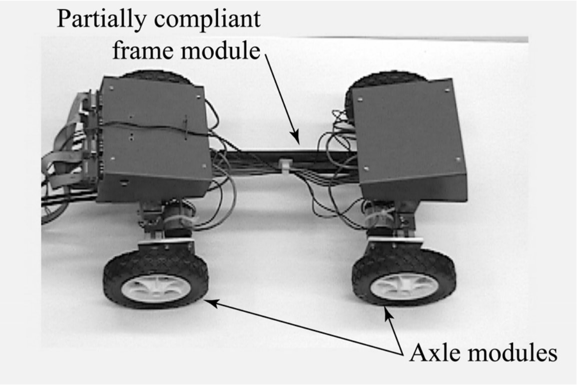


Figure 7: University of Utah partially compliant test robot

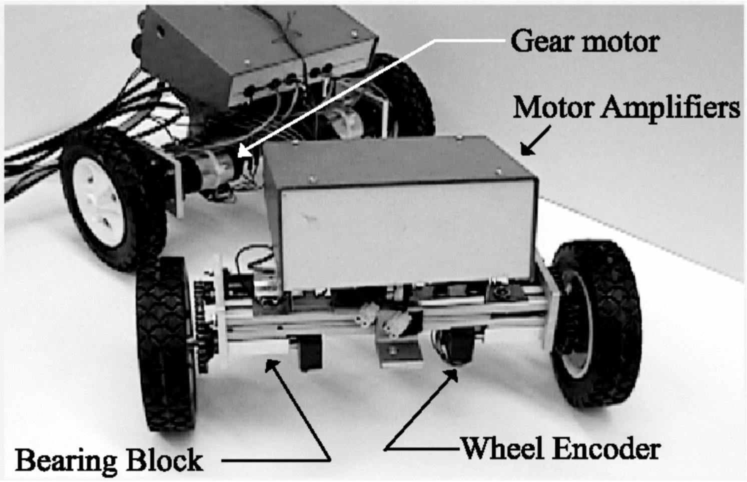


Figure 8: PCTR configuration and components.

supplied to the PCTR from on-board batteries, or a stationary power supply. For all of the experiments in this work, the stationary power supply was used. For more details on PCTR design and construction see [17]. A table of PCTR components and parameters may be found in Appendix B.

Control of the PCTR is performed using a dSPACE 1103 digital signal processor (DSP). The control algorithms developed in Chapter IV were created in Simulink® and compiled and loaded onto the DSP using Real Time Workshop®. Experiments were controlled and data recorded using ControlDesk®. Commands are sent to each of the wheels as voltage signals from four digital to analog outputs on the DSP. These voltage control signals are amplified by servo amplifiers mounted on the aluminum structural elements. Encoder feedback is read as digital signals on the DSP. For all of the experiments, in this study, the DSP ran at a speed of 1 kHz. At this speed, the digital implementation can be considered analog, because the control frequency is significantly higher than the system dynamic frequencies, which are on the order of a few Hz.

System Characterization

Before the control methods developed in Chapter IV may be implemented, the PCTR robot system must be characterized and low-level wheel controllers developed to ensure control inputs and actual wheel response match as closely as possible.

The low-level wheel controller system model can be represented by the system diagram shown in Figure 9. The $\omega_{Desired}$ signal is calculated from the current robot posture by the control algorithms developed in Chapter IV. This signal is compared with the ω_{Sensed} signal calculated from the encoder feedback. The error between these two signals is the input to the controller G_C. The control output is then sent from the DSP to the robot represented by G_P. The absolute wheel position, Θ, is sensed by the encoders and measured by the DSP. This signal is filtered, to smooth out the digital encoder signal, and differentiated to determine the measured wheel angular velocity, ω_{Sensed}, to be compared with the desired wheel angular velocity $\omega_{Desired}$.

The first step in developing the low-level wheel controller is to characterize the robot plant $G_{P.}$ Many of the parameters necessary to find a transfer function characterization of G_P using traditional system modeling techniques would be very difficult to determine for the PCTR. Specifically, determining the coulomb and viscous friction and the rotational moment of inertia for each wheel by calculation would be very

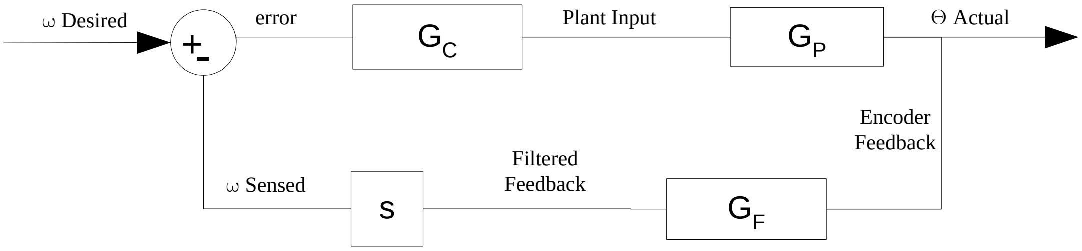


Figure 9: Low-level wheel controller system diagram

difficult. Many assumptions would have to be made that would likely lead to considerable error. Because of this, it was decided that the best method for determining the plant transfer function would be through experimental means.

If the system in Figure 9 is reconfigured to match a desired wheel angular position response $\Theta_{Desired}$, instead of a desired wheel angular velocity $\omega_{Desired}$, then the system diagram is modeled as shown in Figure 10. The filter, G_F, is no longer necessary for position control because the feedback signal is not differentiated and a smooth signal is thus not as critical. The controller gain, K_C, is simply a proportional gain imposed by the controller. All other system gains are included in the plant transfer function $G_{P.}$ These include the amplifier gain, motor gain, gear ratio gain, etc. This method will prove to be convenient because, like other previously mentioned system parameters, these gains are difficult to quantify and vary for each wheel.

The closed loop transfer function for the system shown in Figure 10 is

$$\frac{\theta_{Actual}}{\theta_{Desired}} = \frac{K_C G_P}{1 + K_C G_P}. \tag{90}$$

It is assumed that the plant, G_P, is of the following open loop form

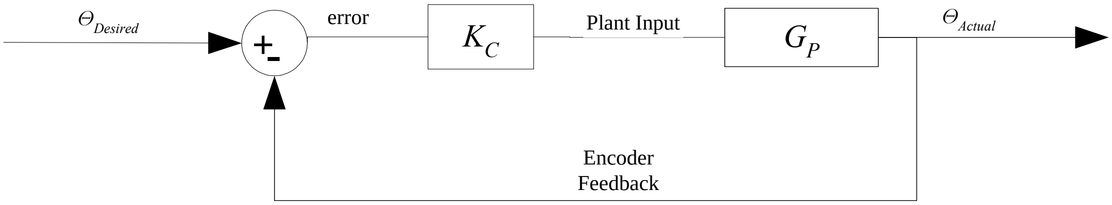


Figure 10: Plant system diagram.

$$G_P = \frac{a_0}{b_2 s^2 + b_1 s} \tag{91}$$

which matches the form of a typical damped mass motor system. Substituting (91) into (90) and simplifying reveals the following closed loop system transfer function

$$\frac{\dfrac{K_C a_0}{b_2}}{s^2 + \dfrac{b_1}{b_2} s + \dfrac{K_C a_0}{b_2}}. \tag{92}$$

Inspection of equation (92) shows that it closely resembles the response of a second order system. Setting (92) equal to a second order response, the following relations can be developed

$$\begin{gathered} \frac{\dfrac{K_C a_0}{b_2}}{s^2 + \dfrac{b_1}{b_2} s + \dfrac{K_C a_0}{b_2}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ 2\zeta\omega_n = \frac{b_1}{b_2} \\ \frac{\omega_n^2}{K_C} = \frac{a_0}{b_2}. \end{gathered} \tag{93}$$

Rearranging (91) into the following form

$$G_P = \frac{\dfrac{a_0}{b_2}}{s^2 + \dfrac{b_1}{b_2} s} \tag{94}$$

it can be seen that if K_C, c, and ω_n are known, then the open loop transfer function G_P may be found.

The value of K_C may be chosen so that the response of the system is under damped. The values ς and ω_n may then easily be found from the system response using the familiar second order response equations for percent overshoot, *P.O.*, and time to peak, T_p.

$$\begin{aligned} P.O. &= \frac{Max - F.V.}{F.V.}100 \\ &= 100e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \\ T_p &= \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \end{aligned} \tag{95}$$

Each of the PCTR wheel controllers was given a step input and the percent overshoot and time to peak of the response was used to find ς and ω_n, with $K_C = 20$. These responses can be seen in Figure 11 compared with the second order system model response generated using the ς and ω_n that were found. The time to peak, percent overshoot, ς, ω_n, and plant transfer function for each of the wheels is displayed in Table 2. It can be seen from Figure 11 that the second order system response matches each of the wheels measured responses quite well.

As an added reassurance that these plant transfer functions were accurate, an HP Dynamic signal analyzer was utilized to find the roots of the system. The signal analyzer provided a swept sine input from 0.25 Hz to 5 Hz for each of the four wheels. One of the wheel encoders was used for position feedback to the DSP, and the DSP then provided a

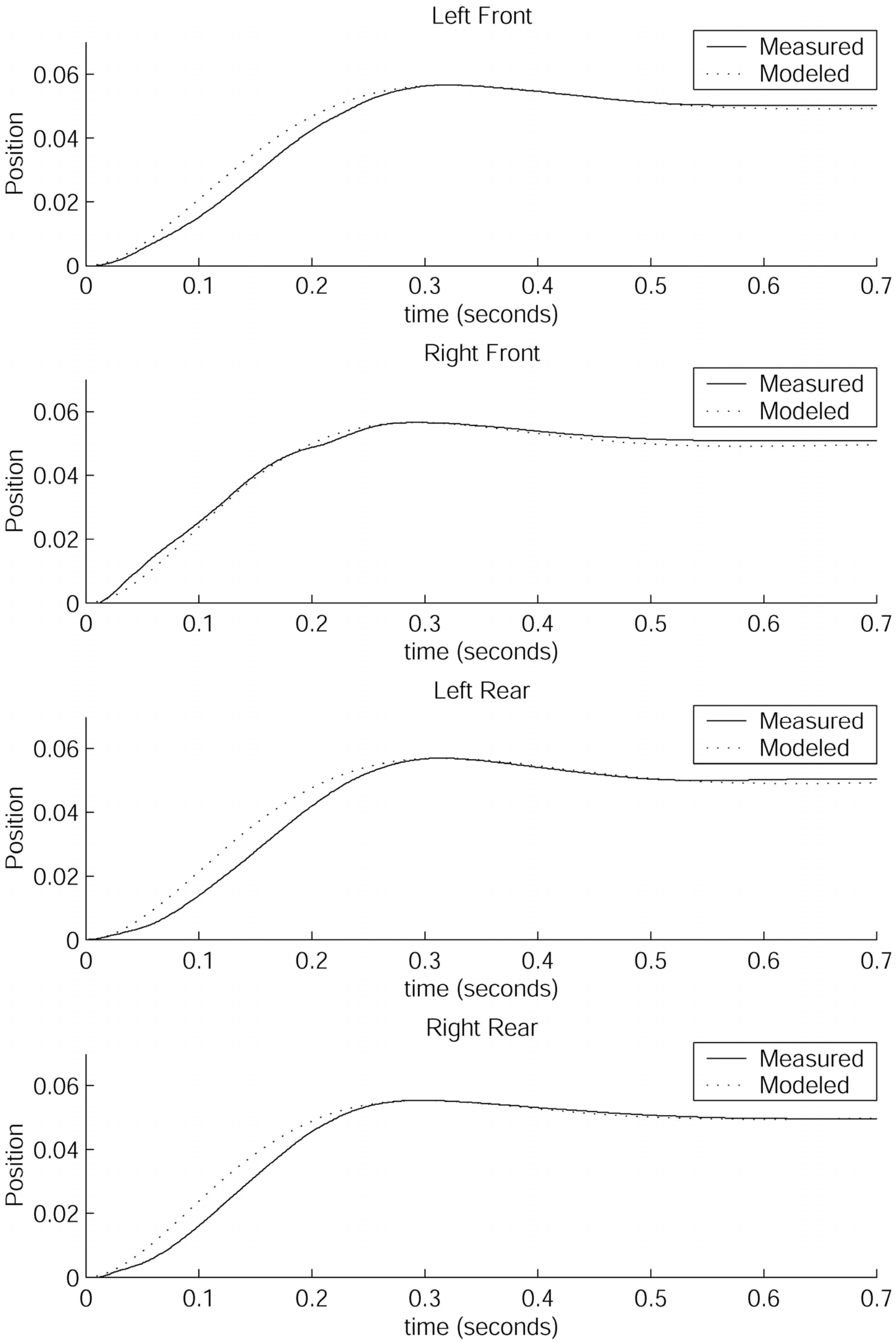


Figure 11: Step response for each wheel of the PCTR robot. Actual measured responses are compared with 2nd order model response.

Table 2: Response values of PCTR robot to a step input and individual wheel plant transfer functions. K_C=20

Wheel	T_p, sec	$P.O.$	ζ	ω_n, rad/sec	G_P
Left Front	0.322	13.2%	.542	11.61	$\frac{6.7396}{s^2+12.5852s}$
Right Front	0.293	13.2%	.542	12.76	$\frac{8.1409}{s^2+13.8318s}$
Left Rear	0.315	13.8%	.533	11.79	$\frac{6.9502}{s^2+12.5681s}$
Right Rear	0.298	10.6%	.581	12.96	$\frac{8.3981}{s^2+15.0595s}$

voltage signal to the HP Dynamic Signal Analyzer. The DSP was necessary to convert the digital encoder feedback signal to an analog feedback signal that could be input into the analyzer. The HP Dynamic Signal Analyzer created a Bode Plot from the robot response and used a curve fit to determine the roots of the system. Several tests were performed because of difficulty in obtaining consistent results. It is believed that this is due to differences in each of the wheel systems that caused the robot to rotate in circles and deflect the frame during the tests. The average of the tests showed roots at 0 and -11.8. This compared favorably from the roots found using the step input of 0 and -12.6 to -15.1. Because of the difficulty in obtaining consistent results from the signal analyzer, the roots obtained from the step input were used.

