



LIBRARY
Michigan State
University

This is to certify that the

thesis entitled

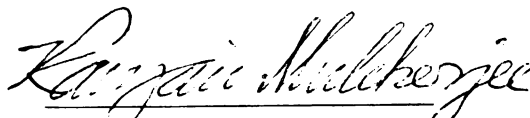
**CONTROL OF NONHOLONOMIC SYSTEMS
WITH ROLLING CONSTRAINTS**

presented by

Jay Tawee Pukrushpan

has been accepted towards fulfillment
of the requirements for

M.S. degree in Mechanical Engineering


Major professor

Date 8/28/98

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Control of Nonholonomic Systems with Rolling Constraints

By

Jay Tawee Pukrushpan

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1998

ABSTRACT

Control of Nonholonomic Systems with Rolling Constraints

By

Jay Tawee Pukrushpan

Two control strategies for two nonholonomic systems with rolling constraints are presented. These systems are the two-wheeled mobile robot and the rolling sphere. For the mobile robot, the problem of point-to-point stabilization is addressed. It is well-known that there does not exist a smooth pure-state feedback that can stabilize the mobile robot to an arbitrary fixed posture. In the past, smooth time-varying feedback, non-smooth time-invariant feedback and their hybrid combination have been used for stabilization of mobile robots. In this report, a new feedback control strategy is presented. Based on a nonlinear oscillator, it generates almost-smooth time-invariant control action that can asymptotically converge the robot from practically any posture to the desired posture. The only exception is an equilibrium manifold of measure zero from where the robot cannot move. These points are however unstable equilibria and, therefore points arbitrary close to them can be asymptotically converged to the desired posture. Simulation results are shown to demonstrate the capability of the controller. Our controller can handle the situation where the robot

has to turn more than 360° between initial and final configurations. Such capability will be useful for tethered mobile robots in factory automation.

We also present an open loop control strategy for the rolling sphere. The open loop controller is based on the concept of geometric phase which is quite popular in nonholonomic motion planning. It uses the idea that when a sphere rolls along a spherical triangle on its surface, the sphere undergoes a net displacement in the Cartesian coordinates with a pure rotation about the vertical axis. The magnitude and direction of the displacement are related to the dimensions of the sphere and the amount of rotation about the vertical axis is proportional to the surface area of the spherical triangle. The relation between the amount of rotation and the surface area is also shown to be true for any arbitrary closed curve on the surface of the sphere. For a given desired location and orientation, a method for finding the spherical triangular path is demonstrated. Simulations are presented to subsequently show convergence of system variables for the motion of the sphere along the spherical triangle.

To my parents

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my major advisor, Dr. Ranjan Mukherjee, for his support, guidance and patience. It has always been a great pleasure to work with him. I would also like to thank the other members of my committee, Dr. Ronald Rosenberg and Dr. Steven W. Shaw for their helpful advice. My appreciation also goes to all my friends in Dynamics Lab, especially Joga Setiawan and Mark Minor for their promptly assistance with my computer difficulties.

I also want to thank Thai government for giving me a scholarships for my graduate study. I would finally like to give special thanks to my dearest family, my dad, my mom and my sister, for their love, care and support. My last thank goes to Chutamanee for her love and encouragement.

TABLE OF CONTENTS

List of Tables	vii
List of Figures	viii
1 Background and Introduction	1
2 Feedback Stabilization of a Two-Wheeled Mobile Robot	6
2.1 Motivation	6
2.2 Kinematic Model of Mobile Robot	7
2.3 Almost-Smooth Time-Invariant Feedback Control Strategy	9
2.3.1 Transformation of variables	9
2.3.2 The feedback control strategy	10
2.3.3 Convergence and stability	12
2.4 Simulation Results	16
3 Open-Loop Control of The Rolling Sphere	21
3.1 Motivation	21
3.2 Review of Spherical Trigonometry	22
3.3 Sphere rolling along the spherical triangle	26
3.4 Sphere rolling along lune	33
3.5 Sphere rolling along an arbitrary closed path	36
3.6 Kinematic Model of Sphere	38
3.7 Formulation of Open Loop strategy	41
3.8 Additional simulations supporting open-loop control strategy	52
4 Conclusion	59
Appendix	62
A Program for calculating vector rotation	62
B Spherical trigonometric identities	65
C Simplification of rotation matrix	67
D Fortran program for numerical root finding	76
Bibliography	82

LIST OF TABLES

2.1	Initial configuration of mobile robot	16
2.2	Controller parameter used in simulations of mobile robot	17
3.1	Numerical values used in simulations 1, 2 and 3 for open loop control of the sphere rolling along spherical triangles.	46
3.2	Results from simulations 4 and 5 of the open loop control of the sphere rolling along arbitrary closed paths.	56

LIST OF FIGURES

2.1	Diagram showing the state variables of the mobile robot	8
2.2	Circle in the $r - \sigma\beta$ plane defining the Lyapunov function V	13
2.3	Trajectory of the mobile robot for Case A.	17
2.4	Trajectories of the mobile robot for Case B with different set of control parameter.	18
2.5	Trajectory of the mobile robot for Case C.	19
2.6	Trajectory of the mobile robot for Case D.	19
2.7	Trajectory of the mobile robot for Case E.	20
3.1	A spherical triangle	23
3.2	The spherical triangle that specify the path in figure 3.3	27
3.3	Motion of the sphere defined by the spherical triangle in figure 3.2 . .	27
3.4	Path generated by the spherical triangle similar to the one in figure 3.3 but with opposite direction.	31
3.5	The Lune	33
3.6	Cartesian path given by the Lune in figure 3.5	34
3.7	Arbitrary Spherical Polygon	36
3.8	Arbitrary closed path that crosses itself	37
3.9	Figure defining the state variables of a sphere	39
3.10	Open loop control strategy of spherical robot	41
3.11	Possible values of d and α for one spherical triangle	44
3.12	Possible values of d and ϕ for one spherical triangle	44
3.13	Possible values of α and ϕ for one spherical triangle	45
3.14	System variables used in simulation	45
3.15	Trajectory of the sphere in simulation 1.	47
3.16	Variables d, ϕ and α from simulation 1 for open loop control of sphere.	48
3.17	Variables θ, x, x_p, y, y_p from simulation 1 for open loop control of sphere.	48
3.18	Trajectory of the sphere in simulation 2.	49
3.19	Variables d, ϕ and α from simulation 2 for open loop control of sphere.	49
3.20	Variables θ, x, x_p, y, y_p from simulation 2 for open loop control of sphere.	50
3.21	Trajectory of the sphere in simulation 3.	50
3.22	Variables d, ϕ and α from simulation 3 for open loop control of sphere.	51
3.23	Variables θ, x, x_p, y, y_p from simulation 3 for open loop control of sphere.	51

3.24	Parameters of closed curve	53
3.25	Surface integral for arbitrary closed curve on sphere	55
3.26	Trajectory of the sphere in simulation 4. Solid line represents trajectory of center of the sphere, dashed line represents trajectory of point P . .	56
3.27	Variables θ and α from simulation 4 for open loop control of sphere. .	57
3.28	Trajectory of the sphere in simulation 5. Solid line represents trajectory of center of the sphere, dashed line represents trajectory of point P . .	57
3.29	Variables θ and α from simulation 5 for open loop control of sphere. .	58

CHAPTER 1

Background and Introduction

In this chapter, we provide a brief introduction and some background in motion planning and control of nonholonomic system. Specifically, we define nonholonomic systems, discuss their characteristics, and summarize the two main problems namely motion planning and stabilization. A general introduction to nonholonomic systems can be found in the book by Murray, Li, and Sastry [26], and a review of the recent literature can be found in the review article by Kolmanovsky and McClamroch [18].

Nonholonomic systems are characterized by nonholonomic constraints of motion. Such constraints are not integrable and therefore they can not be used to solve for a set of independent generalized coordinates. Nonholonomic systems consequently require more coordinates for their description than there are degrees of freedom. Nonholonomic constraints usually arise in two kinds of situations. They appear in rolling contacts where objects roll without slipping. Examples of such systems are mobile robots, wheeled vehicles, the rolling disk and the rolling sphere. Nonholonomic constraints also appear in systems that conserve angular momentum. A free-flying space robot is an example of such a nonholonomic systems. A nonholonomic system such as the wheeled snake board [22] is a result of both the rolling constraint and the conservation of angular momentum.

There are some important properties of nonholonomic systems. First, a set of in-

dependent generalized coordinates does not exist since the non-integrable constraints can not be used to eliminate the dependent generalized coordinates. Secondly, non-holonomic control systems do not have isolated equilibrium points and hence the origin cannot be asymptotically stabilized using pure state feedback [7]. Finally, the nonholonomic characteristic of a given system is invariant with respect to diffeomorphic state transformations and with respect to smooth pure state feedback.

Generally, there are two problems that have been studied for nonholonomic systems, namely, motion planning and stabilization. Motion planning problems are concerned with obtaining open loop controls which will steer the system from an initial state to a final state over a given finite time interval. The difficulty of motion planning for nonholonomic system originates from the non-existence of a set of independent generalized coordinates, which means that not every motion is feasible. Only motions that satisfy the instantaneous nonholonomic constraints are feasible. Researchers have looked into efficient means for planning feasible trajectories and a number of solutions have been proposed. Optimal-trajectory problems and problems of path planning with obstacle avoidance have also been addressed.

Motion planning methodologies can be loosely classified into three methods, namely, differential-geometric techniques, geometric phase based methods, and control parameterization approaches. A variety of motion planning techniques are described in two books ([23] and [26]) and in a review article [18]. Most motion planning tools in differential-geometric techniques are based on Lie-algebra. Tools have been developed to generate motions in the directions of iterated Lie brackets that represent directions that violate the instantaneous constraints. The idea of using piecewise constant inputs to generate motions in the directions of iterated Lie brackets has been proposed by Lafferiere [19] and Lafferiere and Sussmann [20]. Methods based on averaging theory has been developed by Gurvits and Li [12], Leonard and Krishnaprasad [21], and Sussmann and Liu [33]. The fundamental idea of motion planning using ge-

ometric phase is that the change in some states or function of states, called geometric phase, depends on the geometry of the path. The appropriate path that produces the desired geometric phase can be usually solved as a function of path parameters. Examples of publications of the techniques using geometric phase include [6], [12] and [25]. Motion planning using parameterization of inputs are based on a finite dimensional family of functions. Steering the system using a family of sinusoids by Murray and Sastry [27] is an example of this method.

The control problem of tracking is defined as the task of finding a feedback control law that asymptotically stabilizes the motion of the system about a reference trajectory. Solutions to this problem have been developed using classical nonlinear control approaches, as in [9, 15, 35, 30]. The other extensively studied control problem for nonholonomic systems is feedback stabilization. These problems are concerned with obtaining feedback laws, which guarantee asymptotic stability of the equilibrium of the closed loop system. It follows from Brockett's necessary condition for feedback stabilization [7] that there exists no smooth pure state feedback that can asymptotically stabilize a nonholonomic system. The stabilization problems are therefore inherently more difficult than the problem of tracking. In agreement with Brockett's result, primarily two types of stabilizing controllers have been developed, namely, smooth time-varying controllers and non-smooth time-invariant controllers. Hybrid combinations of the two controllers have also been proposed.

Explicit time-varying control laws for feedback stabilization of nonholonomic systems was first proposed by Samson [31, 32]. General existence results can be found in [8] and explicit time periodic feedback laws can be found in [28]. The convergence properties of time-varying feedback has been studied by Gurvits and Li [12] and M'Closkey and Murray [24]. In general, the smooth time-varying controllers are known to have slow convergence rates and faster convergence can be achieved through the design of non-smooth controllers. Bloch [5] initiated the work on discontinuous

controllers for nonholonomic system. Piecewise continuous stabilizing controllers with exponential convergence rates were subsequently developed by Canudas de Wit and Sordalen [10], and Khennouf and Canudas de Wit [17]. In a somewhat different strategy a non-smooth state transformation was used to design a smooth time-invariant controller in the transformed coordinates [1, 2]. The feedback law in the original coordinates for both cases is however discontinuous. Discontinuous controllers based on sliding modes have also been proposed by a number of researchers, [11, 4], but such controllers often lead to chattering.