Filter Design

The next step in developing the low-level wheel controller in Figure 9 is to find a suitable transfer function for the filter, G_F. The filter is necessary to smooth out the wheel angular position output from each of the optical encoders. This output, upon magnification, resembles a staircase because the encoders provide a signal that is incremented by 500 counts per revolution. Each count of the encoder represents a step of approximately 0.7 degrees, or approximately 1 mm translation. The signal is differentiated to find the wheel angular velocity, and this differentiation is unbounded due to the steps in the signal. Thus, a filter is necessary to smooth the signal before it is differentiated.

A second order filter was used, with $\varsigma = 1$ for convenience, and the cutoff frequency ω_f chosen to attenuate the signal at the desired range. The encoder feedback steps are more pronounced as velocity decreases because there are less revolutions of the wheel, and correspondingly less encoder steps per unit of time. Hence, it is desired to have the cutoff frequency as low as possible to combat this effect at low velocities. The cutoff frequency must not be chosen too low, or the system delay caused by the filter will have a significant impairment on controller performance. Consequently, it becomes a balancing act to have the cutoff frequency low enough to attenuate the encoder signal jumps at low speeds without causing too significant of a controller delay.

In the point stabilization case, encoder signal attenuation at low speeds proved to be the most critical. Tests were run on the PCTR and it was observed that vibrations seemed to be induced, by the encoder feedback signal, as the robot decelerates as it nears the origin beginning at a velocity of approximately 0.03 m/s. Consequently the target velocity for attenuation was chosen to be 0.03 m/s. Using the wheel radius and encoder counts per revolution this velocity can be converted by the equation

$$\begin{aligned} \omega &= \frac{v}{r_w} = \frac{.03m/s}{.073m} \\ f &= \frac{\omega}{2\pi} 500 counts/rev = 32.7Hz. \end{aligned} \tag{96}$$

into a target frequency of 32.7 Hz. It was chosen to have a cutoff frequency one tenth of the target frequency so that significant attenuation would be achieved at the target. This corresponds to 3.27 Hz, or an ω_f of approximately 20.5 rad/sec. Using ω_f of 20.5 rad/sec in the second order filter

$$\frac{\omega_f{}^2}{s^2 + 2\zeta\,\omega_f s + \omega_f{}^2} = \frac{20.5^2}{s^2 + 2\left(20.5\right)s + 20.5^2} \tag{97}$$

it can be seen in Figure 12 that the filter has a drop off of 40 dB/decade after the cutoff frequency, and the attenuation at the target frequency of 32.7 Hz is –40 dB. This was found to be acceptable on the PCTR robot system for point stabilization.

For the path following case, delay caused by the filter proved to be more critical than encoder signal smoothness. Generally the desired path velocity will be much faster than the 0.0 to 0.03 m/s critical low-end velocity range experienced in the point stabilization case. Thus, the robot will generally not be traveling inside the critical low-end range and an extremely low filter cutoff frequency is not necessary. The delay caused by an extremely low filter cutoff frequency also proved to cause instability as the path error approached zero. Therefore it was desirable to use a much higher cutoff frequency than for the point stabilization case. It was found to be quite effective to use the desired path velocity as the target attenuation velocity. The filter cutoff frequency ω_f may then be found using the same methodology as for the point stabilization case.

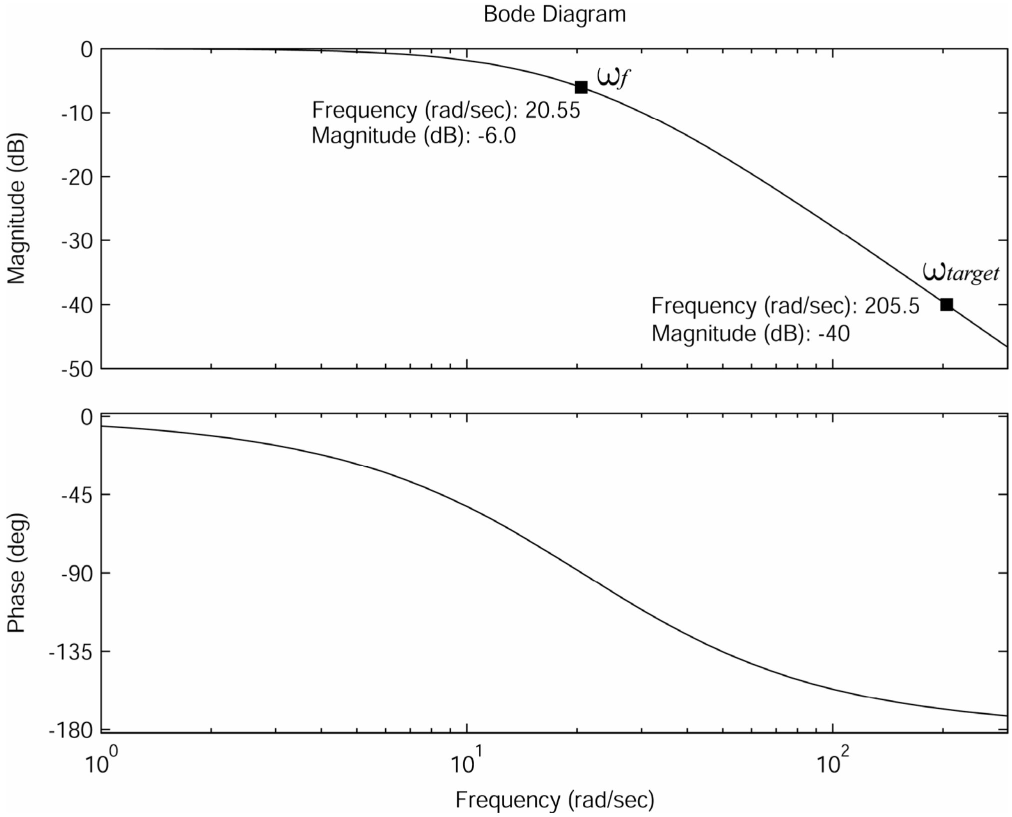


Figure 12: Filter Bode diagram. Shows attenuation at cutoff frequency ω_f, and the target frequency.

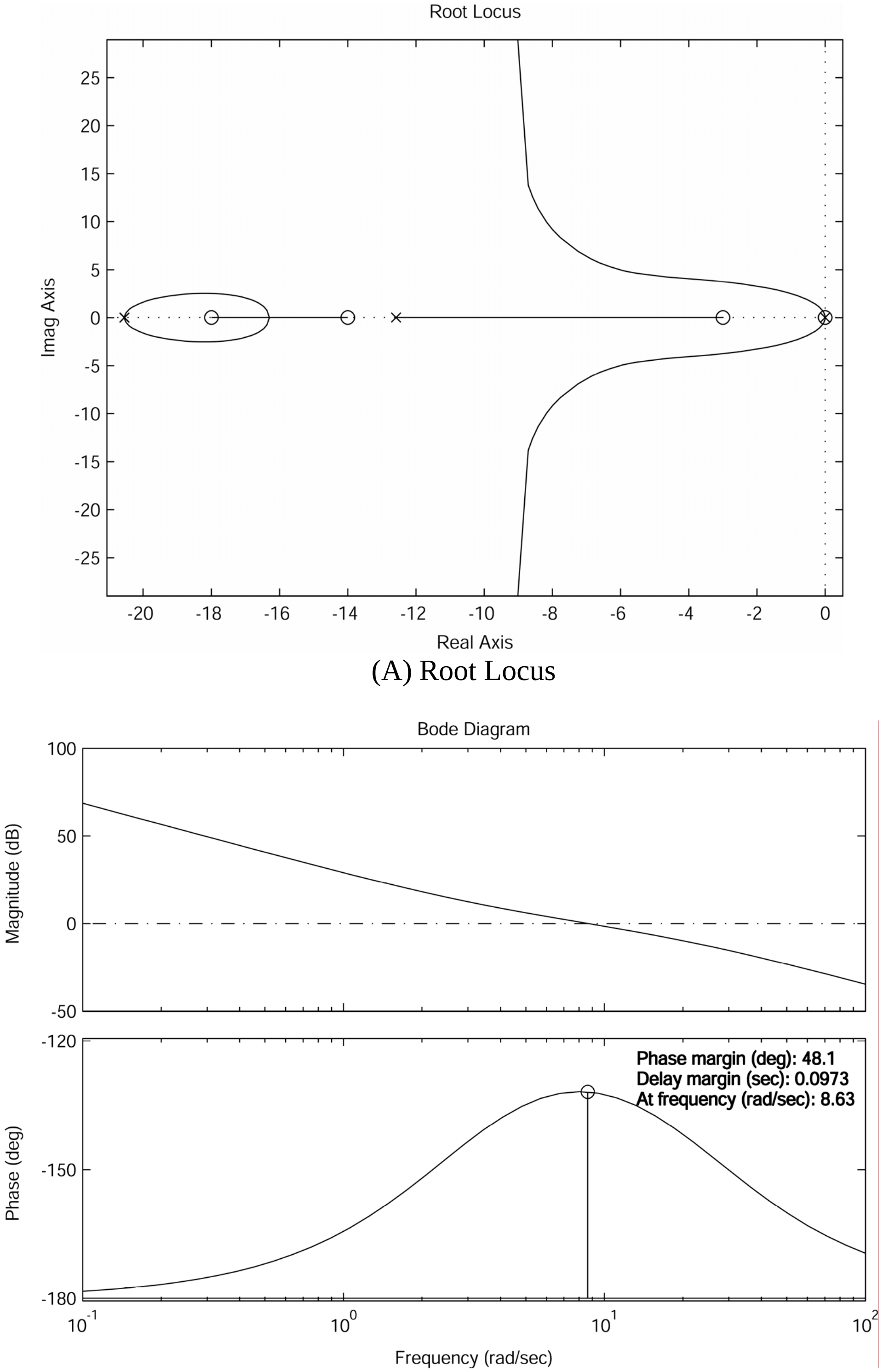
Controller Design

The final step in designing the low-level wheel controller is developing a controller, *G_C*, which will give the system in Figure 9 the desired response. Ideally, it would be desired to have the system type as high as possible. The input to the wheels from the path generators developed in Chapter IV will be similar to a ramp or parabola type input. Hence, a type one system would have at a minimum some constant error and for more parabolic inputs a constantly growing error. A type two system would have at a minimum zero error, and at a maximum a constant error. Therefore it is desired if possible to have a type two system, or in other words two free integrators.

The system transfer function G_P and the filter transfer function G_F were input into the Matlab® SISO design tool to find an appropriate controller for the system. The G_F used for the filter had a cutoff frequency optimized for the point stabilization case to ensure a conservative design. The plant by itself has a free integrator, but this is canceled out by the differentiation in the feedback loop. It was thus attempted first to find a controller with two free integrators to make the closed loop system type two. It was also desired to have the percent overshoot, and settling time as low as possible while maintaining a phase margin of at least 45° to maintain the system dynamic performance given uncertainty [33].

It was found that the best response obtained from a controller with two free integrators was with two poles at 0, and zeroes at –2.5, -14.1, and –17.8. The root locus of the system may be seen in Figure 13 (A). It can be seen from the root locus that there is a vertical asymptote at approximately –9 on the real axis. This asymptote draws the roots from the real axis to increasing imaginary values as gain increases. Because of this, as the gain is increased to make the roots more negative and decrease the settling time, they also have a larger imaginary component increasing the percent overshoot. The best response obtained, with the poles and zeroes above, was a settling time of approximately 0.74 seconds, and a percent overshoot of approximately 35%. The Bode diagram of the system with this controller can be seen in in Figure 13 (B) and the step response of the system may be seen in Figure 14. It was determined that the percent overshoot of this controller is much too high and could lead to system vibrations. The settling time of the response is also much too slow and deemed not acceptable.

A controller with just one free integrator was studied next. Again using the SISO



(A) Root Locus

(B) Bode Diagram

Figure 13: System with two controller poles at 0 and zeroes at –2.5, -14.1, -17.8

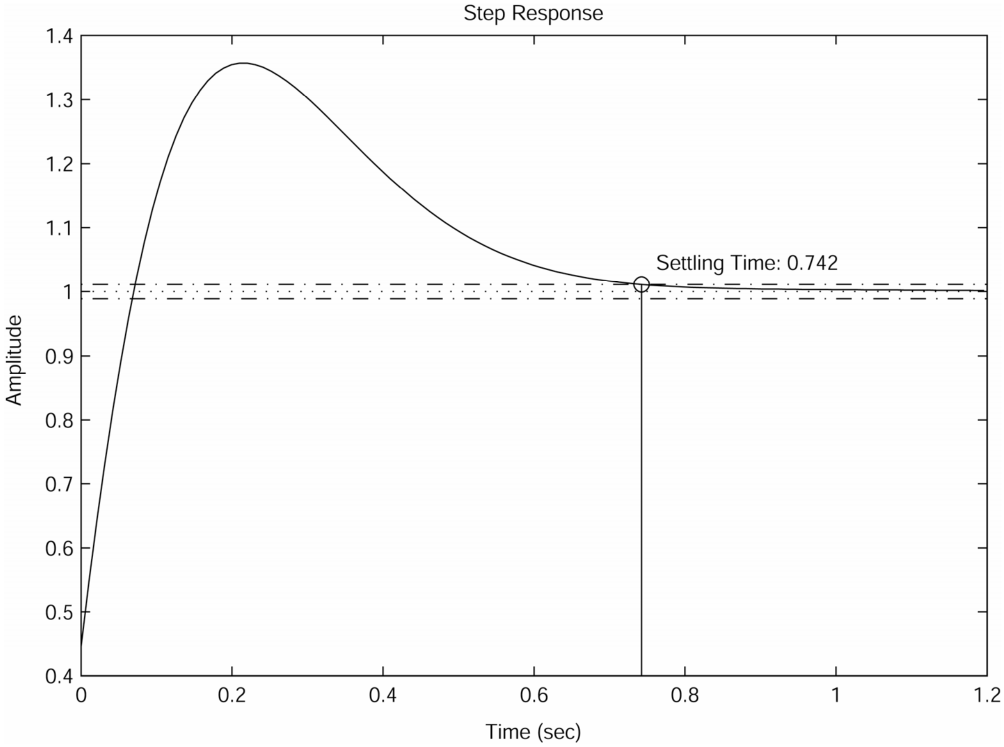


Figure 14: Step response with two controller poles at 0 and zeroes at -2.5, -14.1, -17.8.

Tool, it was determined that the best response was obtained with a single pole at zero and a zero at –11. The Root locus of this response can be seen in Figure 15. This root locus also has asymptotes at ±60° leading the poles to higher imaginary values as gain is increased, but with a proper choice of the gain K_C the percent overshoot is approximately 5% and the settling time is 0.329 seconds. This response is much better than the controller with two free integrators and was deemed acceptable.

It was decided that the settling time and percent overshoot benefits of this response outweigh the steady state error characteristics of the controller with two free integrators. Thus, a PI controller with a root at zero and pole at –11 was used. This leaves the system as a type one system. The value of K_C was chosen so that each of the wheels had a similar time response and overshoot. The gain K_C was chosen as 95.7 for

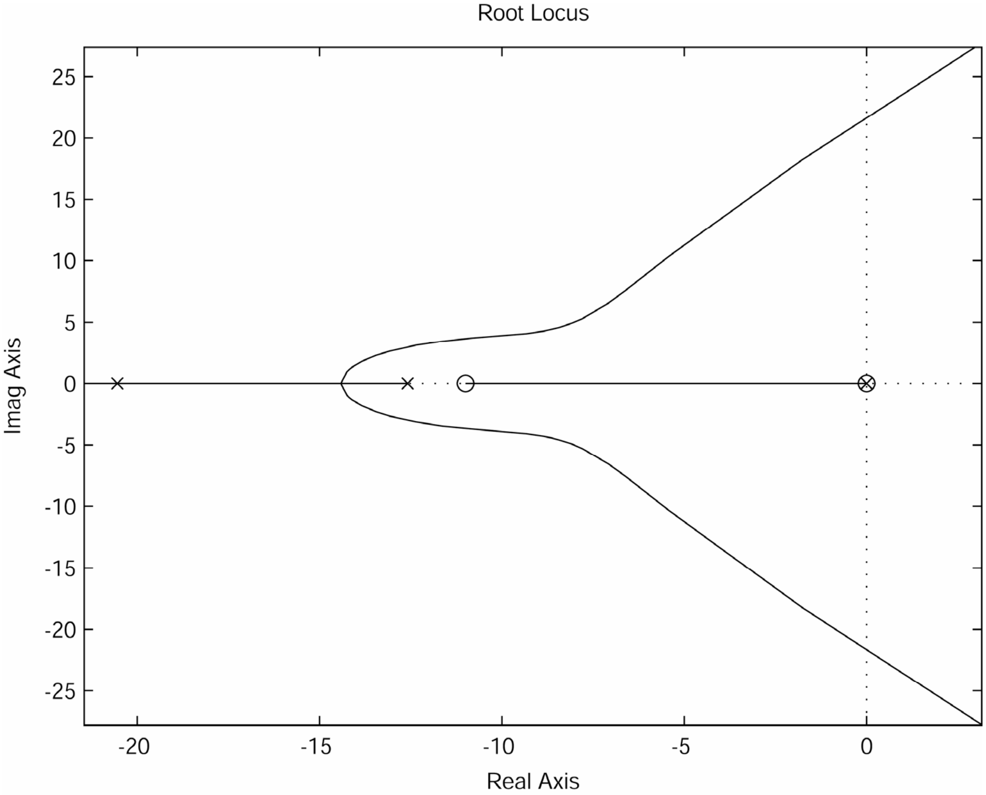
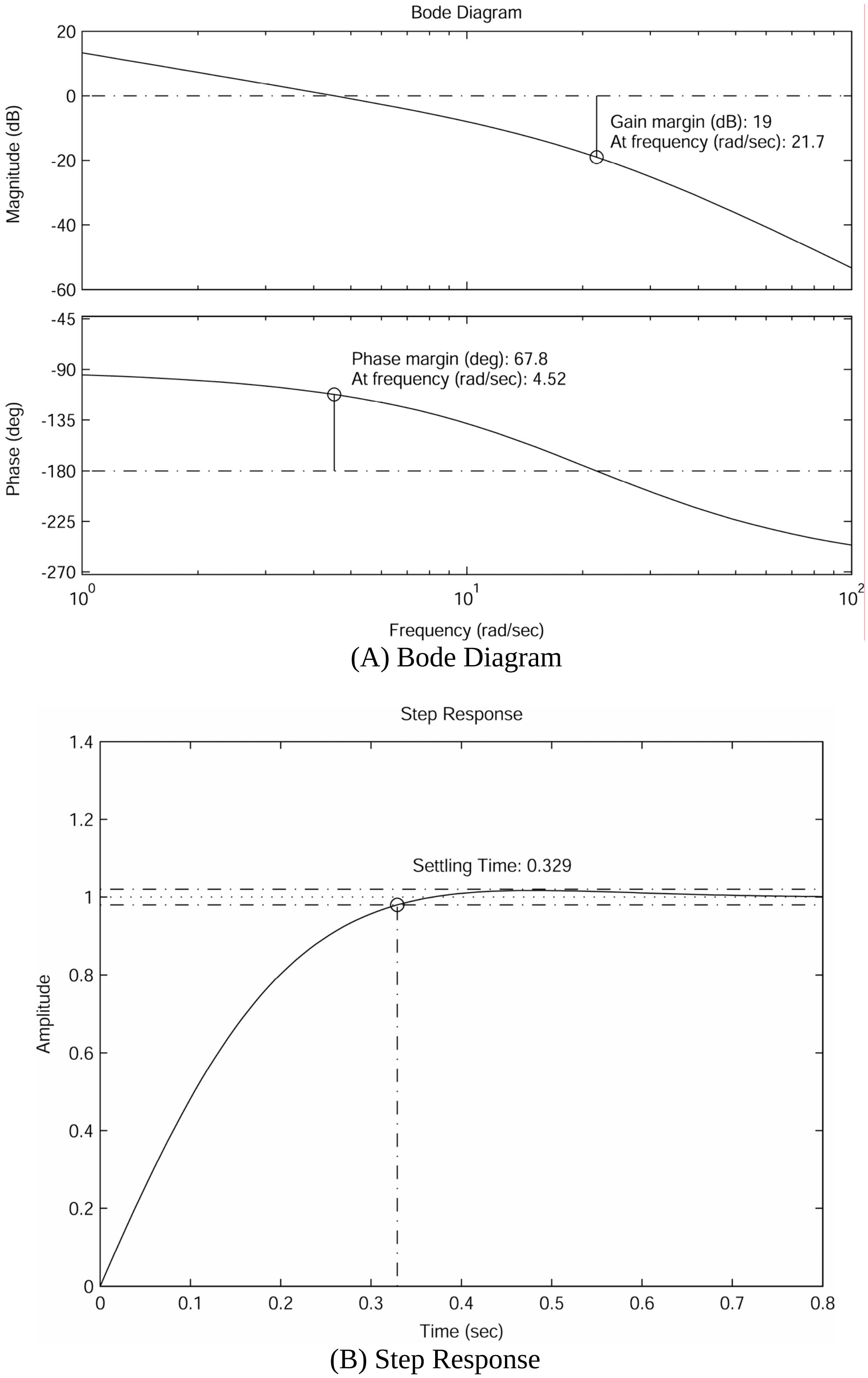


Figure 15: System root locus with controller pole at 0 and zero at -11.

the left front wheel, 88.6 for the right front, 93.0 for the left rear, and 95.2 for the right rear wheel. The Bode diagram of the system with this controller can be seen in Figure 16 (A), and the step response of the system may be seen in Figure 16 (B).

In the path following case, as mentioned above, the filter cutoff frequency ω_f may be chosen much higher than in the point stabilization case. This greatly reduces the amount of delay induced in the system by the filter, and allows higher controller gains to be chosen resulting in improved time responses of the system. Hence, in the path following case the controller gain should be optimized for the expected range of desired path velocities

The following linear equations for K_C may be used to find an appropriate gain



(A) Bode Diagram

(B) Step Response

Figure 16: System with controller pole at 0 and zero at –11, K_C of 95.7.

for a given desired path velocity assuming ω_f has been chosen for the same desired path velocity.

$$\begin{aligned} K_{C_{LF}} &= 1683\dot{s}_{des} + 45.2 \\ K_{C_{RF}} &= 2076\dot{s}_{des} + 26.2 \\ K_{C_{LR}} &= 1632\dot{s}_{des} + 44.0 \\ K_{C_{RR}} &= 2581\dot{s}_{des} + 17.7 \end{aligned} \tag{98}$$

These equations were developed by fitting a linear curve to the optimum gain at a 0.03 m/s target velocity and a 0.3 m/s target velocity. The controller was implemented in Simulink® using the equation with the following form

$$K = \frac{K_C(s+11)}{s}. \tag{99}$$

CHAPTER VI

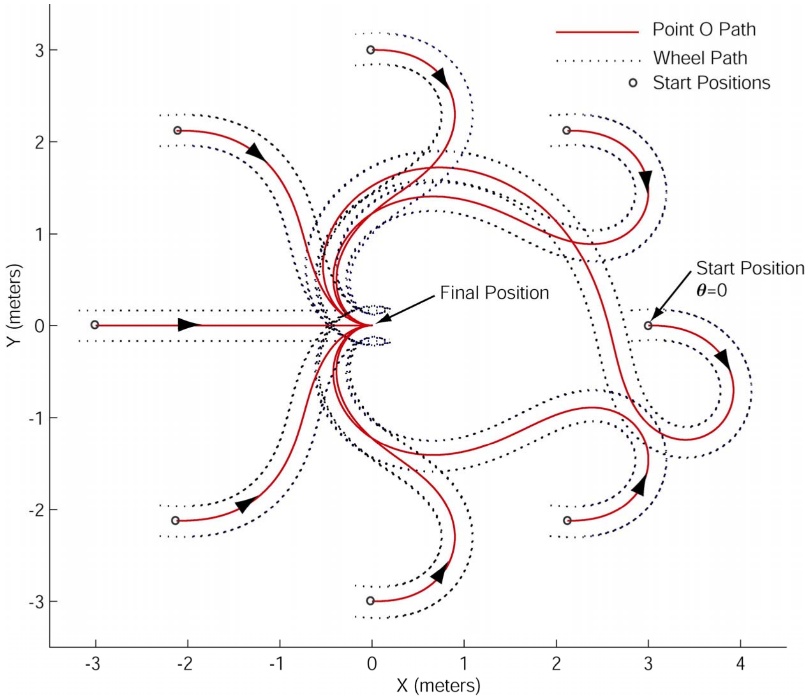
SIMULATION AND EXPERIMENTAL RESULTS

The validity of the point stabilization and path following controllers developed in Chapter IV was tested in simulation and experimentally on the PCTR robot. The control algorithms were realized using Simulink$^{®}$ models, which could be run in simulation or compiled and loaded onto the dSPACE$^{®}$ DSP.

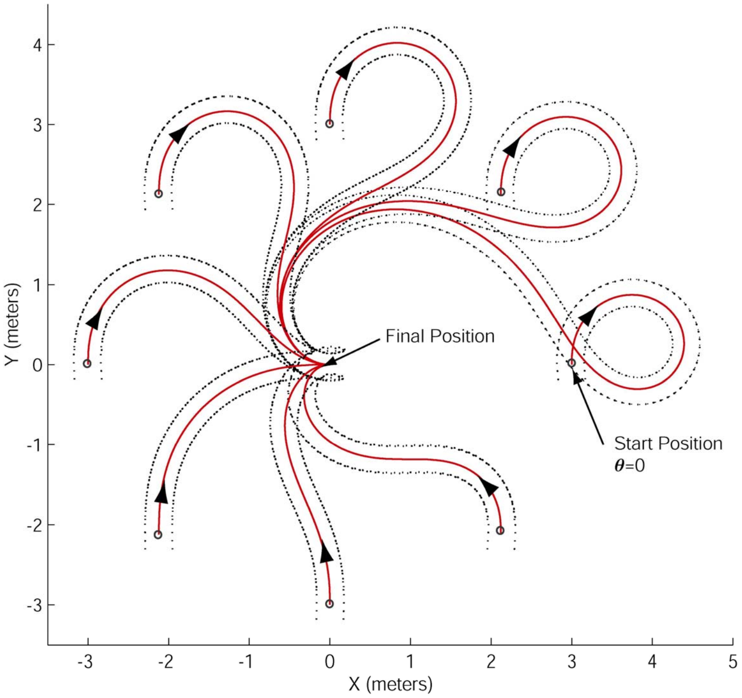
Indiveri Style Point Stabilization Simulation

The Indiveri style point stabilization controller was tested in simulation for several different initial conditions to evaluate its performance in the parking problem. As is shown Figure 17 and Figure 18, the controller does indeed generate smooth natural looking paths of bounded curvature. The paths proceed in forward motion, as required, and are free of cusps.