In this thesis, we concentrate on the control of two nonholonomic systems, namely a two-wheeled mobile robot and the rolling sphere. We introduce a new design method for the feedback control of a two-wheeled mobile robot based on a nonlinear oscillator. Because the stiffness and damping can be functions of the robot states, nonlinear oscillators offer more flexibility than undamped linear oscillators that have been widely used in time-varying control of nonholonomic systems. Our controller generates "almost smooth" [13] time-invariant control action that can asymptotically converge the robot from practically any posture to the desired posture. The only exception is an equilibrium manifold of measure zero (containing the desired posture) from where the robot cannot move. All points on this manifold (except the desired posture) are however rendered unstable and therefore any posture arbitrarily close to, but not exactly on, the manifold can be asymptotically converged to the desired posture. This gap in the posture space is not a limitation of the controller.

The control of the rolling sphere is inspired by the idea of the spherical robot. A spherical robot is conceived to have a spherical exoskeleton and an internal mechanism for self-propulsion. The robot will have a retractable camera that will be deployed and retracted through an orifice in the exoskeleton. Therefore, the motion of the sphere has to be controlled such that this point of deployment of the camera always appear on the top when the robot comes to rest. In this thesis, we design an open loop

control strategy, one that borrows ideas from spherical trigonometry and is based on the concept of geometric phase. The basic idea is that when a sphere rolls along a path described by a spherical triangle on the surface of the sphere, the sphere undergoes a net displacement in the Cartesian coordinates with a pure rotation about the vertical axis. The amount of rotation and the direction and magnitude of displacement depend on the dimensions of the spherical triangle. In this thesis, the idea of the spherical triangle is later extended for motion of the sphere along any arbitrary closed curve on its surface.

This thesis is organized as followed. The control strategy of the two-wheeled mobile robot is presented in chapter 2. After a brief motivation, the kinematic model of the mobile robot is presented, followed by the design of the closed loop controller. The convergence and stability of closed loop system trajectories are demonstrated next. Simulation results are then presented to demonstrate the capabilities of the controller. Chapter 3 presents an open loop control strategy for the rolling sphere. This chapter first provides some motivation and the fundamentals of spherical trigonometry. The expression for the amount of rotation of the sphere rolling along a spherical triangle, a two-sided curve are then derived. The kinematic model of the sphere, used in simulation, is presented next. Simulation results are presented at the end of the chapter to demonstrate the open loop control strategy. Chapter 4 provides concluding remarks and makes suggestion about the scope of future research.

CHAPTER 2

Feedback Stabilization of a Two-Wheeled Mobile Robot

2.1 Motivation

During the past seven decades, the number of robots used in industries has grown enormously. Most of these robots are designed to be rooted to the ground where they perform their tasks. Because they are fixed, their workspace is limited to certain distance from their location. Larger size robots are needed to handle larger workspaces. Mobile robots have the advantage of a large workspace compared to the size of the robot. Such robots are routinely used to transport parts between machining centers in automotive industries. If mobile robots can be made to be autonomous, and they have maneuvering capability, they can be used in remote areas such as in space exploration. In order to have satisfactory performance, mobile robots need to possess intelligence, which is normally associated with their control and sensing systems.

The mobile robot control problem belongs to the class of nonholonomic control systems since the rolling contact at its wheels results in non-integrable velocity constraints. In this thesis, a new control strategy for the mobile robot is introduced. The controller, based on a nonlinear oscillator can generate almost-smooth [13] time-

invariant control action that can asymptotically converge the mobile robot from practically any posture to the desired posture. The only exception is an equilibrium manifold of measure zero (containing the desired posture) where the robot cannot move. However, all points on the manifold (except the desired posture) are rendered unstable and therefore any posture arbitrarily closed to the manifold can be asymptotically converged to the desired posture.

The mathematical model of mobile robot and the state transformation are presented in section 2.2. In section 2.3, we propose the control strategy and establish the stability and convergence properties of the closed loop system trajectories. Simulation results are presented in section 2.4 to demonstrate the capability of the controller to converge the mobile robot to different desired postures.

2.2 Kinematic Model of Mobile Robot

The kinematic of a two-wheeled mobile robot, shown in Figure (2.1), is given as

$$\dot{x} = \cos \theta u_1 \quad (2.1a)$$

$$\dot{y} = \sin \theta u_1 \quad (2.1b)$$

$$\dot{\theta} = u_2 \quad (2.1c)$$

where x and y denote the Cartesian coordinates of the robot and θ denotes the orientation of the robot with respect to the positive x axis. The control inputs u_1 and u_2 are the linear and angular velocities of the robot, respectively.

In reference to Figure 2.1, it is assumed that x , y and θ denote the current configuration of the robot. The desired configuration is assumed to be zero. If they are not zero, a coordinate transformation can be used to set them equal to zero. Now consider the following state and control transformation [31] of the kinematic equations given

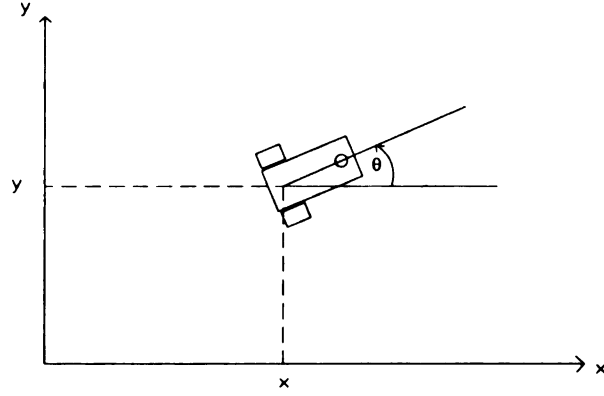


Figure 2.1. Diagram showing the state variables of the mobile robot

by Equation (2.1)

$$\begin{aligned}
 x_1 &= \theta \\
 x_2 &= x \cos \theta + y \sin \theta \\
 x_3 &= x \sin \theta - y \cos \theta \\
 v_1 &= u_2 \\
 v_2 &= u_1 - (x \sin \theta - y \cos \theta) u_2
 \end{aligned} \tag{2.2}$$

It can be easily verified that the transformed kinematics takes the chained form [27]

$$\dot{x}_1 = v_1 \tag{2.3a}$$

$$\dot{x}_2 = v_2 \tag{2.3b}$$

$$\dot{x}_3 = x_2 v_1 \tag{2.3c}$$

The task of converging the mobile robot from its present coordinates to the desired coordinates is equivalent to the task of converging (x_1, x_2, x_3) to $(0, 0, 0)$.

2.3 Almost-Smooth Time-Invariant Feedback Control Strategy

2.3.1 Transformation of variables

For the purpose of controller design, we first transform the state and control variables of the robot model in Equation (2.3). By treating the variables x_1 and x_2 in Equation (2.3) as Cartesian coordinates, we transform them into polar coordinates r and ψ as follows

$$\begin{aligned} x_1 &= r \cos \psi \\ x_2 &= r \sin \psi, \quad r \geq 0 \end{aligned} \tag{2.4}$$

The inverse transformation from $(r, \psi) \rightarrow (x_1, x_2)$ given by Equation (2.4) is defined everywhere. The forward transformation is undefined at the point $(x_1, x_2) = (0, 0)$, where $r = 0$. At all other points, the forward transformation is defined as

$$\begin{aligned} r &= \sqrt{x_1^2 + x_2^2} \\ \psi &= \arctan(x_2/x_1), \quad r \neq 0 \end{aligned} \tag{2.5}$$

By differentiating Equation (2.4) and using Equation (2.3) we get

$$\begin{aligned} v_1 &= \dot{x}_1 = \dot{r} \cos \psi - r \sin \psi \dot{\psi} \\ v_2 &= \dot{x}_2 = \dot{r} \sin \psi + r \cos \psi \dot{\psi} \end{aligned} \tag{2.6}$$

Substitution of Equations (2.4) and (2.6) into Equation (2.3c) results in

$$\dot{x}_3 = \eta_1(r, \psi)\dot{r} + \eta_2(r, \psi)\dot{\psi} \tag{2.7}$$

where $\eta_1 = r \sin \psi \cos \psi$

$$\eta_2 = -r^2 \sin^2 \psi$$

We transform x_3 in Equation (2.3) into a new variable, which we denote by β , using the relation

$$\beta = x_3 - \int_0^r \eta_1 dr = (x_3 - \frac{r^2}{2} \sin \psi \cos \psi) = (x_3 - \frac{1}{2} x_1 x_2) \quad (2.8)$$

Two important ideas behind the definition of β are that, first, $x_3 \rightarrow 0$ when $(r, \beta) \rightarrow 0$, and, second, $\dot{\beta}$ is a function only of $\dot{\psi}$, not \dot{r} . In the transformed coordinates, the kinematics of the robot takes the form

$$\dot{r} = w_1 \quad (2.9a)$$

$$\dot{\psi} = w_2 \quad (2.9b)$$

$$\dot{\beta} = -(r^2/2)w_2 \quad (2.9c)$$

where Equation (2.9c) was obtained by differentiating Equation (2.8) and substituting Equation (2.7). The convergence of the mobile robot to its desired posture now refers to the convergence of r and β to zero. Indeed, from Equations (2.5) and (2.8), $(r, \beta) = (0, 0)$ implies $(x_1, x_2, x_3) = (0, 0, 0)$.

2.3.2 The feedback control strategy

The feedback strategy for the kinematic model of Equation (2.9) is proposed as

$$w_1 = \alpha r (\sigma^2 \beta^2 - r^2) \quad (2.10)$$

$$w_2 = \gamma \beta \quad (2.11)$$

where it is assumed that

$$r(0) \neq 0 \quad (2.12)$$

and where α , γ and σ are positive controller parameters, chosen according to the relation

$$\gamma > 2\alpha\sigma^2 \quad (2.13)$$

The controls for the robot, v_1 and v_2 defined in Equation (2.3), are related to the control variables w_1 and w_2 in Equation (2.9) by Equation (2.6). The control inputs for the asymptotic convergence of the states of the robot are therefore proposed as

$$v_1 = \begin{cases} r[\alpha(\sigma^2\beta^2 - r^2)\cos\psi - \gamma\beta\sin\psi] & \text{if } r > 0 \\ 0 & \text{if } r = 0 \end{cases} \quad (2.14a)$$

$$v_2 = \begin{cases} r[\alpha(\sigma^2\beta^2 - r^2)\sin\psi + \gamma\beta\cos\psi] & \text{if } r > 0 \\ 0 & \text{if } r = 0 \end{cases} \quad (2.14b)$$

The controller proposed in Equation (2.14) cannot stabilize the states of the system to the origin from points where $r(0) = 0$, or $x_1(0) = x_2(0) = 0$. Nevertheless, system states arbitrarily close to, but not exactly on, the manifold of equilibria $r = 0$ can be asymptotically converged to the equilibrium state. This is explained in the next remark, and proved in section 2.3.3.

Remark:

The assumption in Equation (2.12) is reasonable since $r(0)$ cannot be exactly equal to zero. For the system defined by Equations (2.9a) and (2.10), it is apparent that $r = 0$ is an unstable equilibrium point for $\beta \neq 0$. Therefore, the assumption in Equation (2.12) ensures that the system trajectories do not remain stuck at this unstable equilibrium point.

Remark:

The controller in Equation (2.14) is (i) defined everywhere on R^3 , (ii) smooth on the open (and dense) subset $R^3 \setminus \{r = 0\}$ of R^3 , and (iii) continuous at $r = 0$. Therefore,

they are almost-smooth [13] functions of the states of the mobile robot. Other controllers [1, 2], using non-smooth transformations do not satisfy the third condition. The controller in Equation (2.14) is also time invariant, as seen from Equations (2.5) and (2.8).

2.3.3 Convergence and stability

In this section, we establish that the mobile robot converges to its desired posture $(x_1, x_2, x_3) = (0, 0, 0)$ from every posture satisfying Equation (2.12) for the control inputs defined by Equation (2.14). To this end, we establish that $(r, \beta) \rightarrow (0, 0)$ for the control inputs defined by Equations (2.10) and (2.11). We also show in this section that the desired posture $(x_1, x_2, x_3) = (0, 0, 0)$ is stable.

Theorem: (Asymptotic convergence to the desired posture)

Under the control law given by Equation (2.10) and (2.11), the mobile robot described by Equation (2.9) will converge asymptotically from any initial posture satisfying $r(0) \neq 0$, to the desired posture. Specifically $(r, \beta) \rightarrow (0, 0)$ as $t \rightarrow \infty$.

Proof:

Substituting Equation (2.10) and (2.11) into Equation (2.9), the dynamics of (r, β) are seen to be

$$\dot{r} = \alpha r (\sigma^2 \beta^2 - r^2) \tag{2.15a}$$

$$\dot{\beta} = -\left(\frac{\gamma}{2}\right) r^2 \beta \tag{2.15b}$$

It is apparent from Equation (2.15a) that $r = 0$ represents a set of unstable equilibria for any fixed $\beta \neq 0$. Therefore, unless $(r, \beta) = (0, 0)$, the robot can not practically

stay at $r = 0$. Let us therefore assume that $r(0) \neq 0$ and show that $(r, \beta) \rightarrow (0, 0)$ as $t \rightarrow \infty$.

Consider the state space X in (r, β) to be $X = X_0 \cup X_+$, $X_0 = (0, 0)$, $X_+ = \{(r, \beta) : r > 0\}$. In this space, define the Lyapunov function V in the following way. Pick $c \geq 0$ and let $(c, 0)$ be a point on the r axis in the $(r, \sigma\beta)$ half plane, as shown in figure 2.2. Draw a circle that is symmetric about the r axis and passes through $(0, 0)$ and $(c, 0)$. Let V be defined such that its level set $V = c$ is the circle, *i.e.*

$$\left(\frac{V}{2}\right)^2 = \left(r - \frac{V}{2}\right)^2 + \sigma^2\beta^2 \quad (2.16)$$

Clearly, $V \geq 0$ in X and $V = 0$ implies $(r, \beta) = (0, 0)$. Furthermore, if $\|(r, \beta)\| \rightarrow$

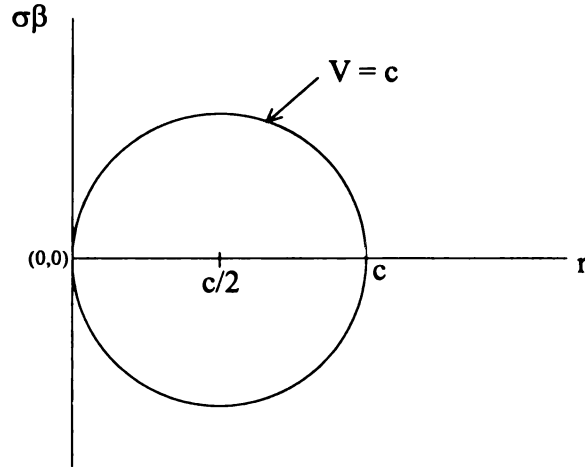


Figure 2.2. Circle in the $r - \sigma\beta$ plane defining the Lyapunov function V .

∞ , $V \rightarrow \infty$. Therefore, V is positive definite and radially unbounded. Note however that, when $r = 0$, V is only defined for $\beta = 0$ since Equation (2.16) becomes $(V/2)^2 = (V/2)^2 + \sigma^2\beta^2$. This is not a problem since V is defined in the set X .

Before we calculate the derivative of V , we rewrite Equation (2.16) in the following two forms

$$r^2 - rV = -\sigma^2\beta^2 \quad (2.17a)$$

$$2r^2 - rV = r^2 - \sigma^2\beta^2 \quad (2.17b)$$

Differentiating Equation (2.17a) with respect to time, substituting Equations (2.15) and (2.17b), and rearranging terms yields

$$\begin{aligned} r\dot{V} &= 2r\dot{r} - \dot{r}V + 2\sigma^2\beta\dot{\beta} \\ &= \alpha(2r^2 - rV)(\sigma^2\beta^2 - r^2) - \gamma\sigma^2r^2\beta^2 \\ &= -\alpha(\sigma^2\beta^2 - r^2) - \gamma\sigma^2r^2\beta^2 \end{aligned} \quad (2.18)$$

Hence $r\dot{V}$ is negative definite in X and is equal to zero at $(r, \beta) = (0, 0)$. Since V is positive definite and radially unbounded in X , and \dot{V} is negative definite in X , we conclude that for all $(r(0), \beta(0)) \in X$, $V \rightarrow 0$ as $t \rightarrow \infty$. This implies that $(r, \beta) \rightarrow (0, 0)$ as $t \rightarrow \infty$.

Remark:

The above theorem establishes asymptotic convergence of the mobile robot to its desired posture $(r, \beta) = (0, 0)$ or $(x_1, x_2, x_3) = (0, 0, 0)$. Even though \dot{V} in Equation(2.18) is negative definite, it does not claim stability of the desired posture because V was defined in the subset X , $X = X_0 \cup X_+$, $X_0 = (0, 0)$, $X_+ = \{(r, \beta) : r > 0\}$, of the posture space. In the complete space of robot postures, we claim stability of the desired posture with the help of the corollary below.

Corollary: (Stability of the desired posture)

Consider the dynamics of the mobile robot described by Equation (2.3) and (2.14). The robot posture $(x_1, x_2, x_3) = (0, 0, 0)$ is a uniformly stable equilibrium point.

Proof:

Define the Lyapunov function candidate

$$V' = \frac{1}{2} \left(x_1^2 + x_2^2 + \left(x_3 - \frac{x_1 x_2}{2} \right)^2 \right) \quad (2.19)$$

Clearly $V' \geq 0$ and $V' = 0$ implies $(x_1, x_2, x_3) = (0, 0, 0)$. Therefore, V' is globally positive definite. Using Equation (2.3c), (2.4), (2.8), (2.13) and (2.14), the derivative of V' can be computed as

$$\begin{aligned} \dot{V}' &= x_1 \dot{x}_1 + x_2 \dot{x}_2 + \left(x_3 - \frac{x_1 x_2}{2} \right) \left(\dot{x}_3 - \frac{x_1}{2} \dot{x}_2 - \frac{x_2}{2} \dot{x}_1 \right) \\ &= (x_1 v_1 + x_2 v_2) + \frac{\beta}{2} (x_2 v_1 - x_1 v_2) \\ &= \alpha r^2 (\sigma^2 \beta^2 - r^2) - (\gamma/2) r^2 \beta^2 \\ &= -\alpha r \left(r^2 + \left(\frac{\gamma}{2\alpha} - \sigma^2 \right) \beta^2 \right) \\ &\leq 0 \end{aligned} \quad (2.20)$$

The uniform stability of $(x_1, x_2, x_3) = (0, 0, 0)$ follows from the negative semi-definiteness of \dot{V}' .

Remark:

The control strategy in this section uses the nonlinear oscillator in polar coordinates

$$\dot{r} = \mu r (nW - r^2) \quad (2.21a)$$

$$\dot{\psi} = \omega \quad (2.21b)$$

where $W \geq 0$ is a function of the states of the robot. The oscillator frequency, denoted by $\dot{\psi}$, is a variable that converges to zero. This is evident from Equation (2.21) and (2.11) and the fact that β converges to zero. As a result, the phase of the oscillator, ψ , converges to some value, which is not of importance.

2.4 Simulation Results

Simulation results are presented here to demonstrate the capability of the controller to converge the mobile robot from different initial configurations to the origin. The initial configuration of the robot are given in table 2.1 and simulation results are shown in figures 2.3 -2.7. Controller parameters for each simulation are given in table 2.2. Of particular interest is Case B, where $r(0) = 0$. This violates the assumption in equation (2.12). This problem is remedied by assuming the initial orientation to be 0.01 degrees instead of exactly zero.

Table 2.1. Initial configuration of mobile robot

	x	y	θ
Case A	1	1	0
Case B	0	1	0
Case C	0	1	180
Case D	-1	0	180
Case E	2	2	0,90,180,-90

Table 2.2. Controller parameter used in simulations of mobile robot

	α	γ	σ
Case A	0.02	1	1
Case B (a)	0.02	1	3.8
Case B (b)	0.02	1	11
Case C	0.02	1	1
Case D	0.02	1	8
Case E	0.02	1	1

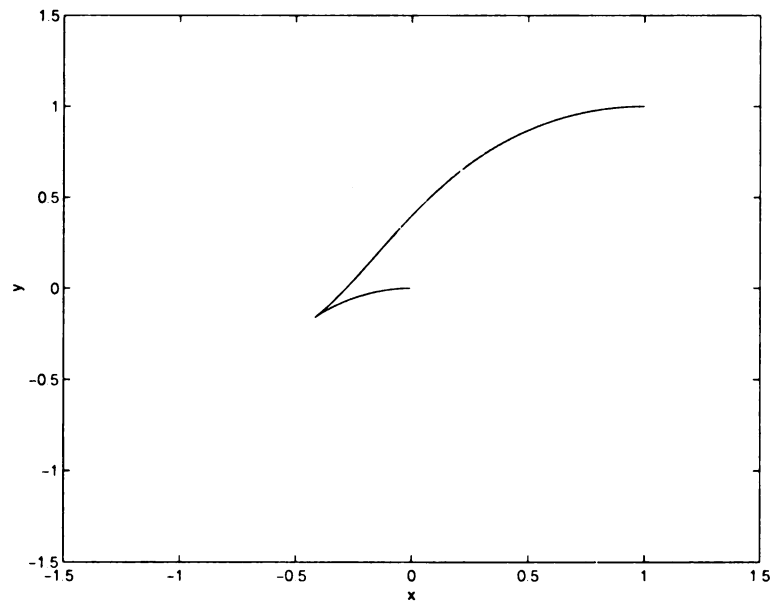


Figure 2.3. Trajectory of the mobile robot for Case A.

Figur
param

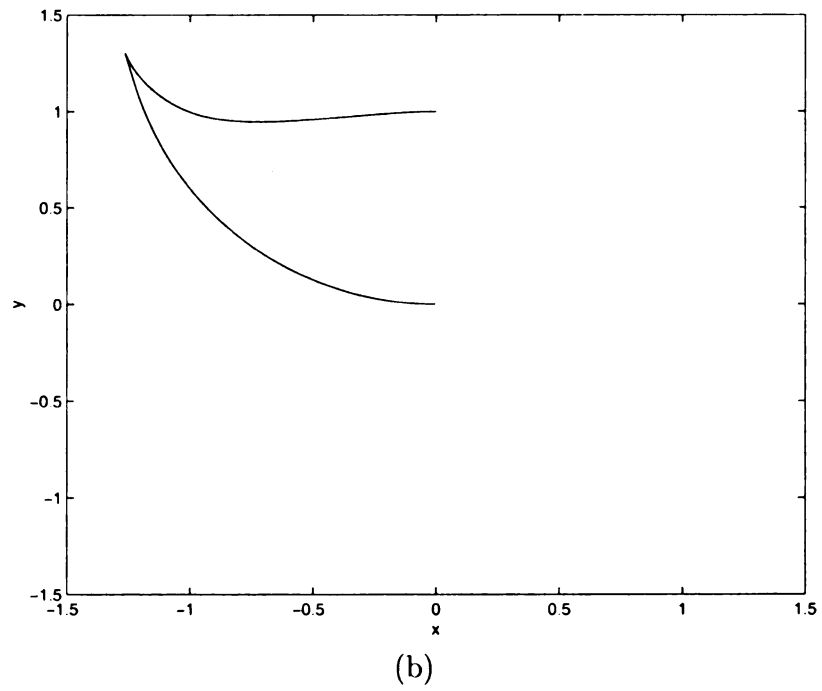
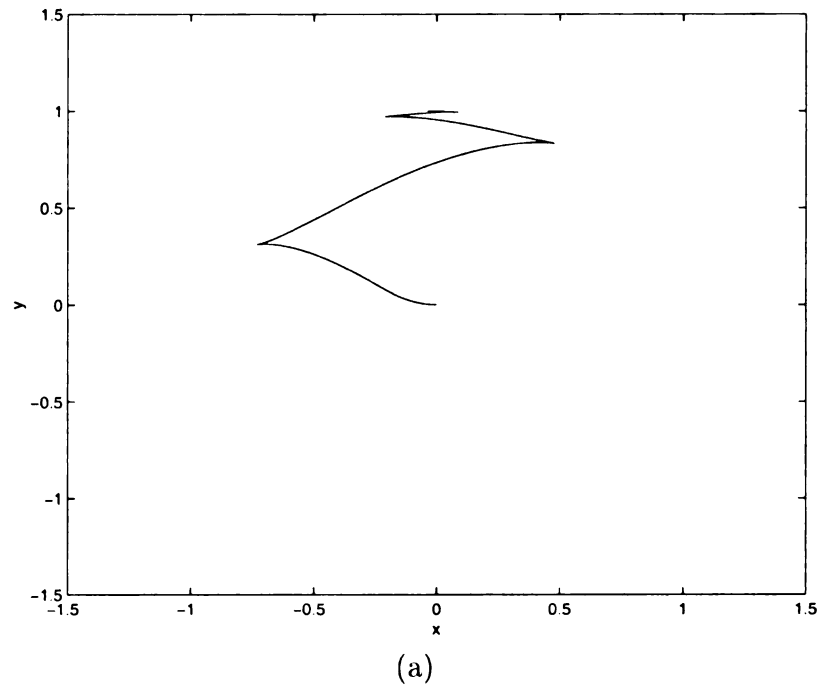


Figure 2.4. Trajectories of the mobile robot for Case B with different set of control parameter.