The extension of the controller dynamics performs well as is shown in Figure 19 (A). The curvature and velocity of the simulated robot converge well to the desired curvature and velocity output by the controller, and the entire system converges to the origin. The polar states also converge well to zero as shown in Figure 19 (B). The only item of concern is shown on the right in Figure 19 (A), where it is shown that the curvature control variable breaks down and becomes unstable as time gets large. It is believed this is due to machine precision error as the polar states become very small.

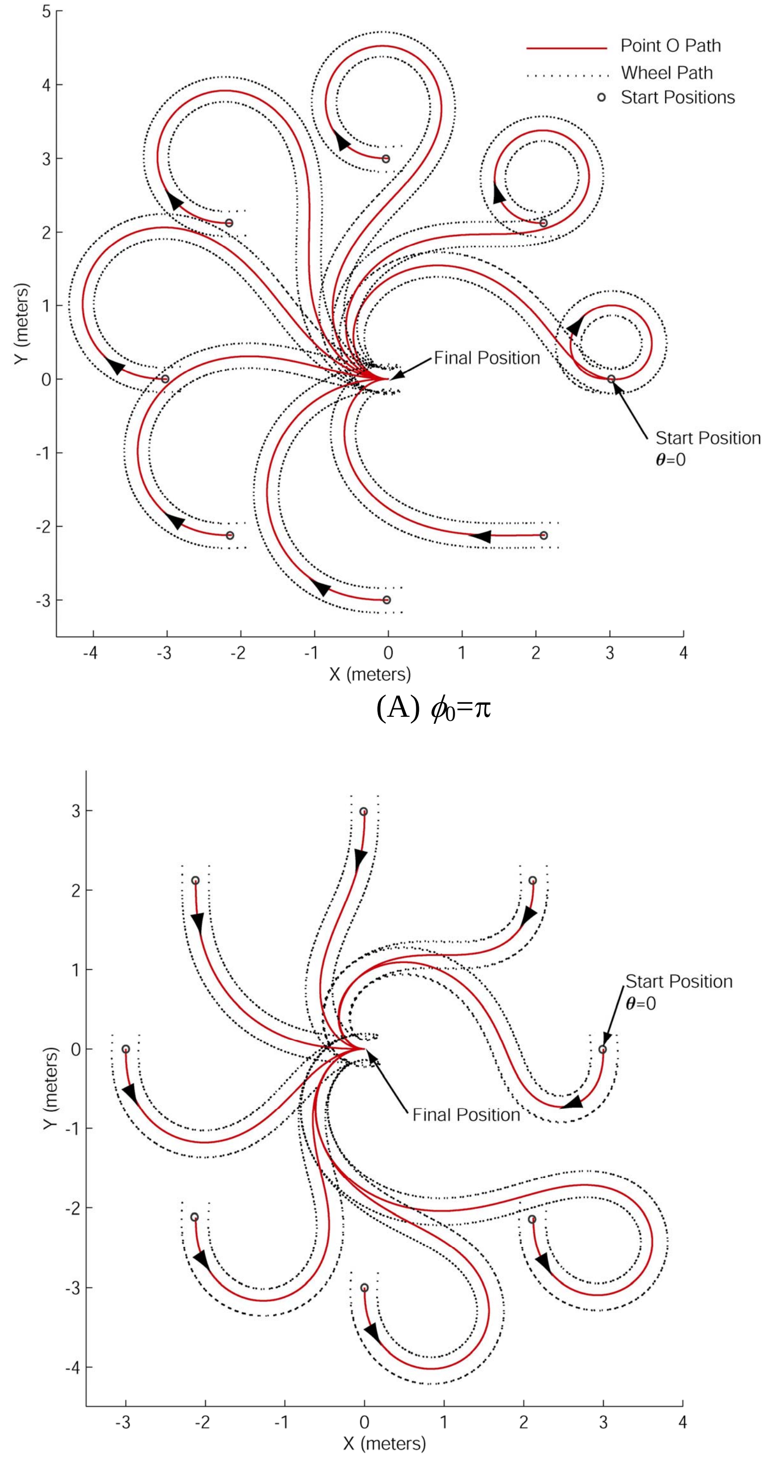


(A) ϕ_0=0



(B) ϕ_0=π/2

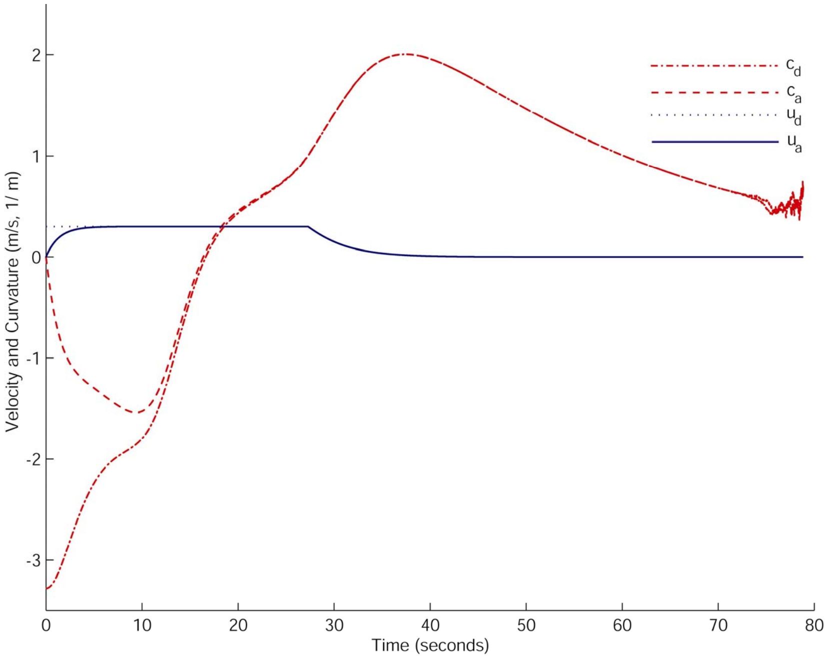
Figure 17: Simulation of CFR Indiveri style point stabilization controller paths. Initial conditions e_0=3 meters, and k_1=0.3, k_2=2, k_3=2.9, k_u=0.75, k_c=0.25, $\overline{u}$ =0.3.



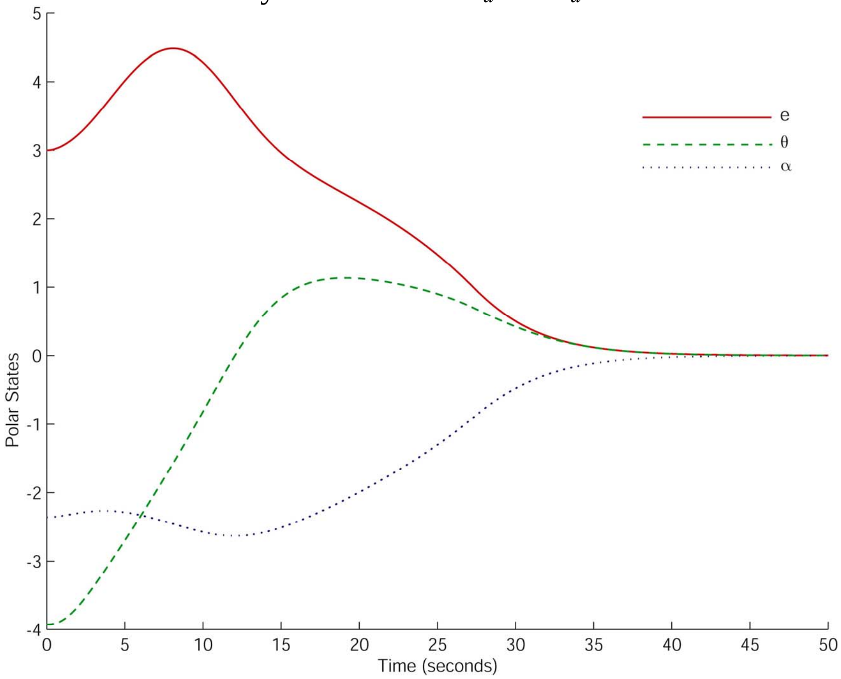
(A) $\phi_0=\pi$

(B) $\phi_0=3\pi/2$

Figure 18: Simulation of CFR Indiveri style point stabilization controller paths. Initial conditions e_0=3 meters, and k_1=0.3, k_2=2, k_3=2.9, k_u=0.75, k_c=0.25, $\overline{u}$ =0.3.



(A) Desired velocity and curvature u_d and c_d compared with actual velocity and curvature u_a and c_a



(B) Polar states e, θ, and α.

Figure 19: Simulation of CFR Indiveri style point stabilization controller control variables and states. Initial conditions e_0=3 meters and k_1=0.3, k_2=2, k_3=2.9, k_u=0.75, k_c=0.25, $\overline{u}$ =0.3, for θ_0=π/4, ϕ_0= -π/2.

Notice that the instability occurs well after the simulated robot's velocity has vanished and the robot is essentially at the origin. Hence, this instability in the curvature due to machine precision may be easily overcome by setting the control variables to zero when the desired velocity becomes less than some small value.

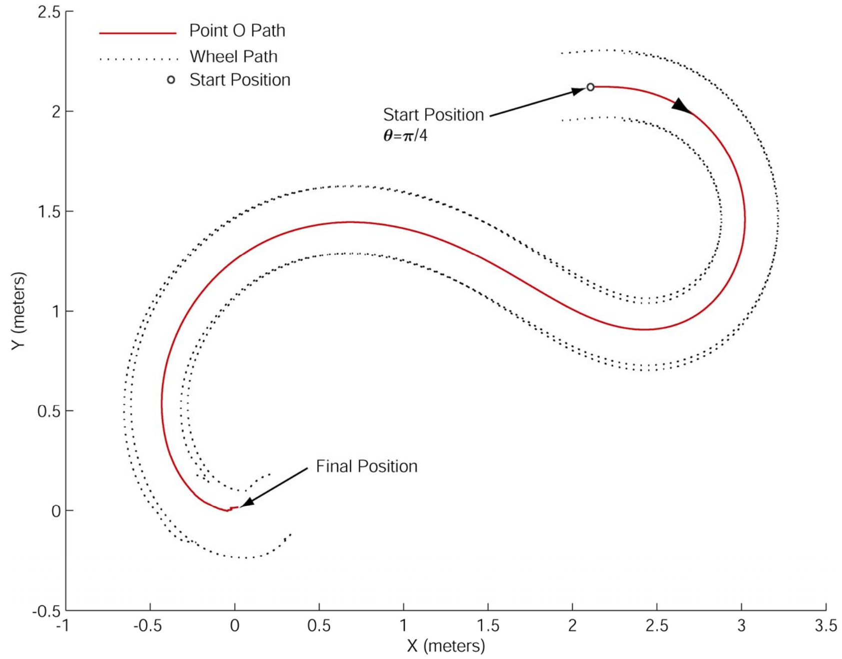
Indiveri Style Point Stabilization Experimental Results

The experimental results of the Indiveri style point stabilization controller showed excellent coherence to the simulation results presented above. Results of the two tests presented in Figure 20 show that the PCTR stabilizes to the origin as predicted. Simulation results are not shown in Figure 20, because they are essentially coincident with the experimental results.

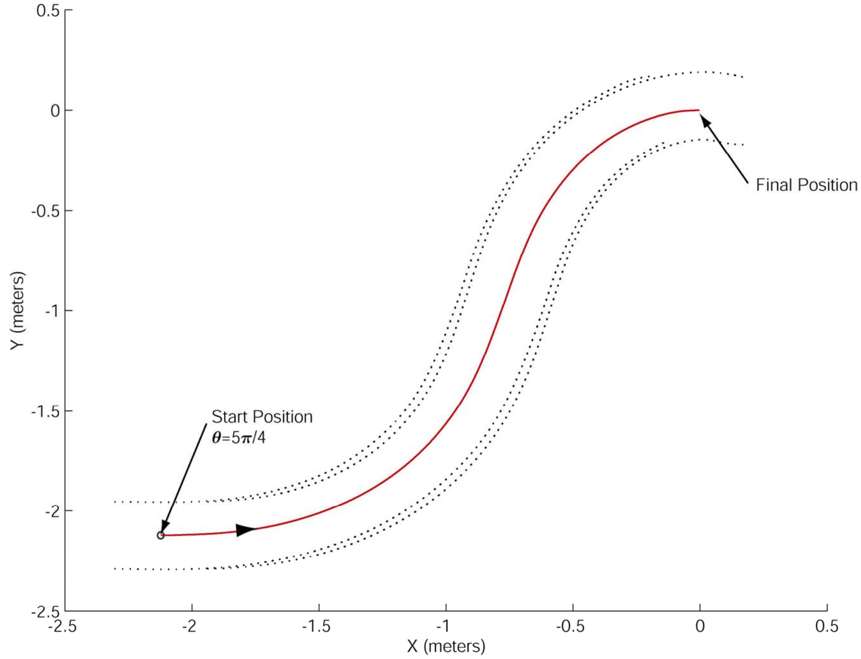
The major challenge faced in implementing the controller was that the controller proved to be especially sensitive to perturbations near the origin. All real world systems will have perturbations introduced into the system from unmodeled disturbances. The controller extension performed in Chapter IV guarantees the system will still stabilize to the origin in the presence of small perturbations. Problems occur as the robot approaches the origin though, because of the way the control variable c_d is defined. Equation (26) may be rewritten in the form

$$c_d = \frac{1}{e}\left(\sin\alpha + k_2\theta\frac{\sin\alpha}{\alpha} + k_3\alpha\right) \tag{100}$$

where it may be seen that as the polar state e tends smaller, it will act to make the control variable c_d larger. In Chapter IV the constants k_2 and k_3 were chosen to guarantee that the states θ and α would diminish faster than e, which will guarantee that c_d will tend to



(A) $\theta_0=\pi/4$



(B) $\theta_0=5\pi/4$

Figure 20: Experimental data collected on PCTR robot utilizing Indiveri style point stabilization controller. Initial condition e_0=3 meters, ϕ_0=0, and k_1=0.3, k_2=2, k_3=2.9, k_u=0.75, k_c=0.25, $\overline{u}$ =0.3 ϕ_0= -π/2.