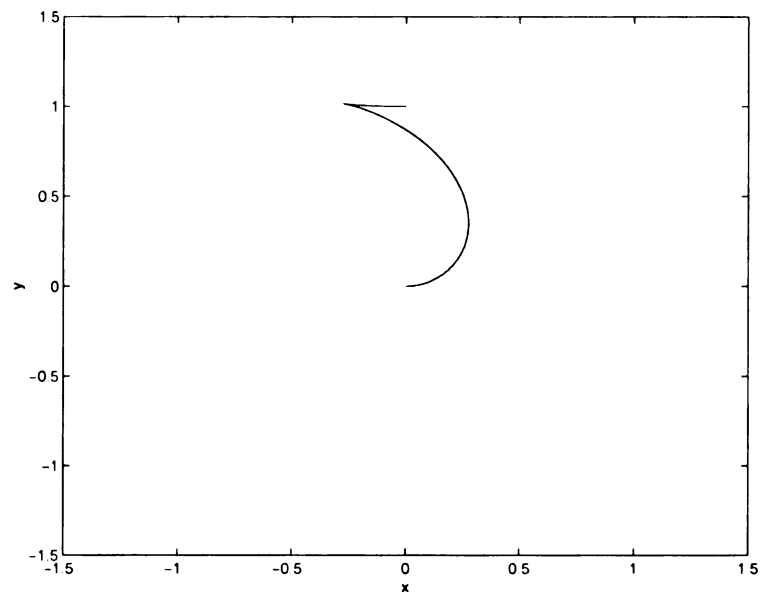


Figure 2.5. Trajectory of the mobile robot for Case C.

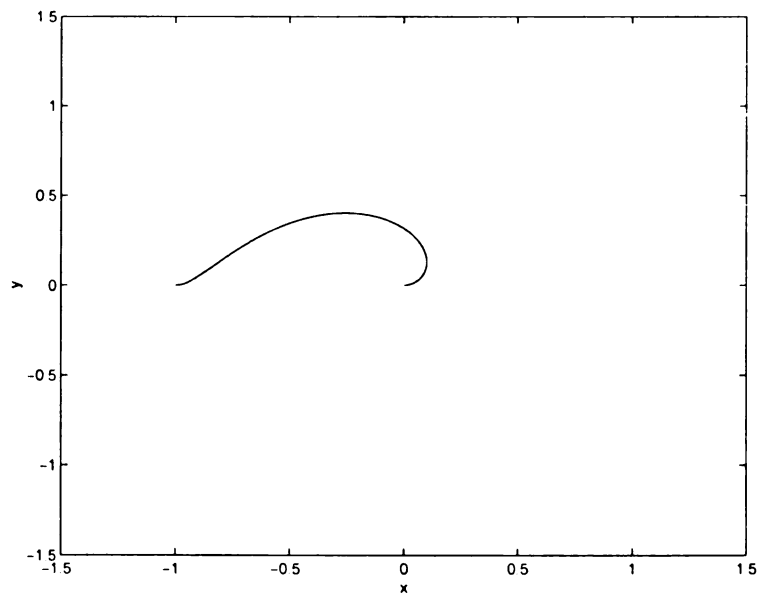


Figure 2.6. Trajectory of the mobile robot for Case D.

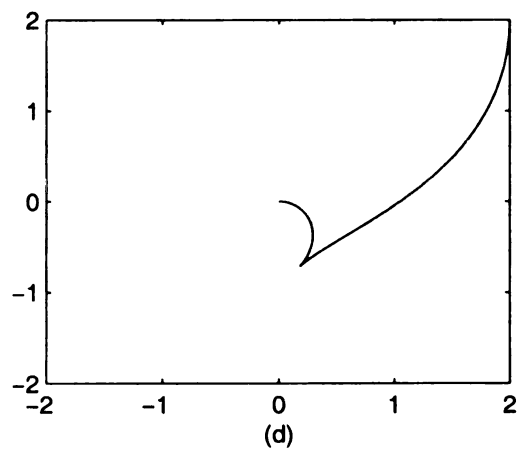
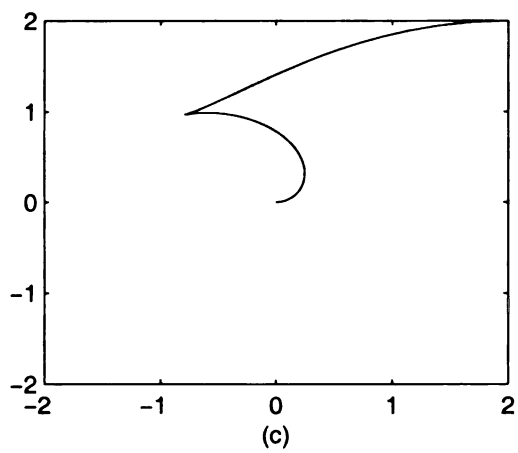
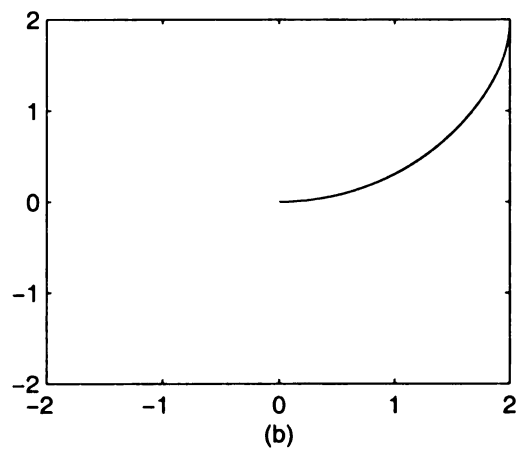
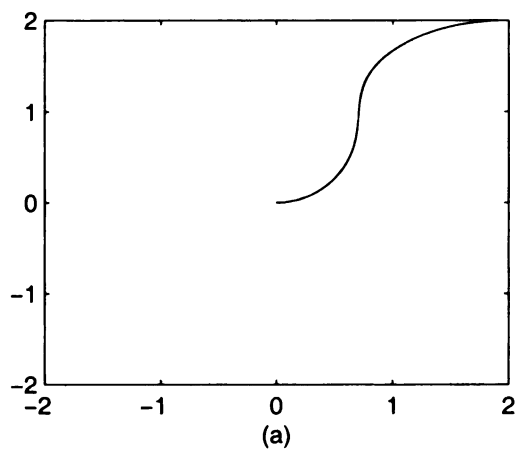


Figure 2.7. Trajectory of the mobile robot for Case E.

CHAPTER 3

Open-Loop Control of The Rolling Sphere

3.1 Motivation

This chapter is motivated by the idea of the spherical robot. The spherical robot has been conceived as a mobile robot with a spherical exoskeleton and an internal mechanism for self-propulsion. It will also have retractable manipulators and/or limbs that can be deployed through ports in the exoskeleton after the robot comes to rest and can be retracted before the robot resumes its motion. It will also have a retractable camera that can be deployed and turned to record an image around the robot. With its unique structure, this robot will have a number of advantages over conventional mobile robots and have a wide range of applications. For example, with a fire-resistant exoskeleton, the robot can be useful in fire-fighting and rescue missions. It can be used for inspection in hazardous environments, such as in chemical factories and nuclear power plants. In every application, the superlative locomotion capabilities of the robot will be an added advantage. The spherical exoskeleton will provide the robot with the maximum stability. It will be able to move over rough terrain with relative ease. This robot will be an ideal candidate for space exploration.

It can also find significant military applications. In addition to reconnaissance, it can be used for terrorist countermeasures or as a robotic soldier in the line of fire.

In this thesis, an open-loop controller is presented for the spherical robot. We will focus on the robot with a retractable camera, which, to be effective, must be deployed from the top. Given an arbitrary point on the sphere, the basic idea behind the open-loop strategy is to design a trajectory such that the point comes to the top of the sphere when the sphere is commanded to a certain coordinate on the x - y plane. Our open loop control strategy is based on the concept of geometric phase applied to spherical triangular paths on the surface of the sphere. Later, the trajectories will be generalized to arbitrary closed curves from triangular paths.

3.2 Review of Spherical Trigonometry

When a plane intersects with a sphere, it produces a circle. If the intersecting plane passes through the center of the sphere, the circle is called a great circle. Any angle on the sphere formed by two intersecting arcs of great circles is called a spherical angle. The portion of the sphere bounded by the arcs of three great circles is called a spherical triangle. Three bounding arcs are called the sides and three vertices of each spherical angle are the vertices of the spherical triangle. We shall designate the vertices by P, Q, R , spherical angles by A, B, C and the corresponding opposite sides by a, b, c respectively. Each angle of the spherical triangle is measured by the dihedral angles made by the plane defining its sides. The spherical triangle is shown in figure 3.1. In general, we shall consider only those spherical triangles of which each side and angle is less than π . The important relations between the sides and angles of the spherical triangle are

1. The sum of the angles of a spherical triangle is greater than 180° and less than 540° ; that is, $180^\circ < A + B + C < 540^\circ$.

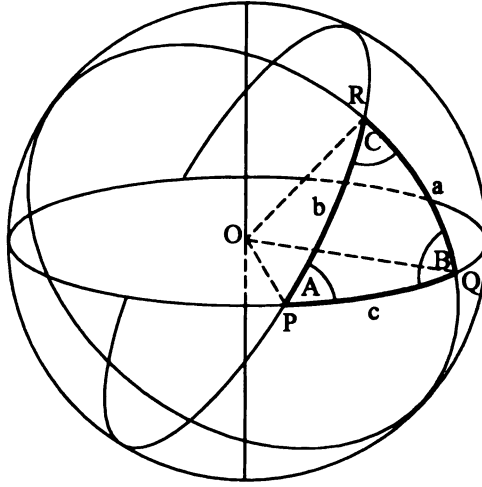


Figure 3.1. A spherical triangle

2. If two angles of a spherical triangle are equal, the opposite sides are equal; and conversely.
3. If two angles of a spherical triangle are unequal, the opposite sides are unequal, and the greater side lies opposite the greater angle; and conversely.
4. The sum of any two sides of a spherical triangle is greater than the third side.

Some basic formulas that relate sides and angles of a spherical triangle are

Law of sines: In any spherical triangle ABC ,

$$\frac{\sin a}{\sin A} = \frac{\sin b}{\sin B} = \frac{\sin c}{\sin C} \quad (3.1)$$

Law of cosines for sides: In any spherical triangle ABC ,

$$\begin{aligned} \cos a &= \cos b \cos c + \sin b \sin c \cos A \\ \cos b &= \cos c \cos a + \sin c \sin a \cos B \\ \cos c &= \cos a \cos b + \sin a \sin b \cos C \end{aligned} \quad (3.2)$$

Law of cosines for angles: In any spherical triangle ABC ,

$$\begin{aligned}\cos A &= -\cos B \cos C + \sin B \sin C \cos a \\ \cos B &= -\cos C \cos A + \sin C \sin A \cos b \\ \cos C &= -\cos A \cos B + \sin A \sin B \cos c\end{aligned}\tag{3.3}$$

Some useful relations that can be derived from the law of cosines of sides, are

$$\begin{aligned}\sin a \cos B &= \sin c \cos b - \sin b \cos c \cos A \\ \sin b \cos C &= \sin a \cos c - \sin c \cos a \cos B \\ \sin c \cos A &= \sin b \cos a - \sin a \cos b \cos C\end{aligned}\tag{3.4}$$

$$\begin{aligned}\sin A \cos b &= \sin C \cos B + \sin B \cos C \cos a \\ \sin B \cos c &= \sin A \cos C + \sin C \cos A \cos b \\ \sin C \cos a &= \sin B \cos A + \sin A \cos B \cos c\end{aligned}\tag{3.5}$$

$$\begin{aligned}\sin a \cos C &= \sin b \cos c - \sin c \cos b \cos A \\ \sin b \cos A &= \sin c \cos a - \sin a \cos c \cos B \\ \sin c \cos B &= \sin a \cos b - \sin b \cos a \cos C\end{aligned}\tag{3.6}$$

$$\begin{aligned}\sin A \cos c &= \sin B \cos C + \sin C \cos B \cos a \\ \sin B \cos a &= \sin C \cos A + \sin A \cos C \cos b \\ \sin C \cos b &= \sin A \cos B + \sin B \cos A \cos c\end{aligned}\tag{3.7}$$

The der

etry. su

The

angles c

where

In gen

three

the ki

analo

Giv

Th

an

th

tw

The derivation of these formulas can be found in most books on spherical trigonometry, such as [3, 16, 34].

The area of a spherical triangle is proportional to the excess of the sum of its three angles over the sum of two right angles, or

$$\text{Area of Spherical Triangle } ABC = (A + B + C - \pi)r^2 = Er^2 \quad (3.8)$$

where $E = (A + B + C - \pi)$ is the spherical excess and r is the radius of the sphere. In general, if any three of the sides and angles a, b, c, A, B, C are given, the remaining three can be found. In this thesis, we will consider solving the spherical triangle with the knowledge of two sides (a, b) and the included angle (C). Deriving from Napier's analogies [16], the formulas for solving the spherical triangle are

$$\frac{\cos \frac{1}{2}(A - B)}{\cos \frac{1}{2}(A + B)} = \frac{\tan \frac{1}{2}(a + b)}{\tan \frac{1}{2}c} \quad (3.9)$$

$$\frac{\sin \frac{1}{2}(a - b)}{\sin \frac{1}{2}(a + b)} = \frac{\tan \frac{1}{2}(A - B)}{\cot \frac{1}{2}C} \quad (3.10)$$

$$\frac{\cos \frac{1}{2}(a - b)}{\cos \frac{1}{2}(a + b)} = \frac{\tan \frac{1}{2}(A + B)}{\cot \frac{1}{2}C} \quad (3.11)$$

Given two sides and the included angle, a spherical triangle is uniquely determined. The sum and the difference of the other two angles can be found using formulas (3.10) and (3.11). Then the two angles can be found from the sum and the difference. The third side can be found from formula (3.9). In other words, given a, b and C , the other two angles A, B and the third side c can be obtained from the sequence of equations

$$\begin{aligned} (A - B) &= 2 \arctan\left(\frac{\sin \frac{1}{2}(a - b)}{\sin \frac{1}{2}(a + b) \tan \frac{1}{2}C}\right) \\ (A + B) &= 2 \arctan\left(\frac{\cos \frac{1}{2}(a - b)}{\cos \frac{1}{2}(a + b) \tan \frac{1}{2}C}\right) \end{aligned}$$

$$A = \frac{(A - B) + (A + B)}{2} \quad (3.12)$$

$$B = (A + B) - A$$

$$c = 2 \arctan\left(\frac{\tan \frac{1}{2}(a + b) \cos \frac{1}{2}(A + B)}{\cos \frac{1}{2}(A - B)}\right)$$

3.3 Sphere rolling along the spherical triangle

In this section, basic trigonometric identities and coordinate transformations are used to compute the final location and orientation of a sphere resulting from following a path specified by its own spherical triangle. Since a spherical triangle is a closed path on the surface of the sphere, the point of contact with the ground after rolling is the same point as the one before rolling. It implies that the sphere will undergo some amount of rotation about the vertical axis passing through this point of contact and the center of the sphere. Although the spherical triangle is a closed path on the sphere, it does not generate a closed path on the ground. Our goal is to find the final location, and the rotation of the sphere in terms of the dimensions of the spherical triangle. This will enable us to plan the open-loop control strategy for the spherical robot, using the concept of geometric phase.

Let us suppose that the center of the sphere in figure 3.2 is at the origin (point 1) of the xyz coordinate frame in figure 3.3. Also suppose that the point Q on the sphere (figure 3.2) is at the bottom of the sphere. To roll along the spherical triangle PQR in figure 3.2, the sphere is first rolled by the distance of a in the negative y direction. This brings the sphere to position 2 in figure 3.3 and the point R in figure 3.2 is now at the bottom of the sphere. The sphere is then rolled by the distance of b in the direction angled C from the previous direction. This brings the sphere to point 3 in figure 3.3. Finally, the sphere is rolled by the distance of c in the direction angled A from the previous direction. This bring the sphere to point 4 in figure 3.3. The final

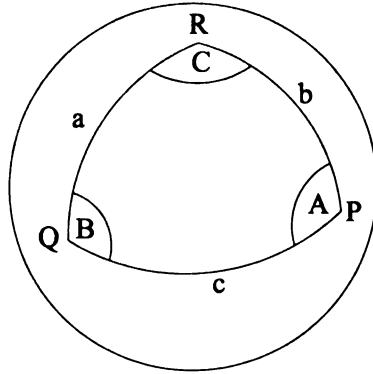


Figure 3.2. The spherical triangle that specify the path in figure 3.3

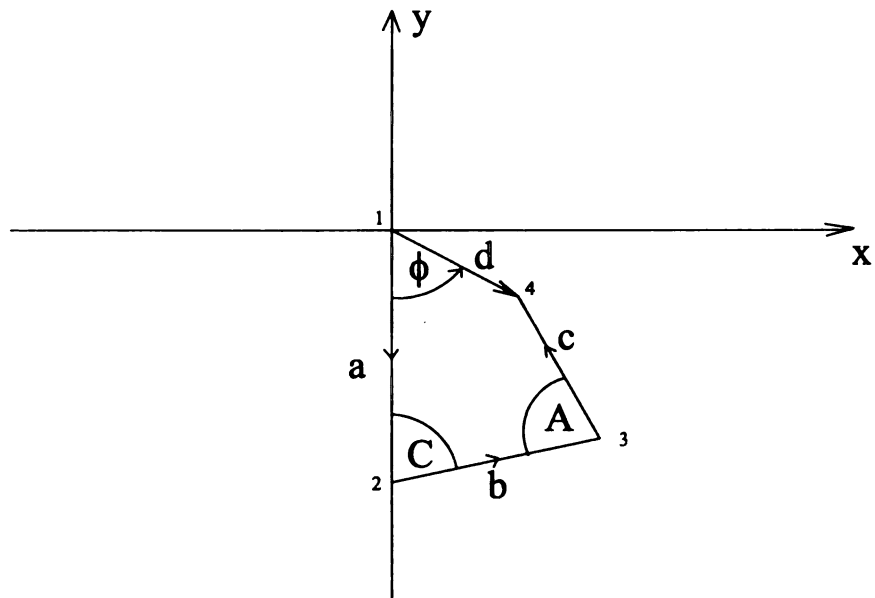


Figure 3.3. Motion of the sphere defined by the spherical triangle in figure 3.2

position

o. The

of 272.

and o c

figure 3

There

tr

R

to

w

sy

the

cal

coor

position of the sphere in the x - y plane is represented by the distance d and the angle ϕ . The distance d is measured from the starting position, or, in our case, the origin of xyz . The angle ϕ is measured clockwise from the first direction of rolling. Both d and ϕ can be found in terms of the sides and angles of the spherical triangle PQR in figure 3.3, using basic trigonometry. Following the path in figure 3.3, we get

$$d_x = b \sin C - c \sin(A + C)$$

$$d_y = a - b \cos C + c \cos(A + C)$$

Therefore,

$$d^2 = a^2 + b^2 + c^2 + 2ac \cos(A + C) - 2ab \cos C - 2ab \cos A \quad (3.13)$$

$$\phi = \arctan \left(\frac{b \sin C - c \sin(A + C)}{a - b \cos C + c \cos(A + C)} \right) \quad (3.14)$$

Now, consider amount of rotation of the sphere. Since rolling along the spherical triangle is equivalent to a pure rotation about the vertical axis, the rotation matrix R , which relate the original coordinate and final coordinate of the sphere, is expected to be in the form of

$$\begin{bmatrix} \hat{i}' \\ \hat{j}' \\ \hat{k}' \end{bmatrix} = \begin{bmatrix} & & \\ & R & \\ & & \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix} \quad (3.15)$$

where α is the amount of rotation, $\hat{i}, \hat{j}, \hat{k}$ are unit vectors along the fixed reference system, and $\hat{i}', \hat{j}', \hat{k}'$ are unit vectors along the sphere fixed coordinate system. At the initial time $\hat{i}', \hat{j}', \hat{k}'$ coincide with $\hat{i}, \hat{j}, \hat{k}$, respectively. We use vector rotation to calculate the rotation matrix R . Recall that each row of matrix R is the rotated coordinate represented in the original coordinate system. For example, the first row

of matrix R is the \hat{i}' coordinate represented in the original coordinate system $\hat{i}\hat{j}\hat{k}$. Hence, we can use vector rotation to find each individual \hat{i}', \hat{j}' or \hat{k}' in terms of \hat{i}, \hat{j} and \hat{k} and, then, determine the entries of the rotation matrix R .

From vector analysis, rotation of a vector r about a vector l by an angle of α results in the vector r' , defined by [14]

$$r' = (1 - \cos \alpha)(l \cdot r)l + \cos \alpha \cdot r + \sin \alpha(l \times r) \quad (3.16)$$

For convenience, we assume the sphere to have unit radius. The results can however be extended to a sphere of radius other than unity. We have assumed that the sphere-fixed coordinates described by the unit vectors $\hat{i}', \hat{j}', \hat{k}'$ are initially coincides with the reference coordinates $\hat{i}, \hat{j}, \hat{k}$. Therefore, rolling of the sphere along side a of the spherical triangle, or from point 1 to point 2, in figure 3.3, on the ground is equal to a rotation of the sphere fixed axis about i by the angle of a . Rolling from point 2 to point 3 on the ground, as seen in Figure 3.3, is equal to a rotation about the axis $[\cos C\hat{i} - \sin C\hat{j}]$ by the angle of $-b$. Finally, rolling from point 3 to point 4 is equal to the rotation about the axis $[\cos(A + C)\hat{i} - \sin(A + C)\hat{j}]$ by the angle of c . The calculation of vector rotation is done using Mathematica. The code is shown in Appendix A. Using spherical triangle identities (Appendix B) to simplify the results from Mathematica, the rotation matrix becomes (Appendix C)

$$R = \begin{bmatrix} -\cos(A + B + C) & \sin(A + B + C) & 0 \\ -\sin(A + B + C) & -\cos(A + B + C) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

or

$$R = \begin{bmatrix} \cos(\pi - (A + B + C)) & \sin(\pi - (A + B + C)) & 0 \\ -\sin(\pi - (A + B + C)) & \cos(\pi - (A + B + C)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Comparing the matrix R in equations (3.15) and (3.18), the amount of rotation of the sphere about the vertical axis is determined to be

$$\alpha = \pi - (A + B + C) \quad (3.19)$$

From equation (3.8), we can now say that by following the spherical triangle path in figure 3.3, the coordinate of the sphere rotates about z axis by an angle equal to negative of the area of the spherical triangle, assuming that the radius of the sphere is unity.