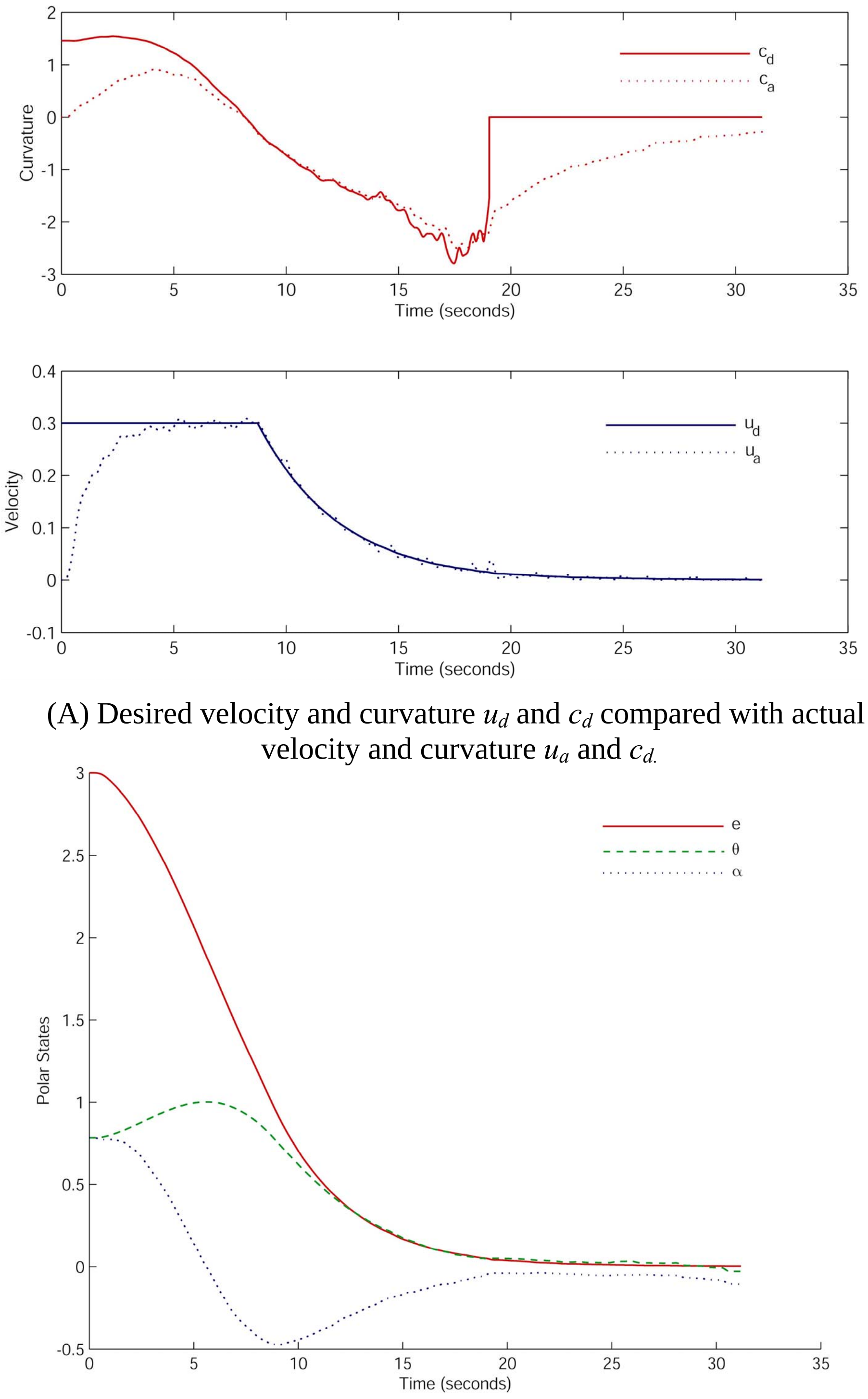
zero as the robot approaches the origin. This approach was verified in simulation, and it was seen that c_d tended to zero until computer precision error caused the variable to become unstable. This occurred well after the robot had reached its destination. In real world systems such as the PCTR, perturbations in the system will always cause some finite value of θ and α. Hence, it is no longer guaranteed that c_d will tend to zero as the polar state e tends to zero. In fact, it was observed that c_d does become quite large as the robot approaches the origin. This effect causes the robot to vibrate to the point of becoming unstable near the origin.

One major contributor to the perturbations that cause this instability is the encoder feedback. As discussed in Chapter V, the encoder feedback is a digital signal that resembles a staircase where each count of the encoder represents a step of approximately 0.7 degrees. The signal is differentiated to find the wheel angular velocity, which has the effect of magnifying the steps in the signal. Each jump of the encoder feedback causes a small jump in the value of θ and α and a corresponding jump in the control variable c_d. These jumps become magnified as e diminishes, hence the vibration induced near the origin. To compound the problem, the encoder feedback steps are more pronounced as velocity decreases, because there are fewer revolutions of the wheel, and correspondingly less encoder steps per unit of time. From the definition of u_d in (23) the velocity exponentially decreases as e diminishes near the origin.

To minimize the vibrations induced at the origin, the filter cutoff frequency ω_f was chosen as low as feasible, as discussed in Chapter V. This ensures as smooth an encoder feedback signal as possible. The controller gains k_c and k_v were also chosen to be very low. This ensures that the system response to any large c_d is very slow, and prevents the vibrations from growing unbounded as fast.

As a subject of future research, gain scheduling would likely improve the system performance. Small gains and a low filter cutoff frequency could be utilized at low velocities to minimize vibration. At higher velocities the gains and cutoff frequency could be switched to allow for improved system performance. Efforts are also currently under way to develop a dynamic controller that should improve performance by considering the complete system dynamics including the frame element.

In addition to selecting a low filter cutoff frequency and gains, a region was defined near the origin that upon entry of the PCTR robot into the region, the control variable c_d becomes zero. Therefore, once the robot enters this region, it ceases to try and diminish the states θ and α and only seeks to diminish e. This region will therefore prevent chatter near the origin if the robot can successfully enter the region. For the tests conducted on the PCTR shown in Figure 20, this region was defined as a 5 cm by 5 cm square area centered vertically at the origin and horizontally to the left of the origin. In consequence of defining this region, it is guaranteed only that the robot will stabilize within this square, but it was observed that generally the robot would stabilize to within 1 cm of the origin or less. The velocity and curvature of the PCTR for a stabilization maneuver are shown in Figure 21 (A). It can be seen that the robot enters the region at approximately 18 seconds, at which point c_d is set to zero. The polar states are presented in Figure 21 (B). Note that θ and α do not converge completely to zero because of the square region. Figure 22 shows snapshots of the PCTR performing a parking maneuver at different stages in time.



(A) Desired velocity and curvature u_d and c_d compared with actual velocity and curvature u_a and $c_{d.}$

(B) Polar states e, θ, and α.

Figure 21: Experimental control variable and state data collected on PCTR robot utilizing Indiveri style point stabilization controller. Initial conditions e_0=3 meters, θ_0=5π/4, ϕ_0=0, and k_1=0.3, k_2=2, k_3=2.9, k_u=0.75, k_c=0.25, $\overline{u}$ =0.3 ϕ_0=0.

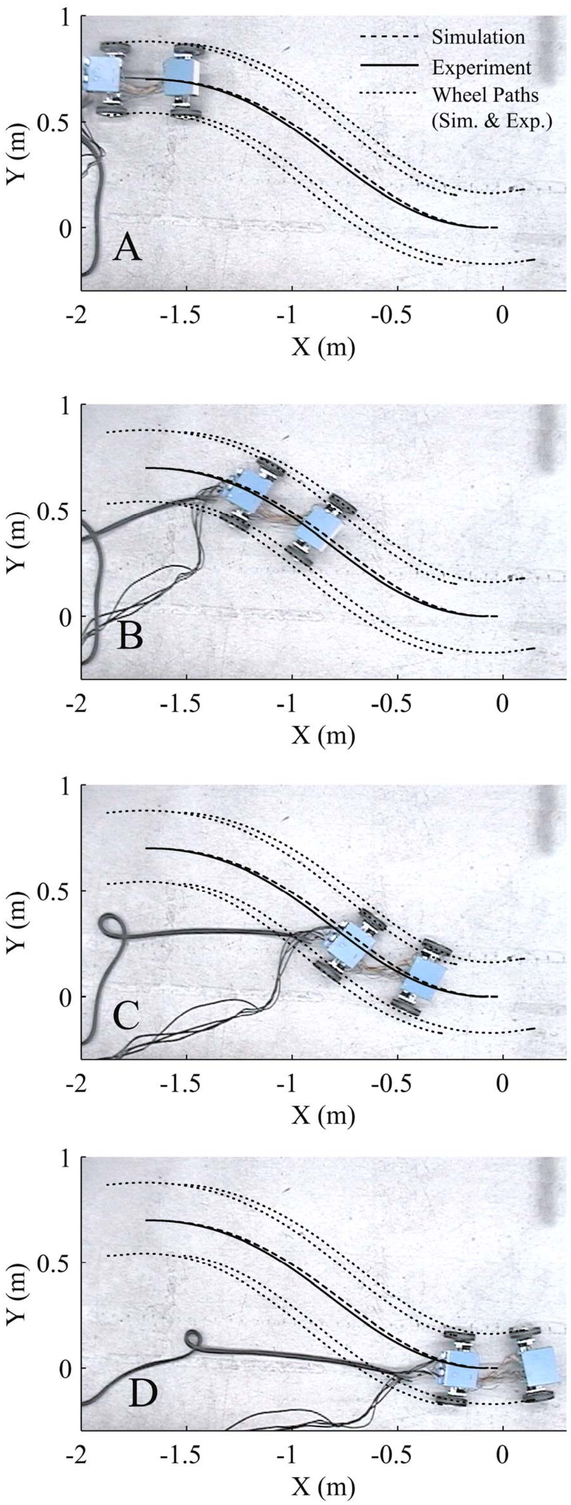


Figure 22: Robot snapshots. (A) t=0, (B) t=3, (C) t=5, (D) t=12 sec., for k_1=0.3, k_2=2, k_3=2.9, k_u=0.75, k_c=0.25, $\bar{u}$ =0.3, e_0=1.8 meters, θ_0=7π/8, ϕ_0=0. Simulation and experimental wheel paths are essentially coincident.

Tayebi Style Point Stabilization Simulation

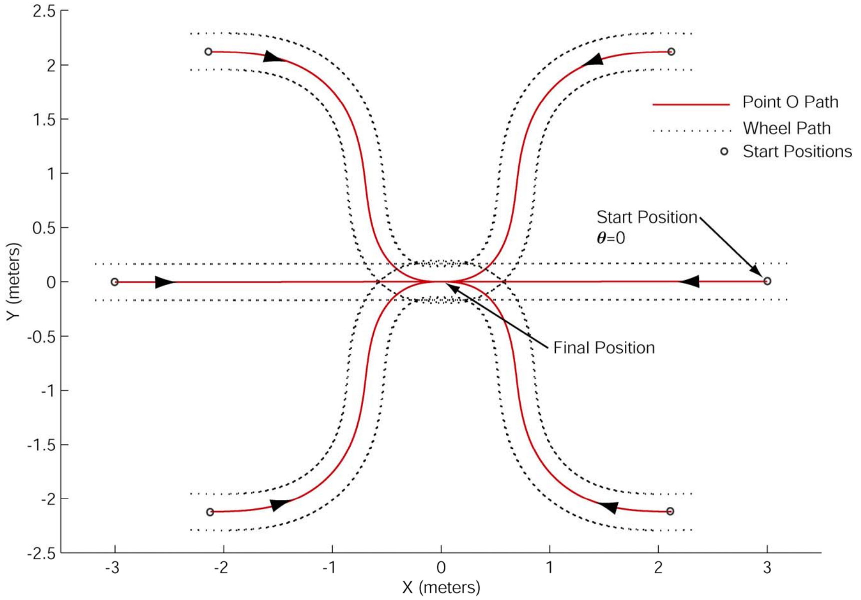
The modified Tayebi style point stabilization controller was tested in simulation for several different initial conditions to evaluate its performance in the parking problem. As shown in Figure 23, the controller generates smooth natural looking paths of bounded curvature similar to the Indiveri style point stabilization controller. Unlike the Indiveri controller, the paths do not all proceed in forward motion. Some paths may even reverse motion during the maneuver as shown in Figure 24.

The extension of the controller dynamics again performs well as is shown in Figure 25. The curvature and velocity of the simulated robot converge well to the desired curvature and velocity output by the controller, and the entire system converges to the origin. Figure 25 also shows that the polar states converge well to zero.