$$\alpha = \pi - (A + B + C) = -(\text{Area of the Spherical Triangle}) \quad (3.20)$$

If we consider a sphere of radius r , $r \neq 1$, the analysis presented above changes marginally. When the sphere of radius r rolls from point 1 to point 2 on the ground, the coordinates of the sphere rotate about the i axis by the angle (a/r) . Rolling from point 2 to point 3 is equal to a rotation of the coordinates about the axis $[\cos C\hat{i} - \sin C\hat{j}]$ by the angle $(-b/r)$. Rolling from point 3 to point 4 is equivalent to rotation of the coordinates about the axis $[\cos(A + C)\hat{i} - \sin(A + C)\hat{j}]$ by an angle (c/r) . In summary, for a sphere of radius r , the set of angles used in our analysis of coordinate rotation (Appendix A and C) is changed from a, b, c, A, B, C to $(a/r), (b/r), (c/r), A, B, C$, respectively. Note that angles A, B and C are not changed and the spherical triangle identities in Appendix B still hold good. Since A, B and C are unchanged, the only change in Appendix C is affected by replacing a, b and c with $(a/r), (b/r)$ and (c/r) , respectively. At the end, the rotation matrix resulting from simplification is the same as the matrix in equation (3.17) since it depends only

on A , B and C . The amount of rotation can still be expressed as in equation (3.19). However, by comparing with the area of the spherical triangle in equation (3.8), we can state that for a sphere of radius r , the rotation about the z axis is equal to

$$\alpha = \pi - (A + B + C) = - \left(\frac{\text{Area of the Spherical Triangle}}{r^2} \right) \quad (3.21)$$

One may question the amount of rotation if the sphere were to roll to the right instead of rolling to the left, as the path in figure 3.4. Considering figures 3.3 and

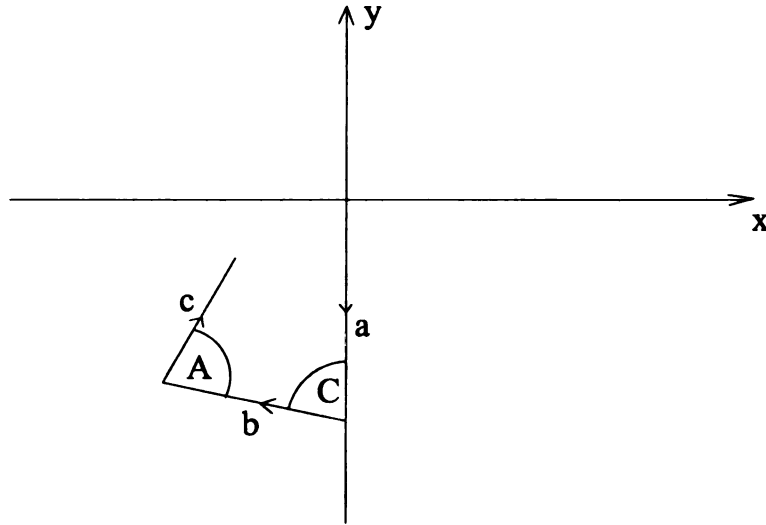


Figure 3.4. Path generated by the spherical triangle similar to the one in figure 3.3 but with opposite direction.

3.4, it is not hard to see that the analysis of the path in figure 3.4 is similar to the one in figure 3.3 except that the spherical angles are negative of each other. In other words, we have to use $-A$, $-B$ and $-C$ as the spherical angles in the rotation matrix

for the path in figure 3.4. The rotation matrix R in equation (3.17) would become

$$R = \begin{bmatrix} -\cos(-A - B - C) & \sin(-A - B - C) & 0 \\ -\sin(-A - B - C) & -\cos(-A - B - C) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

or

$$R = \begin{bmatrix} \cos((A + B + C) - \pi) & \sin((A + B + C) - \pi) & 0 \\ -\sin((A + B + C) - \pi) & \cos((A + B + C) - \pi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

For the path in figure 3.4, the rotation of the sphere is equal to the area of the spherical triangle divided by square of the radius, or

$$\alpha = A + B + C - \pi = \frac{\text{Area of the Spherical Triangle}}{r^2} \quad (3.24)$$

The path in figure 3.4 will be referred to as a right-handed path, since, to an observer within the sphere, the sphere always turns to the right. The path in figure 3.3 is a left-handed path.

In summary, when a sphere rolls along the path specified by its own spherical triangle, the final location of the sphere can be calculated from equation (3.13) and (3.14). If the path is right-handed, the amount of rotation of the sphere about the vertical axis is equal to the area of the spherical triangle divided by the square of the radius of the sphere. If the path is left-handed, the amount of rotation is negative of the area of spherical triangle divided by the square of the radius of the sphere.

3.4 Sphere rolling along lune

In this section, we consider the effect of rotation of the sphere along a closed path given by the two sides of a lune. A lune [34], as shown in figure 3.5, is the portion of the surface of a sphere, which is comprised between two great circles. Two sides of a

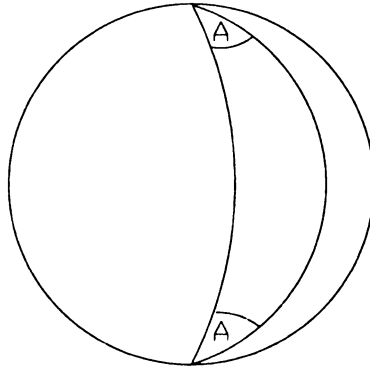


Figure 3.5. The Lune

lune make a closed path on the surface of a sphere. Both sides have equal length of πr . Both angles, which are formed by the two great circles, are equal. The area of a lune is proportional to its angle and is given as

$$\text{Area of a Lune} = 2Ar^2 \quad (3.25)$$

where A is the angle of the lune. The cartesian path generated by a sphere rolling along a lune is shown in figure 3.6. First the sphere rolls by the distance of πr . Then it changes the direction by the angle of A and rolls for the distance of πr . The coordinate of the sphere at position 2 in figure 3.6 can be found easily. Assume that the coordinates of the sphere at location 1, $\hat{i}_1\hat{j}_1\hat{k}_1$, coincide with the reference

coordina

coordina

To find

vecto

coordinate, $\hat{i}\hat{j}\hat{k}$. Since the coordinates rotate by an angle π about the x axis, the coordinates of the sphere at point 2 in figure 3.6 are

$$\hat{i}_2 = \hat{i}_1 = \hat{i}$$

$$\hat{j}_2 = -\hat{j}_1 = -\hat{j}$$

$$\hat{k}_2 = -\hat{k}_1 = -\hat{k}$$

To find the coordinates of the sphere at final position (3 in figure 3.6) we again use

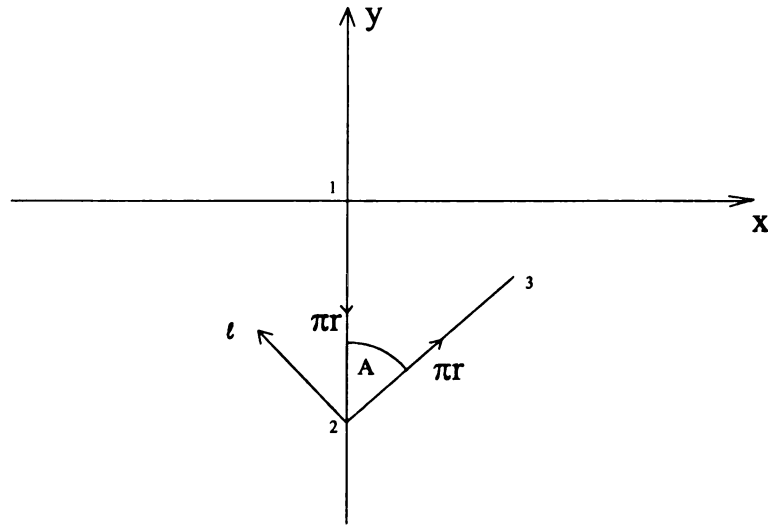


Figure 3.6. Cartesian path given by the Lune in figure 3.5

vector rotation in equation (3.16). The axis of rotation is

$$l = -\cos A \hat{i} + \sin A \hat{j} \quad (3.26)$$

Therefore,

$$\begin{aligned}
\hat{i}_3 &= \cos(2A)\hat{i} - \sin(2A)\hat{j} \\
\hat{j}_3 &= \sin(2A)\hat{i} + \cos(2A)\hat{j} \\
\hat{k}_3 &= \hat{k}
\end{aligned} \tag{3.27}$$

The relation between the coordinates $\hat{i}_3\hat{j}_3\hat{k}_3$ and $\hat{i}\hat{j}\hat{k}$ can be written in matrix form as

$$\begin{bmatrix} \hat{i}_3 \\ \hat{j}_3 \\ \hat{k}_3 \end{bmatrix} = \begin{bmatrix} & & \\ & R & \\ & & \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix} \tag{3.28}$$

where the rotation matrix

$$R = \begin{bmatrix} \cos 2A & -\sin 2A & 0 \\ \sin 2A & \cos 2A & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(-2A) & \sin(-2A) & 0 \\ -\sin(-2A) & \cos(-2A) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.29}$$

Comparing equations (3.15) and (3.25), the amount of rotation of the sphere is given by

$$\alpha = -2A = -\left(\frac{\text{Area of a lune}}{r^2}\right) \tag{3.30}$$

Note that this amount of rotation is derived from the left-handed path in figure 3.6. Similar to the spherical triangle, if the path is right-handed, the amount of rotation about the vertical axis is equal to the area of the lune divided by the square of radius. For a right-handed path, we have

$$\alpha = 2A = \frac{\text{Area of a lune}}{r^2} \tag{3.31}$$

3.5 Sphere rolling along an arbitrary closed path

The results of section 3.3 indicate that when a sphere of unit radius rolls along a spherical triangle, the net rotation of the sphere is equal to the area of the spherical triangle. The direction of rotation is positive for a right-handed path and negative for a left-handed path. The result for a two-sided curve in section 3.4 is similar, the amount of rotation is equal to the surface area enclosed by the two sided curve. In this section, we attempt to generalize this result to arbitrary closed paths on the sphere. First consider the simple spherical polygon in figure 3.7. This polygon 12345 can be divided into three spherical triangles by the arcs 13 and 14. The final orientation

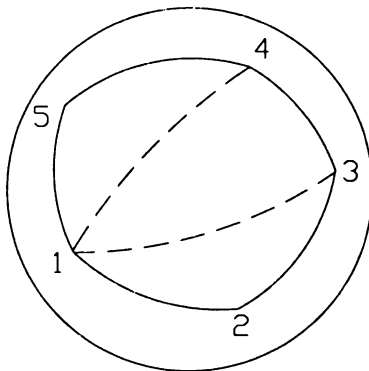


Figure 3.7. Arbitrary Spherical Polygon

of the sphere rolling along the path 123451 is the same as rolling along the path 1231341451 since paths 313 and 414 do not change the orientation of the sphere. Path 1231341451 is composed of three spherical triangular paths. When following each spherical triangle, the sphere rotates by an amount equal to the area of that spherical triangle. All rotations are positive since the spherical triangles are right-

ha
re
at
if

handed according to the definition in section 3.3. Clearly, the effect of the sphere rolling along the spherical polygon is a rotation by an amount equal to the surface area of the spherical polygon. If the path is right-handed, the rotation is positive and if it is left-handed, the rotation is negative.

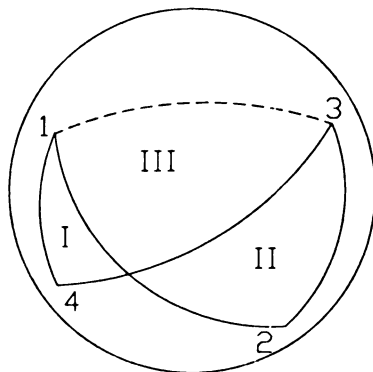


Figure 3.8. Arbitrary closed path that crosses itself

Now, consider the path in figure 3.8 that crosses itself. We can represent such paths by two consecutive spherical triangular paths. First, the sphere rolls along the spherical triangle path 1231 and then path 1341. The right-handed path 1231 results in the rotation of the sphere by an amount equal to the sum of area II and III. The left-handed path 1341 cause the sphere to rotate by an amount equal to the sum of area I + III in the negative direction. Thus, the sphere rolling along either path 12341 or 1231341 has the amount of rotation equal to the difference of areas II and I. For an arbitrary path that crosses itself, the surface area bounded by the path can be divided into two areas at the cross point. The amount of rotation of the sphere rolling along the arbitrary path equals the difference of the two areas. The direction of rotation depends on the direction of the path. If the area is bounded by

the right-handed path is larger than the one which is left-handed, the total rotation is positive. If area bounded by left-handed path is larger, the net rotation is negative.

Any closed curve on the surface of a sphere can be always represented by an n -sided polygon, where n is a large number. Therefore, it can be deduced that the amount of rotation of the sphere after rolling along any closed curve is equal to the surface area enclosed by the curve on the sphere.

3.6 Kinematic Model of Sphere

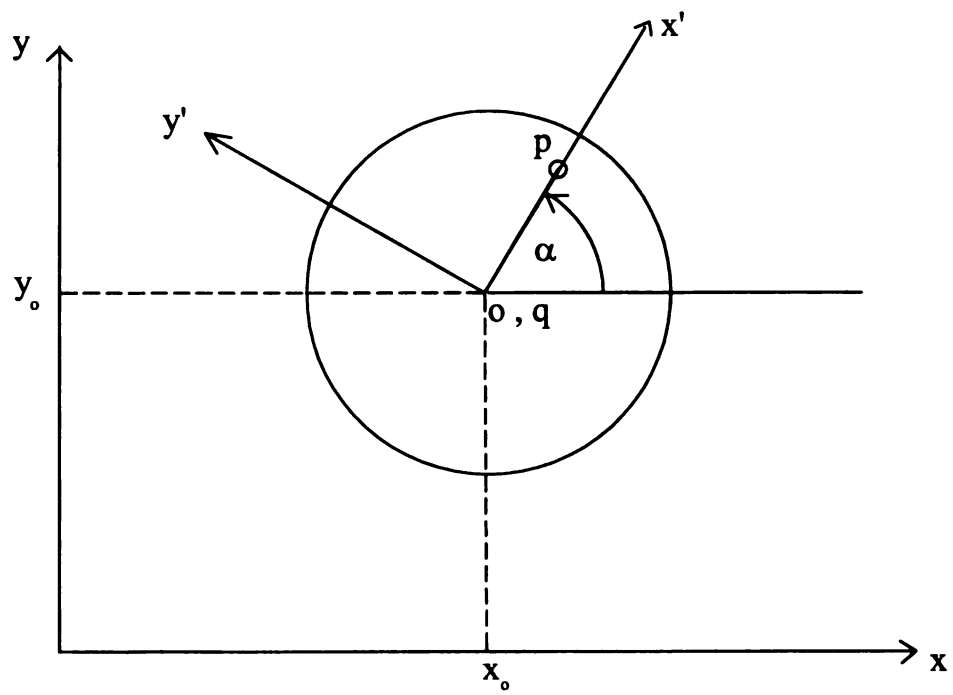
In view of the specific control problem we have defined for the sphere, namely control of the sphere to a certain x - y coordinate such that a certain point on the surface of the sphere appears on the top, we propose the kinematic model in terms of four state variables. Two of these states include the two angles (θ and α) that represent a certain point (p) on the surface of the sphere. The other two states denote the coordinates of the center of the sphere (x_0 and y_0). The variables are shown in figure 3.9. The angle α is measured from the x axis on the x - y plane as shown in figure 3.9(a) and θ is measured from the vertical axis in the x' - z plane as shown in figure 3.9(b). The two control inputs are the angular velocities in x and y direction, namely (ω_x and ω_y). The kinematic equations for x_0 and y_0 are

$$\dot{x}_0 = r\omega_y \quad (3.32)$$

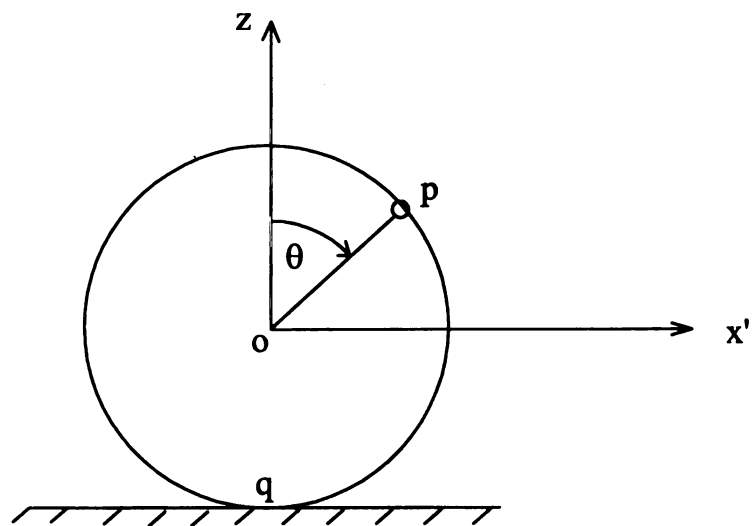
$$\dot{y}_0 = -r\omega_x \quad (3.33)$$

where r is the radius of the sphere. The derivative of θ with respect to time is the angular velocity in the y' direction. Therefore, we have

$$\dot{\theta} = -\omega_x \sin \alpha + \omega_y \cos \alpha \quad (3.34)$$



(a)



(b)

Figure 3.9. Figure defining the state variables of a sphere

The differential equation for α can be found from the velocity of point p , namely

$$\dot{\alpha} = -\cot \theta \omega'_x = -\cot \theta (\omega_x \cos \alpha + \omega_y \sin \alpha) \quad (3.35)$$

The kinematic equation for α in equation (3.35) has the problem of singularity at $\theta = 0$. This is unavoidable since α is undefined at $\theta = 0$. However, with equation (3.35), the range of α is not limited within $-\pi$ to π . Therefore, to find α , we track point p and, from relative position of point p with the center of the sphere, we compute α . Consider figure 3.9(b), since velocity of point q is zero, the velocity of point p can be written in vector form as

$$\vec{V}_p = \vec{\omega} \times \vec{R}_{p/q}$$

where $\vec{\omega}$ is the angular velocity of the sphere

$$\vec{\omega} = \omega_x \hat{i} + \omega_y \hat{j}$$

and $\vec{R}_{p/q}$ is the position vector from point q to point p

$$\vec{R}_{p/q} = (r \sin \theta \cos \alpha) \hat{i} + (r \sin \theta \sin \alpha) \hat{j} + r(1 + \cos \theta) \hat{k} \quad (3.36)$$

Thus,

$$\vec{V}_p = \omega_y r(1 + \cos \theta) \hat{i} - \omega_x r(1 + \cos \theta) \hat{j} + (\omega_x r \sin \theta \sin \alpha - \omega_y r \sin \theta \cos \alpha) \hat{k} \quad (3.37)$$

Therefore, the velocity of point p in x and y directions are

$$\dot{x}_p = \omega_y r(1 + \cos \theta) \quad (3.38)$$

$$\dot{y}_p = -\omega_x r(1 + \cos \theta) \quad (3.39)$$

At any instant of time, the angle α can be found from the relation

$$\alpha = \arctan \left(\frac{y_p - y_o}{x_p - x_o} \right) \quad (3.40)$$

3.7 Formulation of Open Loop strategy

In this section, we formulate an open loop control strategy that can lead the sphere to the desired final position and orientation. This contributes positioning the sphere at the origin of the x - y coordinate, in figure 3.10, and locating point P at the top of

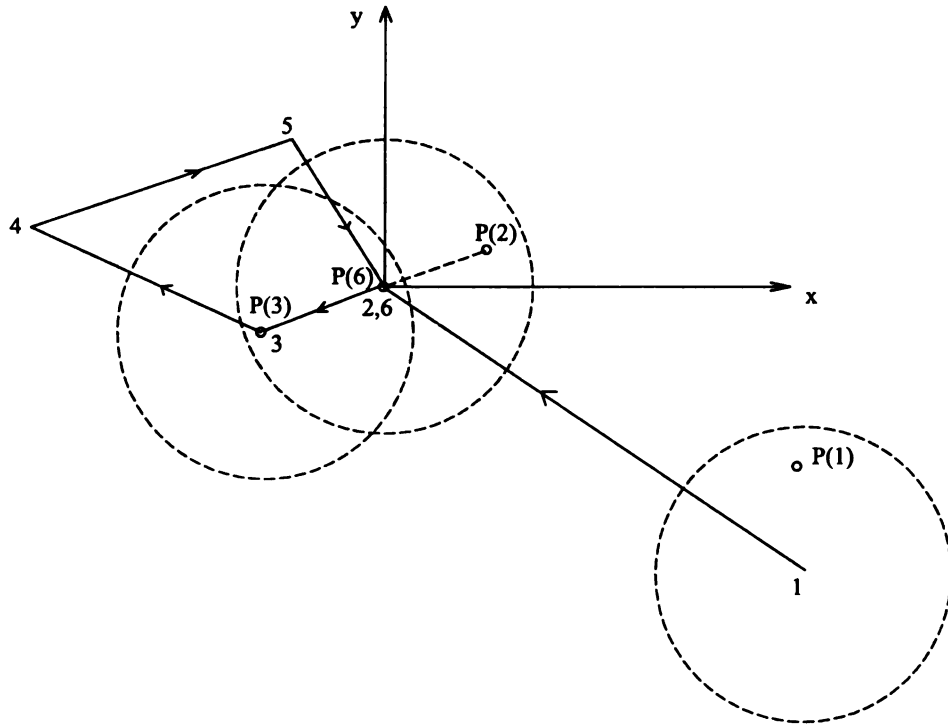


Figure 3.10. Open loop control strategy of spherical robot

the sphere. It can also include an arbitrary orientation of the sphere with point P on

the top. At the initial configuration of the sphere, shown by point 1 in figure 3.10, point P is arbitrary located on the sphere, $P(1)$. The control strategy consists of three main steps. First, the sphere rolls from the initial configuration to the origin of the $x-y$ coordinate frame, denoted by the straight line path from point 1 to point 2 in figure 3.10. Point P is then located at a position other than the top of the sphere, $P(2)$. Second, the sphere rolls away from origin $x-y$ in order to move point P to the top. This is shown by the straight line trajectory from point 2 to point 3. Finally, a spherical triangle is used to transfer the sphere to the desired location, point 6, while providing the proper amount of rotation. At this location, point P is on the top and center of the sphere is back at the origin of $x-y$ frame.

The first two processes are trivial to implement. Therefore, in this section, we concentrate only on the third step which is the motion along the spherical triangle. The task is to find a spherical triangle that can lead the sphere from stage 3 to stage 6, in figure 3.10, and give the desired amount of rotation to the sphere. We know from section 3.3 that when the sphere roll along the spherical triangle PQR , in figure 3.2, the final position and orientation of the sphere are given by three variables d , ϕ and α . Distance d and angle ϕ represent the final position of the sphere and can be found by solving equations

$$d^2 = a^2 + b^2 + c^2 + 2ac \cos(A + C) - 2ab \cos C - 2ab \cos A \quad (3.41)$$

$$\phi = \arctan \left(\frac{b \sin C - c \sin(A + C)}{a - b \cos C + \cos(A + C)} \right) \quad (3.42)$$

Distance d is measured from the initial position and, with our control strategy, is limited to the maximum value of π for unit radius sphere. ϕ is measured from the direction in which the sphere begin to roll.

The amount of rotation about the vertical axis α can be found by equation (3.21) or (3.24) depending on the direction of rolling. In our simulation, for convenience, we

will assume the sphere to have unit radius. Therefore, the amount of rotation of the sphere rolling along the spherical triangle ABC is

$$\alpha = \pm(A + B + C - \pi) = \pm \text{Area of the spherical triangle} \quad (3.43)$$

The rotation is positive for the right-handed path and negative for left-handed path.