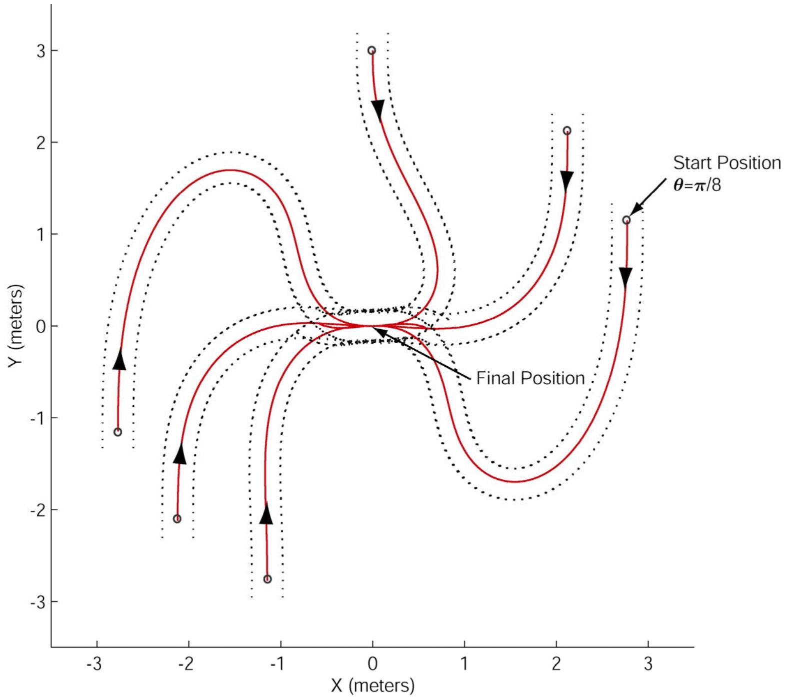
The controller also has the disadvantage that the control variable c_1 becomes unbounded as α approaches $\pm\pi/2$. This proved to cause problems if the initial condition for α of the simulated robot was near $\pm\pi/2$; thus no plots were generated in this region in Figure 23 and Figure 24. It may be possible to impose a saturation function on c_1, to prevent the variable from becoming unbounded, but this was not performed for this work. The Tayebi style point stabilization controller was not tested experimentally on the PCTR.

Path Following Simulation

The modified path following controller developed in Chapter IV was tested in simulation for several different path types to evaluate its performance in path following. Figure 26 and Figure 27 show the simulated robot tracking a line, circle and sine wave that were chosen as representative paths to show the ability of the controller. More



(A) $\phi_0 = 0$



(B) $\phi_0=\pi/2$

Figure 23: Simulation of CFR Tayebi style point stabilization controller paths. k_1=0.5, k_2=1.5, k_3=3, k_4=0, k_u=0.75, k_c=0.25.

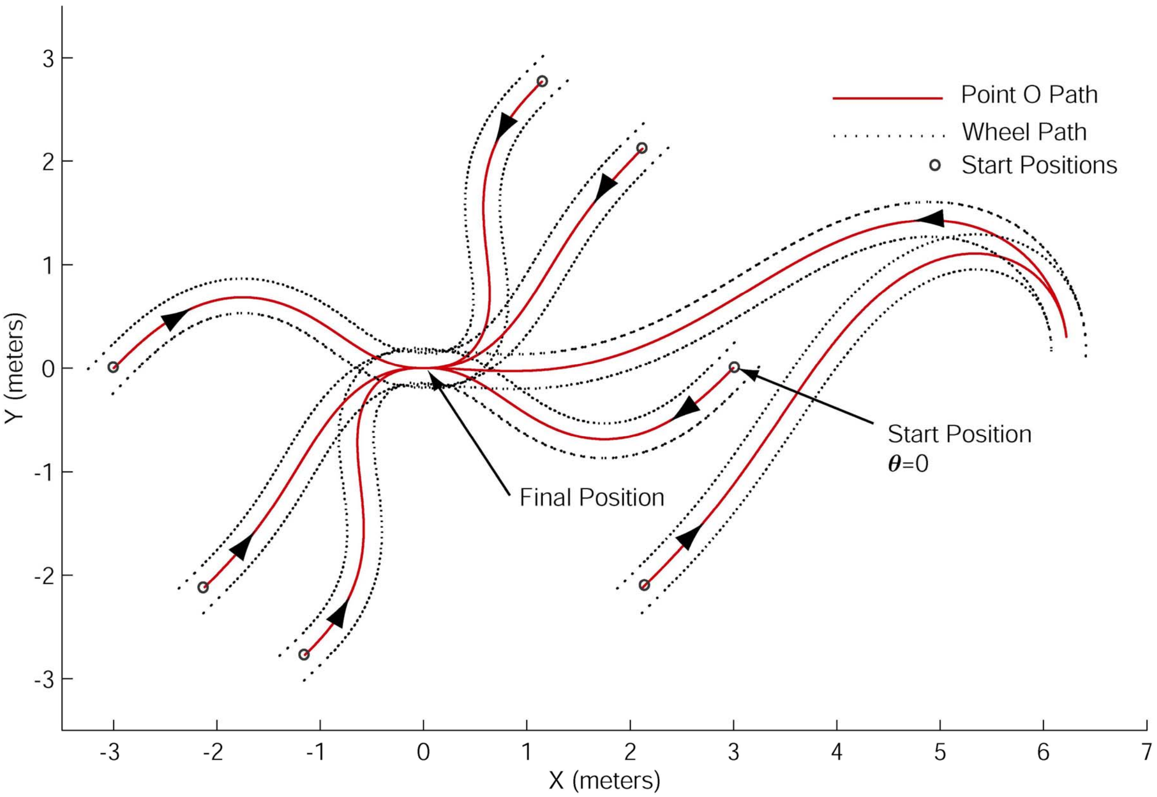
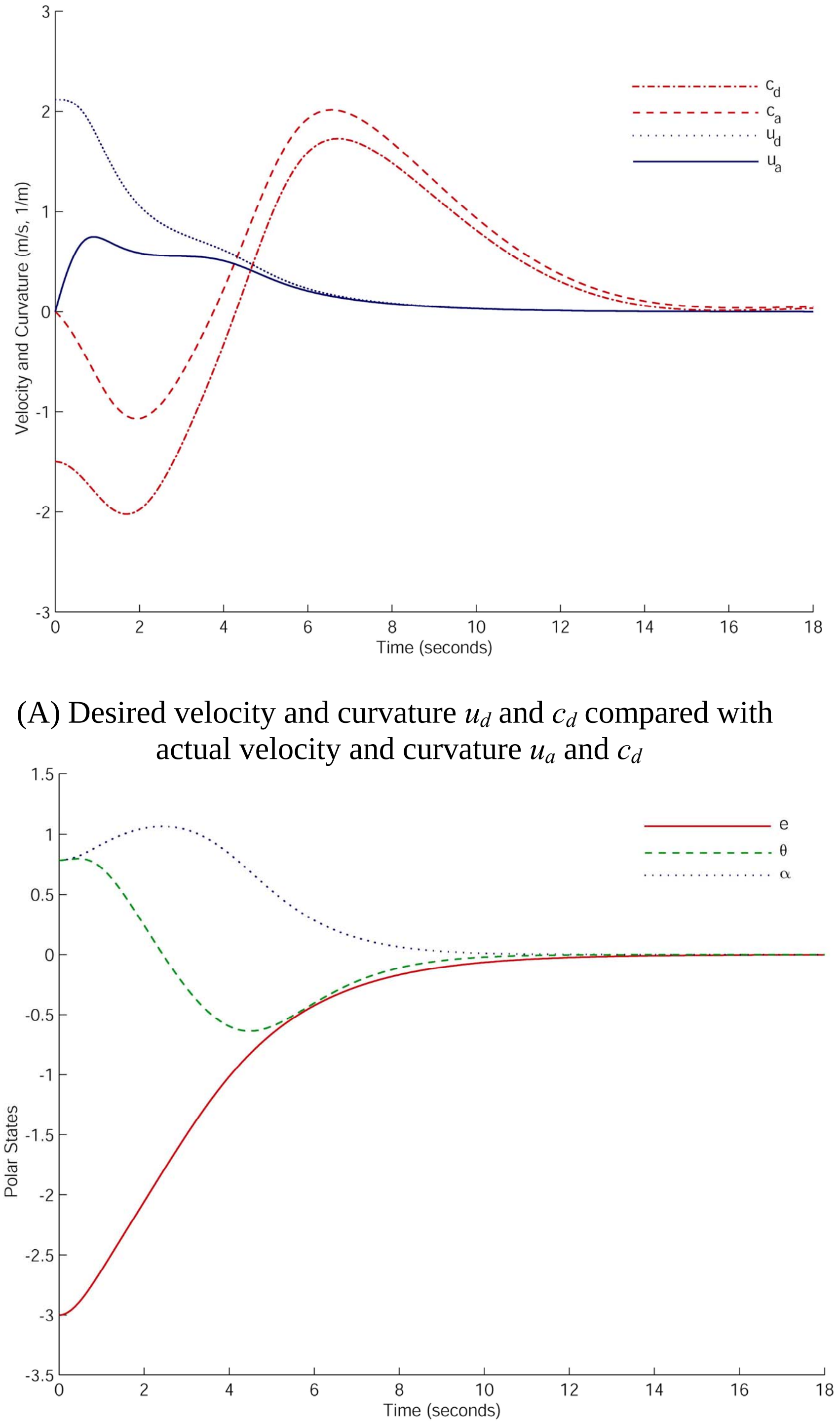


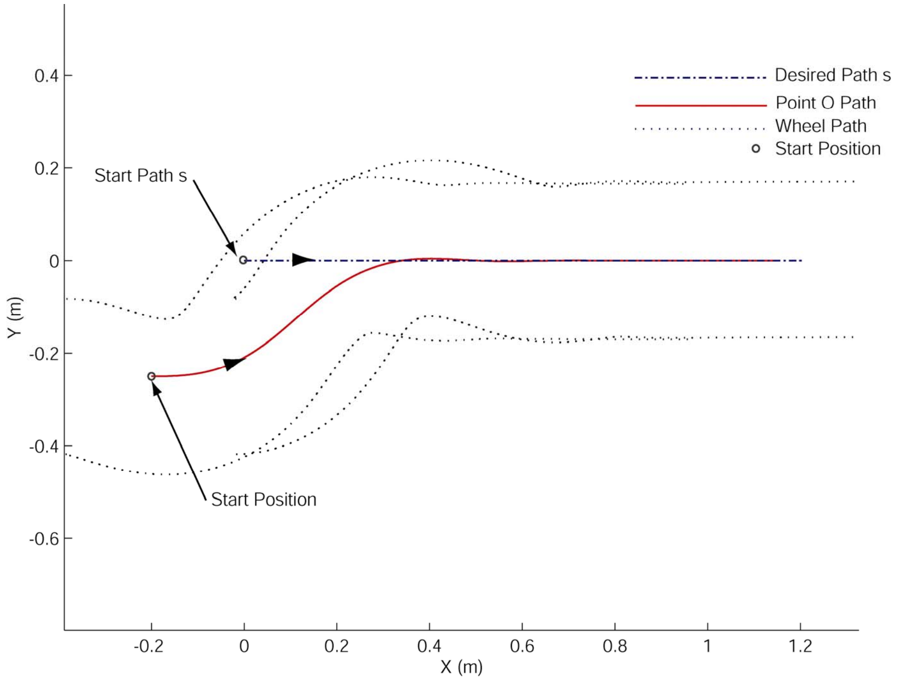
Figure 24: Simulation of CFR Tayebi style point stabilization controller path. Initial conditions $e_0 = 3$ meters, $\theta_0 = 0, \pi/4, 3\pi/8, -\pi/4$, $e_0 = -3$ meters, $\theta_0 = 0, -\pi/4, -3\pi/8$ and $k_1 = 0.5$, $k_2 = 1.5$, $k_3 = 3$, $k_4 = 0$, $k_u = 0.75$, $k_c = 0.25$, for $\phi_0 = \pi/4$.



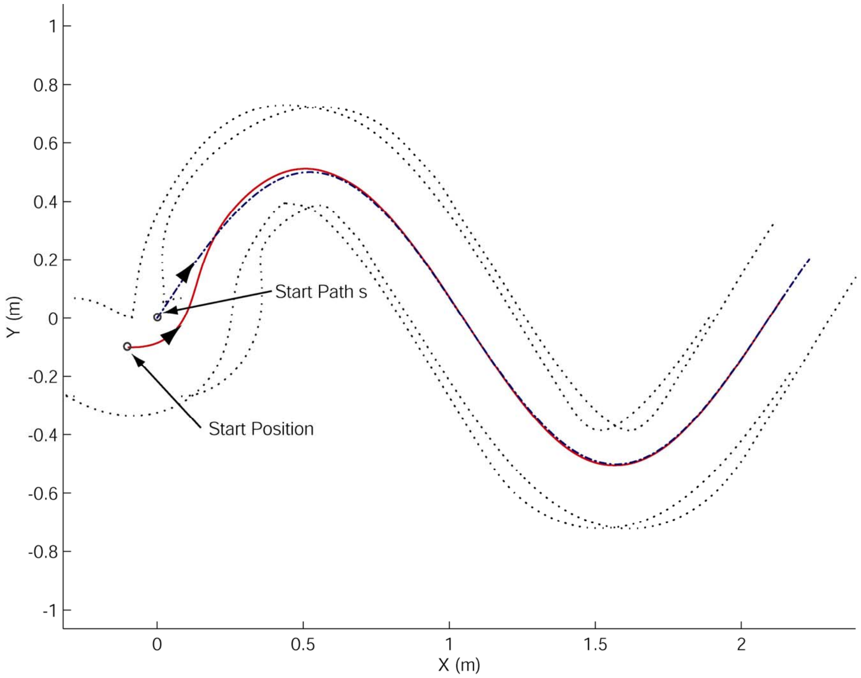
(A) Desired velocity and curvature u_d and c_d compared with actual velocity and curvature u_a and c_d

(B) Polar states e, θ, and α.

Figure 25: Simulation of CFR Tayebi style point stabilization controller control variables and states. Initial conditions e_0 = -3 meters, $\theta_0 = \pi/4$, $\phi_0 = 0$, and $k_1 = 0.5$, $k_2 = 1.5$, $k_3 = 3$, $k_4 = 0$, $k_u = 0.75$, $k_c = 0.25$.

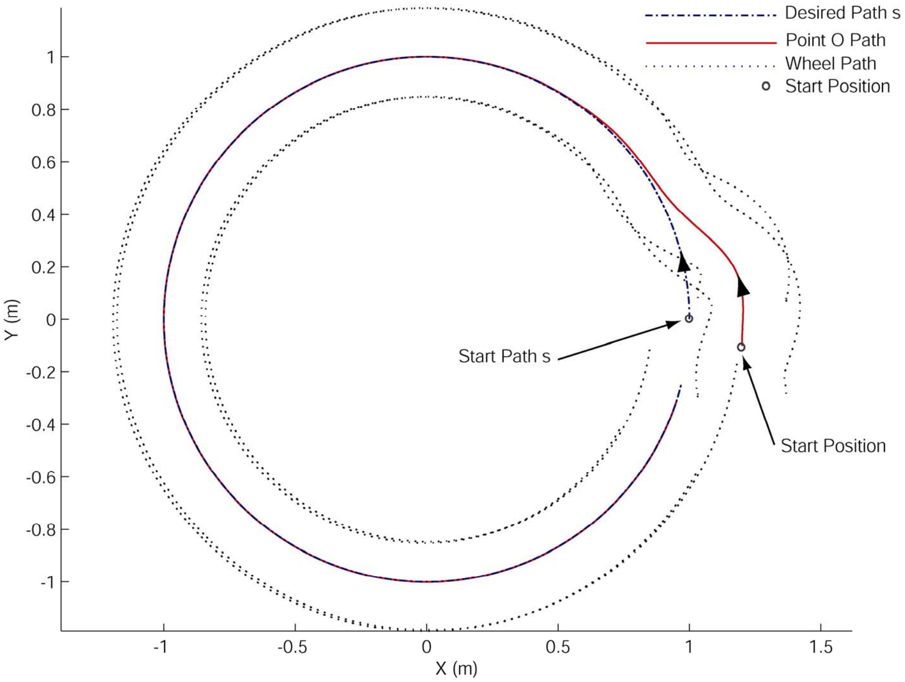


(A) Line with initial error of x_{err}= -0.2 meters, y_{err} = -0.25 meters, $\phi_{\mathrm{err}} = 0$, $\dot{s}_{des}$ = 0.2 m/s.

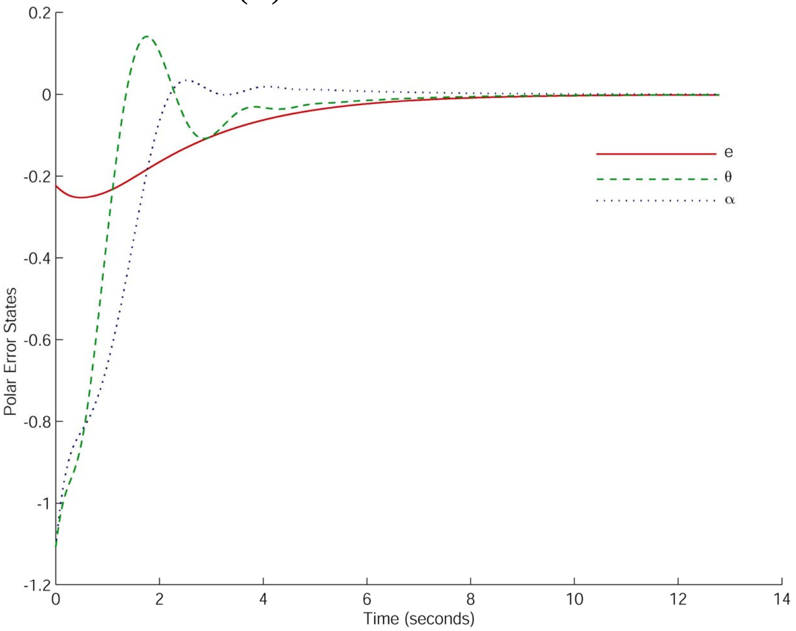


(B) Sine wave of 1 meter pk. to pk. amplitude and $2\pi/3$ period, initial error x_{err}= -0.1 meters, y_{err} = -0.1 meters, ϕ_{err}=-1 radian, $\dot{s}_{des}$ =0.05 m/s.

Figure 26: Simulation of CFR path following controller tracking a line and sine wave. $k_1 = 0.5$, $k_2 = 1.5$, $k_3 = 3$, $k_4 = 0.5$, $k_u = 2$, $k_c = 1$.



(A) Circular Path



(B) Polar states e, θ, and α.

Figure 27: Simulation of CFR path following controller tracking one meter radius circle. Initial error x_{err} = 0.2 meters, y_{err} = -0.1 meters, ϕ_{err} = 0, $\dot{s}_{des}$ = 0.1 m/s and k_1 = 0.5, k_2 = 1.5, k_3 = 3, k_4 = 0.5, k_u = 2, k_c = 1.

complicated paths could easily be built out of these basic paths types. It can be seen from the line tracking that the simulated robot will converge to the path even in the presence of significant initial error. The simulated robot tracks changing path curvature well in the case of the sine wave. Figure 27 illustrates that the polar error states quickly converge to zero as expected.

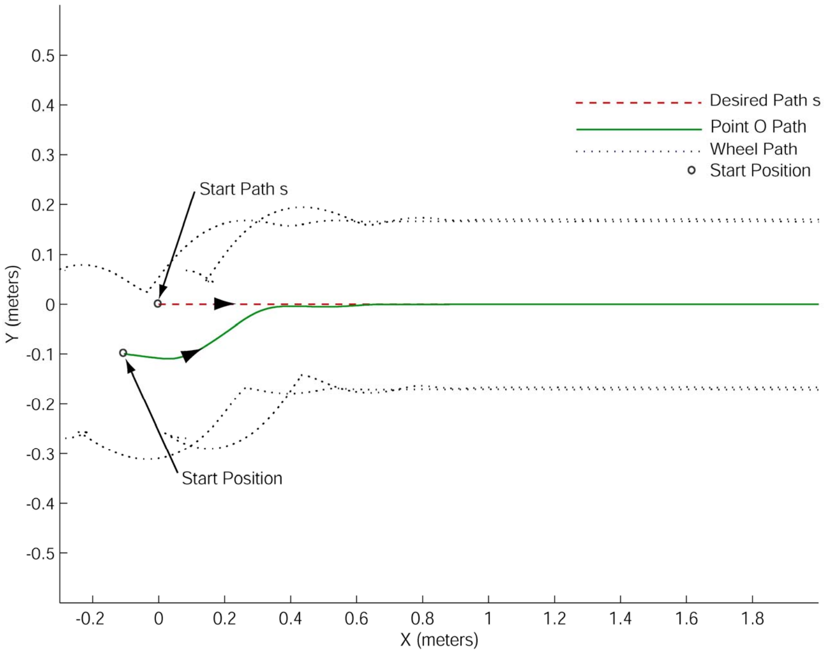
Path Following Experimental Results

The experimental results of the modified path following controller developed in Chapter IV showed excellent coherence to the simulation results presented above. Results of the two tests presented in Figure 28 and Figure 29 show that the PCTR follows the paths as predicted. Simulation results are not shown in the figures, because they are essentially coincident with the experimental results.

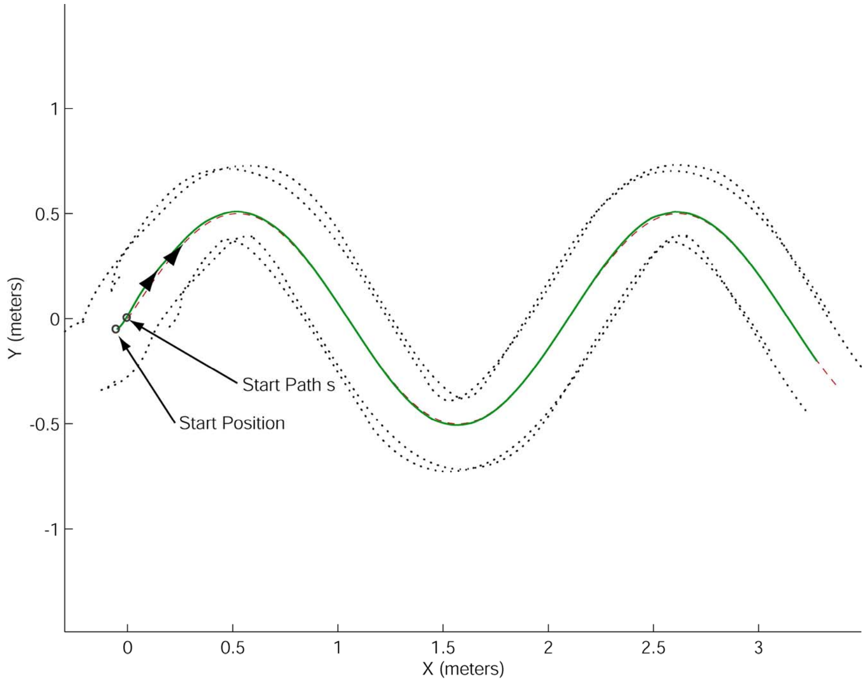
Like the experimental implementation of the Indiveri Style point stabilization controller, the major challenge faced in applying the controller proved to be sensitivity to perturbations near the moving origin. Equation (66) may be rewritten in the form

$$\begin{aligned}&\text{Left Half Plane}\\ &c_2 = -k_2\alpha + \frac{1}{e}\left(\frac{k_3 c_1 \theta \sin\alpha}{\alpha} + \frac{k_3 \dot{s}\theta \sin\alpha}{\alpha} + c_1 \sin\alpha + \dot{s}\sin\alpha - \dot{s}\sin\theta\right)\\ &\text{Right Half Plane}\\ &c_2 = k_2\alpha - \frac{1}{e}\left(\frac{k_3 c_1 \theta \sin\alpha}{\alpha} + \frac{k_3 \dot{s}\theta \sin\alpha}{\alpha} + c_1 \sin\alpha + \dot{s}\sin\alpha - \dot{s}\sin\theta\right)\end{aligned} \tag{101}$$

where it may be seen that as the polar error state e tends smaller, it will act to make the control variable c_2 larger. As in the point stabilization case, perturbations in the system

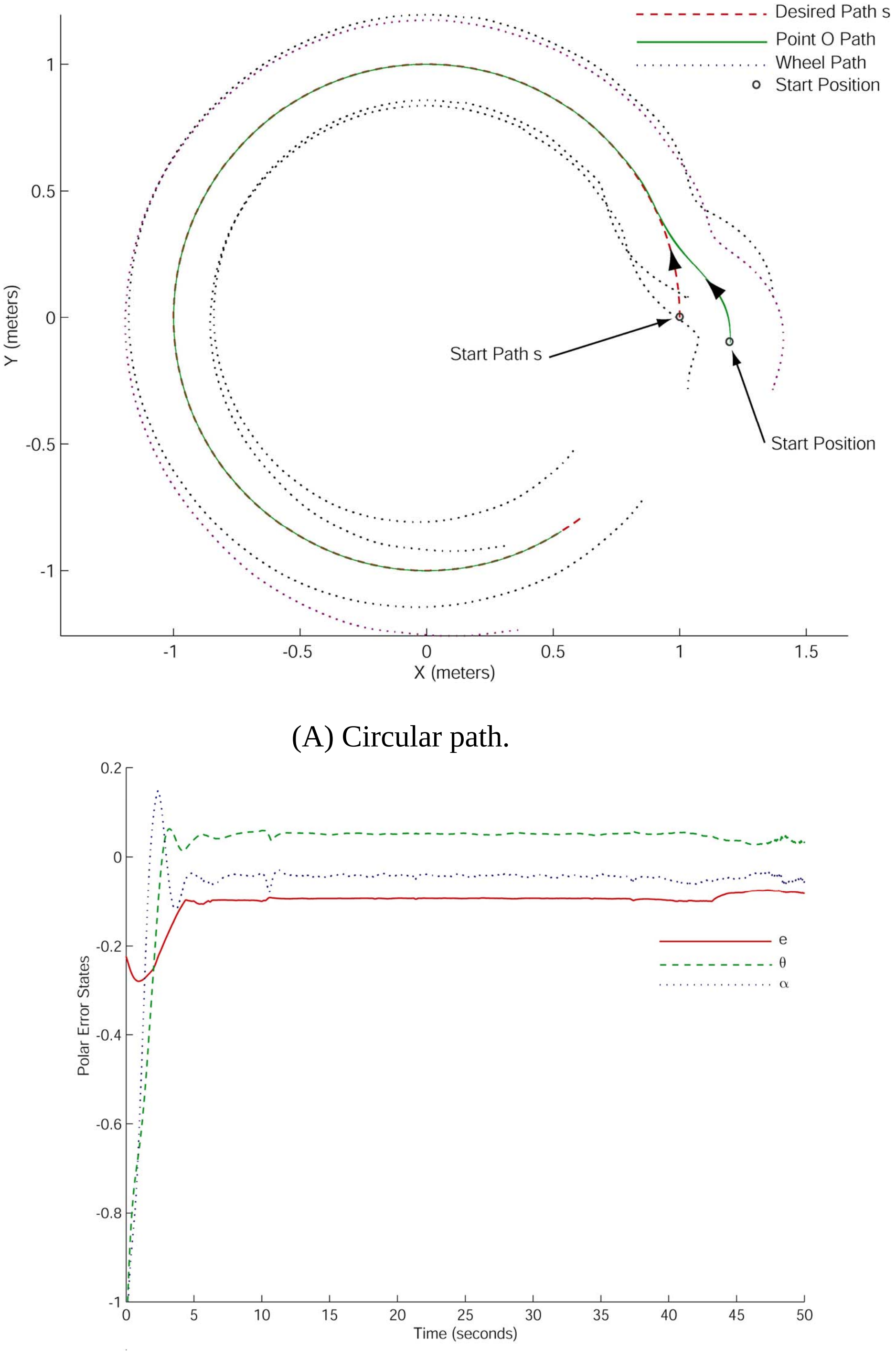


(A) Line with initial error of x_{err}= -0.1 meters, y_{err} = -0.1 meters, ϕ_{err}=0, $\dot{s}_{des}$ =0.1 m/s, ε=0.07 meters.



(B) Sine wave of 1 meter pk. to pk. amplitude and 2π/3 period, initial error x_{err}= -0.01 meters, y_{err} = -0.01 meters, ϕ_{err}= 0, $\dot{s}_{des}$ =0.05 m/s, ε=0.1 meters.

Figure 28: Experimental data collected on PCTR robot tracking a line and sine wave. k_1=0.5, k_2=1.5, k_3=3, k_4=0.5, k_u=2, k_c=1.



(A) Circular path.

(B) Polar states e, θ, and α.

Figure 29: Experimental data collected on PCTR robot tracking one meter radius circle. Initial error of x_{err}=0.2 meters, y_{err}= -0.1 meters, ϕ_{err}=0, $\dot{s}_{des}$ = 0.1 m/s, ε = 0.1 meters. k_1=0.5, k_2=1.5, k_3=3, k_4=0.5, k_u=1, k_c=1.

will always cause some finite value of θ and α, which causes vibrations as the robot approaches the moving origin.

This problem may simply be solved, in the path following case, by diminishing the control variable c_1 as in (75) when the polar error state $e < \varepsilon$. It was found for the experiments conducted on the PCTR that letting $\varepsilon = 7 - 10 \text{ cm}$ is sufficient. This implies that, after sufficient time, the robot will follow the path with a maximum $e = \varepsilon$. Figure 29 shows the polar error states for the tracking of a circular path. Note that e, θ and α do not converge completely to zero because c_1 is diminished when $e < \varepsilon$.

The error characteristics of the wheel controllers can be see in Figure 29 (A). As the robot proceeds around the circular path, error builds due to the tracking ability of the type one wheel controllers as discussed in Chapter IV. This causes the wheels to gradually drift out of the desired pure bending alignment. This provides further motivation for the implementation of gain scheduling or a dynamic wheel controller to help alleviate this error.

CHAPTER VII

CONCLUSIONS

There are many attractive advantages to the CFR robot platform. The simple hardware design lends itself well to the concept of modular, multipurpose robotics with the ability to be mass-produced. The greatest challenge in the novel concept has been the increased complexity of the robot kinematics and control compared to traditional mobile robot designs. Utilizing the equivalent kinematic formulation and control designs developed herein, the CFR concept proves that it is a viable design that may be controlled accurately in simulation and practice.

The equivalent kinematic formulation greatly reduces the complexity of the vehicle kinematics, allowing the vehicle to be controlled by the posture of a single point located at the center of the robot. This permits the kinematics to be described in a fashion similar to less complex vehicles, while ensuring the most efficient steering maneuvers, maximizing mobility, and minimizing required traction forces for improved towing capacity.

The CFR may be accurately controlled in the point stabilization and path following cases. Utilizing the equivalent kinematic formulation and describing the robot kinematics through polar state variables, time invariant, global exponential asymptotically converging control laws were developed for the posture stabilization problem. Using this same polar representation, a control law was developed for both the path following and posture stabilization case. These control laws may be extended to compensate for nonideal initial conditions and system drift.

The control laws perform well in both simulation and experimentally on the CFTR, provided actions are taken to account for real world perturbations near the origin. The trajectories generated are smooth and meet the desired requirements of the equivalent curvature based model. The experimental results matched the simulated results well and show that the performance of the robot may be easily predicted.

The CFR concept is still in its infancy, and there remain many areas for improvement in control of the platform. Further investigation needs to be conducted into the Tayebi style point stabilization controller to see if the requirement for $\alpha \neq \pm \pi/2$ may be removed. This would allow a single control law to be used for both the path following and posture stabilization problems for all initial conditions. The robot would benefit from an improved wheel controller that takes into account the dynamics of the entire system, including the compliant frame. Curvature feedback could be improved through the use of strain gauge sensors on the compliant frame element to ensure the desired pure bending condition is maintained. Despite the need for future investigation, this thesis has shown that the CFR concept is a viable design for mobile robots that may be controlled easily and efficiently. Most of all, it has shown that the CFR concept warrants further research and attention in the future.

APPENDIX A

ALTERNATE DERIVATION OF CONTROLLER EXTENSION

Based upon the Lyapunov function,

$$V \equiv \frac{1}{2}\left(\alpha^2 + k_2\theta^2\right) \tag{102}$$

Indiveri suggests the use of the control law

$$\begin{aligned} u &= k_1 e : \mathrm{sat}\left(\gamma \mathrm{e}, \overline{u}\right) \\ c &= \frac{\sin\alpha}{e} + k_2 \frac{\theta}{e}\frac{\sin\alpha}{\alpha} + k_3 \frac{\alpha}{e} \end{aligned} \tag{103}$$

which are smooth time functions that asymptotically drive the state (e, θ, α) towards the origin. The parameters k_1, k_2, and $k_3 > 0$ are constant gains and $\mathrm{sat}\left(k_1 \mathrm{e}, \overline{u}\right)$ is a positive continuous saturation function that prevents the proportional control input *u* to grow larger than some upper bound. The controller dynamics may be extended and written as a cascade system of the form

$$\begin{aligned} \dot{y} &= f(e, \alpha, \theta, u_a, c) \\ \dot{u} &= g(u_a, u_d) \\ \dot{c} &= j(c_a, c_d) \end{aligned} \tag{104}$$

where u_d and c_d are the desired velocity and curvature relations established by equation (103), and u_a and c_a are the actual measured feedback velocity and curvature. We then perform a change of coordinates,

$$\begin{aligned} Y &= y \\ U &= u_a - u_d \\ C &= c_a - c_d. \end{aligned} \tag{105}$$

where U and C are error states for our desired velocity and curvature. The transformed system becomes,

$$\begin{aligned} \dot{Y} &= f(e,\alpha,\theta,U + u_d, C + c_d) \\ &= F(e,\alpha,\theta,U,C) \\ \dot{U} &= \dot{u}_a - \dot{u}_d \\ \dot{C} &= \dot{c}_a - \dot{c}_d \end{aligned} \tag{106}$$

After applying the feedback,

$$\begin{aligned} \dot{u}_a &= k_u(u_d - u_a) + \dot{u}_d \\ \dot{c}_a &= k_c(c_d - c_a) + \dot{c}_d \end{aligned} \tag{107}$$

where k_u and k_c are positive constants, the closed-loop system becomes,

$$\begin{aligned} \dot{Y} &= f(e,\alpha,\theta,U,C) \\ \dot{U} &= -k_u U \\ \dot{C} &= -k_c C \end{aligned} \tag{108}$$

Thus, U and C are exponentially stabilized. As time goes to infinity, U=0 and C=0, and the system formulation becomes,

$$\begin{aligned} Y &= y = F(e,\alpha,\theta,0,0) \\ &= f(e,\alpha,\theta,u_d,c_d) \end{aligned} \tag{109}$$

which is asymptotically stabilized about the origin as Bacciotti [31] Theorem 19.2 shows. In summary, substituting u_D and c_D into (107) and combining with (15) the polar state equations of the compliant framed mobile robot now become,

$$\begin{aligned} \dot{u} &= k_u(k_1 e - u_a) + \dot{u}_d \\ \dot{c} &= k_c(\frac{\sin\alpha}{e} + k_2\frac{\theta}{e}\frac{\sin\alpha}{\alpha} + k_3\frac{\alpha}{e} - c_a) + \dot{c}_d \\ \dot{e} &= -u_a\cos\alpha \\ \dot{\alpha} &= -u_a\left(c_a - \frac{\sin\alpha}{e}\right) \\ \dot{\theta} &= u_a\frac{\sin\alpha}{e}. \end{aligned} \tag{110}$$

APPENDIX B

SYSTEM PARAMETERS

Table 3: Partially compliant test robot parameters

Parameter	Value
Frame Element	
Length, l	0.368 m
Height	0.051 m
Thickness	0.5 mm
Axle Half Width, a	0.343 m
Wheel Radius, r_w	0.076 m
Robot Mass	9.53 kg

Table 4: Partially compliant test robot component information.

Component	Manufacturer	Model	Miscellaneous
Motors	Maxon	F-Series 2140-937-22-116-050	R=44.6 Ω K_T=55.2 mNm/A K_n=173 rpm/V
Amplifiers	Advanced Motion Controls	10A8	
Amplifier Mounting Cards	Advanced Motion Controls	MC2X510	
Encoders	Agilent Technologies	HEDS-5540 A12	500 CPR

REFERENCES

[1] K. D. Kotay and D. L. Rus, "Task-reconfigurable robots: navigators and manipulators," presented at Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97, Grenoble, France, pp. 1081-9, 1997.

[2] S. Kimura, S. Tsuchiya, S. Nishida, and T. Takegai, "A module type manipulator for remote inspection in space," presented at IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics, Tokyo, Japan, pp. 819-24, 1999.

[3] A. Casal and M. Yim, "Self-reconfiguration planning for a class of modular robots," *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 3839, pp. 246-57, 1999.

[4] D. Rus, "Self-reconfiguring robots," *IEEE Intelligent Systems*, vol. 13, pp. 2-4, 1998.

[5] A. Castano and P. Will, "Representing and discovering the configuration of Conro robots," presented at Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, Seoul, South Korea, pp. 3503-9, 2001.

[6] A. Pamecha and G. Chirikjian, "A useful metric for modular robot motion planning," presented at Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, pp. 442-7, 1996.

[7] S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," presented at Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, pp. 441-8, 1994.

[8] K. D. Kotay and D. L. Rus, "Algorithms for self-reconfiguring molecule motion planning," presented at Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu, Japan, pp. 2184-93, 2000.

[9] M. Yim, K. Roufas, D. Duff, Y. Zhang, and S. Homans, "Modular reconfigurable robots in space applications," presented at 10th IEEE International Conference on Advanced Robotics (ICAR 2001), Budapest, Hungary, pp. 153-9, 2001.

[10] R. O. Ambrose, M. P. Aalund, and D. Tesar, "Designing modular robots for a spectrum of space applications," *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 1829, pp. 371-81, 1992.

[11] M. G. Bekker, "Vehicle with Flexible Frame." US Patent No. 3235020, 1962.

[12] P. L. Spanski, "Flexible Frame Vehicle." US Patent No. 3550710, 1970.

[13] J. Borenstein, "Control and kinematic design of multi-degree-of-freedom mobile robots with compliant linkage," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 21-35, 1995.

[14] C. Altafini, "Why to use an articulated vehicle in underground mining operations?," presented at Proceedings of International Conference on Robotics and Automation, Detroit, MI, USA, pp. 3020-5, 1999.

[15] A. Kemurdjian, V. Gromov, V. Mishkinyuk, V. Kucherenko, and P. Sologub, "Small Marsokhod configuration," presented at Proceedings - IEEE International Conference on Robotics and Automation, Nice, France, pp. 165-168, 1992.

[16] K. J. Waldron and C. J. Hubert, "Control of contact forces in wheeled and legged off-road vehicles," presented at Proceedings of Sixth International Symposium on Experimental Robotics, Sydney, NSW, Australia, pp. 205-14, 2000.

[17] C. Schwensen, "Control of a compliant, fixed-wheeled vehicle via independent wheel control," in *Department of Mechanical Engineering*. Salt Lake City: University of Utah, pp. 58, 2001.

[18] B. Albiston and M. Minor, "Curvature based point stabilization for compliant framed wheeled modular mobile robots," presented at International Conference on Robotics and Automation, Taipei, Taiwan, 2003.

[19] G. Indiveri, "Kinematic time-invariant control of a 2D nonholonomic vehicle," presented at Proceedings of 1999 Conference on Decision and Control, Phoenix, AZ, USA, pp. 2112-17, 1999.

[20] A. Tayebi and A. Rachid, "A unified discontinuous state feedback controller for the path-following and the point-stabilization problems of a unicycle-like mobile robot," presented at Proceedings of the 1997 IEEE International Conference on Control Applications, Hartford, CT, USA, pp. 31-5, 1997.

[21] R. W. Brockett, R. S. Millman, and H. J. Sussman, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*. Boston: Birkhauser, pp. 181-191, 1983.

[22] C. Samson, "Control of chained systems application to path following and time-varying point-stabilization of mobile robots," *IEEE Transactions on Automatic Control*, vol. 40, pp. 64-77, 1995.

[23] R. Colbaugh, E. Barany, and K. Glass, "Adaptive control of nonholonomic robotic systems," *Journal of Robotic Systems*, vol. 15, pp. 365-93, 1998.

[24] A. Tayebi, M. Tadjine, and A. Rachid, "Invariant manifold approach for the stabilization of nonholonomic chained systems: application to a mobile robot," *Nonlinear Dynamics*, vol. 24, pp. 167-81, 2001.

[25] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Transactions on Neural Networks*, vol. 9, pp. 589-600, 1998.

[26] I. Kolmanovsky and N. H. McClamroch, "Developments in nonholonomic control problems," *IEEE Control Systems Magazine*, vol. 15, pp. 20-36, 1995.

[27] A. Astolfi, "On the stabilization of nonholonomic systems," presented at Proceedings of 1994 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, pp. 3481-6, 1994.

[28] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle like vehicles via Lyapunov techniques," *IEEE Robotics & Automation Magazine*, vol. 2, pp. 27-35, 1995.

[29] F. Diaz del Rio, G. Jimenez, J. L. Sevillano, S. Vicente, and A. Civit Balcells, "A generalization of path following for mobile robots," presented at Proceedings of International Conference on Robotics and Automation, Detroit, MI, USA, pp. 7-12 vol.1, 1999.

[30] H. K. Khalil, *Nonlinear Systems*, 3rd. ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[31] A. Bacciotti, *Local Stabilizability of Nonlinear Control Systems*, vol. 8, Singapore: World Scientific, 1991.

[32] S. P. T. James M. Gere, *Mechanics of Materials*, 4th ed. Boston, Massachusetts: PWS Publishing Company, 1997.

[33] C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 4th ed. Upper Saddle River, New Jersey: Prentice-Hall, 2000.