Our task is to find the spherical triangular path that results in the desired final position and orientation of the sphere. In other words, given the value of d , ϕ and α , we need to find sides a and b and angle C of the spherical triangle that satisfy equations (3.41)-(3.43). Along with these equations, the set of equations (3.12) in section 3.2 are also used to find the other three remaining angles A , B and side c of the spherical triangle. To get an idea of achievable values of d , ϕ and α , we compute them by inputting value of a , b and C within their range, 0 to 180° , into equations (3.12) and (3.41)-(3.43). Figure 3.11 show the possible value of d and α . If the desired value of d and α lie outside the dark band, we are not able to find a spherical triangle that satisfy those desired value. A set of spherical triangles will then be needed to satisfy the desired value of d and α . The d and ϕ plot and the α and ϕ plot are shown in figures 3.12 and 3.13, respectively. Recall that ϕ is measured from the direction the sphere begins to roll and we have the freedom to choose this direction, denoted by ψ in figure 3.14. The desired value of ϕ can then lie outside the band in figures 3.12 and 3.13 since it can be corrected by changing the direction the sphere start to roll.

The set of equations:, (3.12) and (3.41)-(3.43), are nonlinear and difficult to solve for a closed form solution of a , b and C in terms of d , ϕ and α . Therefore, we use numerical methods to solve for a , b and C when the desired value of d , ϕ and α are given. The Fortran programs are given in Appendix D.

In the simulations, the magnitude of the angular velocity of the sphere is set

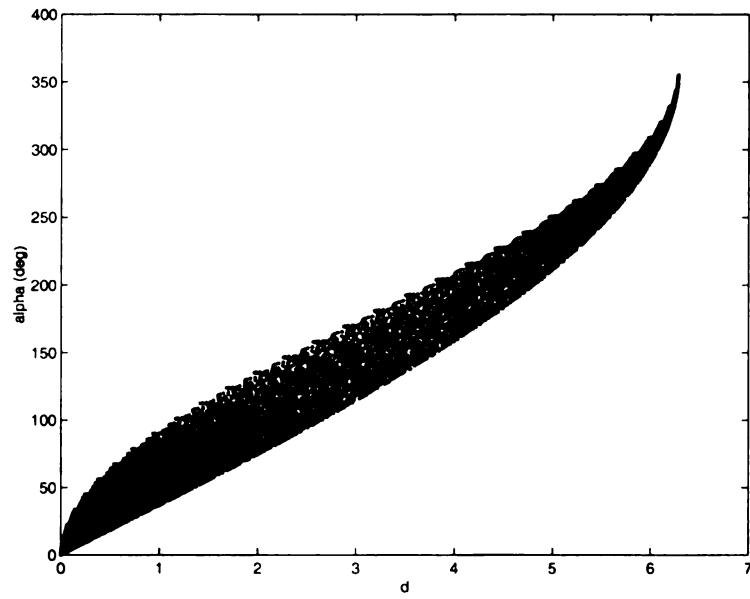


Figure 3.11. Possible values of d and α for one spherical triangle

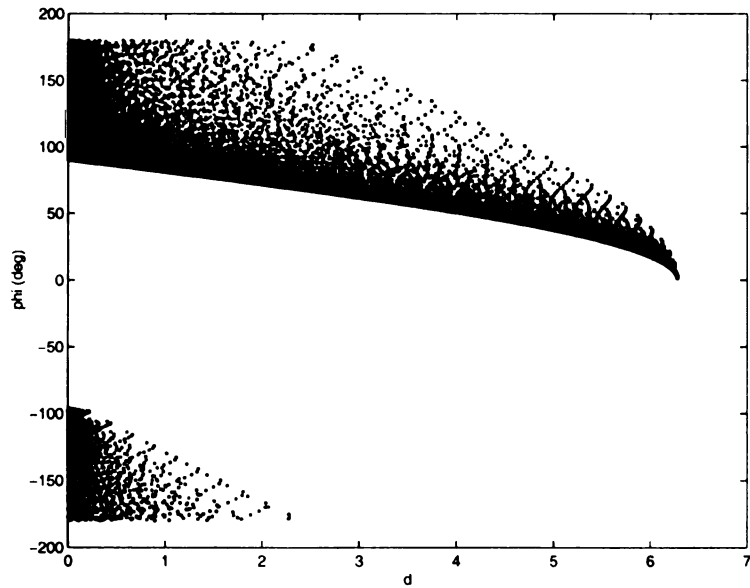


Figure 3.12. Possible values of d and ϕ for one spherical triangle

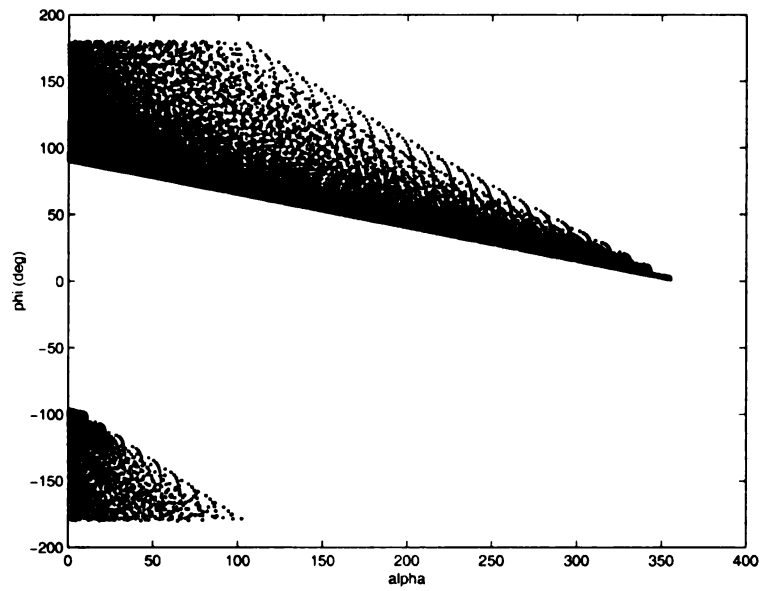


Figure 3.13. Possible values of α and ϕ for one spherical triangle

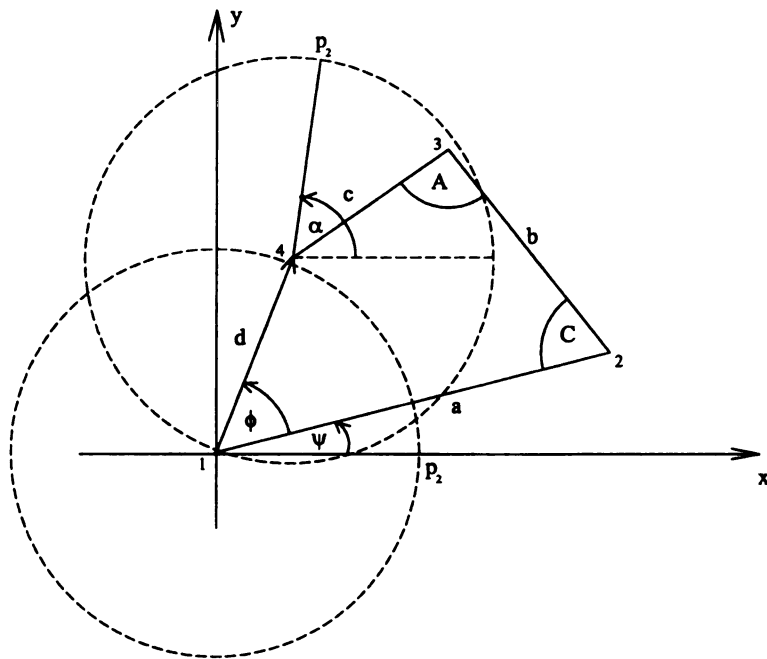


Figure 3.14. System variables used in simulation

to 0.1 rad/sec. We present simulation results of three cases with different desired configurations. The configurations of the sphere for these simulations are shown in table 3.1.

Table 3.1. Numerical values used in simulations 1, 2 and 3 for open loop control of the sphere rolling along spherical triangles.

Parameter	Simulation 1	Simulation 2	Simulation 3
x_d	0.1	1	0.5
y_d	0.5	-1.5	0.5
α_d	-45°	90°	50°
d_d	0.51	1.8	0.7071
ϕ_d	78.69°	-56.3099°	45°
ϕ	100°	80°	90°
ψ	-21.31°	23.6901°	135°
a	1.81022	1.8618	2.27049
b	2.30807	1.19845	2.93917
C	30.1449°	95.4131°	26.1455°
<i>Direction</i>	left-handed	right-handed	right-handed

In the first simulation, the desired location is set at $x_d = 0.1$, $y_d = 0.51$ and $\alpha_d = -45$ degree. This desired location is equivalent to the desired value of $d = 0.5$, $\phi = 78.69^\circ$ and $\alpha = -45^\circ$. From figure 3.11, the desired location lie inside the band. Thus, we can use one spherical triangle to accomplish the task. However, from figure 3.12 and 3.13, the desired location lies outside the dark band. We then used the value ϕ of 100° to solve for the spherical triangle. This modification in ϕ can be corrected by using $\psi = 78.69^\circ - 100^\circ = -21.31^\circ$. Since the desired amount of rotation (α) is negative we use a left-handed path. Two sides and the included angle of the spherical triangle were numerically solved and were found to be $a = 1.81022$, $b = 2.30807$ and $C = 30.1449^\circ$. The trajectory of the sphere is shown in figure 3.15 and the system variables are shown in figures 3.16 and 3.17. In figure 3.16, all variables reach their desired values as expected. Figure 3.17 shows that the point on the top of the sphere

comes back to the top after rolling along the spherical triangle path.

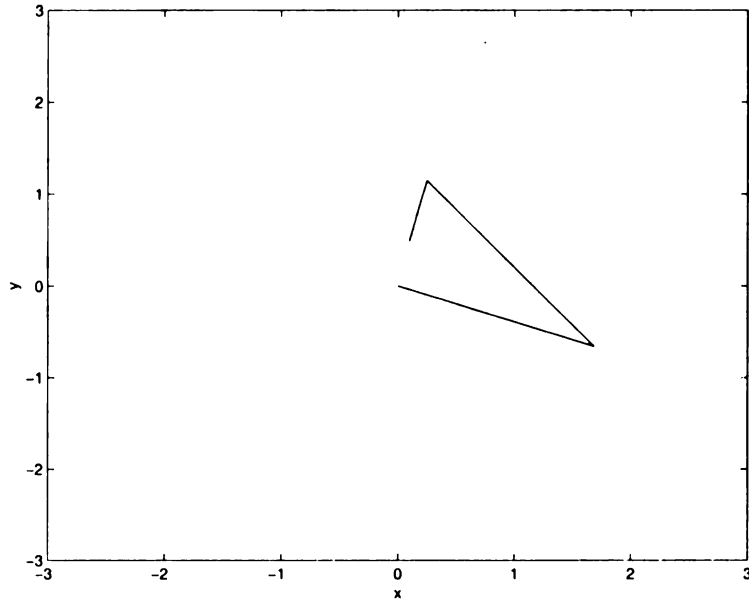


Figure 3.15. Trajectory of the sphere in simulation 1.

In the second simulation, the desired configuration is chosen as $x_d = 1, y_d = -1.5$ and $\alpha_d = 90^\circ$ or, equivalently, $d = 1.8, \phi = -56.3099^\circ$ and $\alpha = 90^\circ$. The spherical triangle is computed by setting $\phi = 80^\circ$. Therefore, the start direction is $\psi = 23.6901^\circ$. Two sides and the included angle of the spherical triangle are computed to be $a = 1.8618, b = 1.19845$ and $C = 95.4131^\circ$. The trajectory is shown in figure 3.18. Figures 3.19 and 3.20 show that all system variables converge to their desired value. The path is right-handed since the desired amount of rotation is positive.

In the third simulation, the desired configuration is set at $x_d = 0.5, y_d = 0.5$ and $\alpha_d = 50^\circ$ or, equivalently, $d = 0.7071, \phi = 45^\circ$ and $\alpha = 50^\circ$. The spherical triangle is computed by setting $\phi = 90^\circ$. Therefore, the start direction, ψ is set at 135° . The three computed parameters of the spherical triangle are $a = 2.27049, b = 2.93917$ and

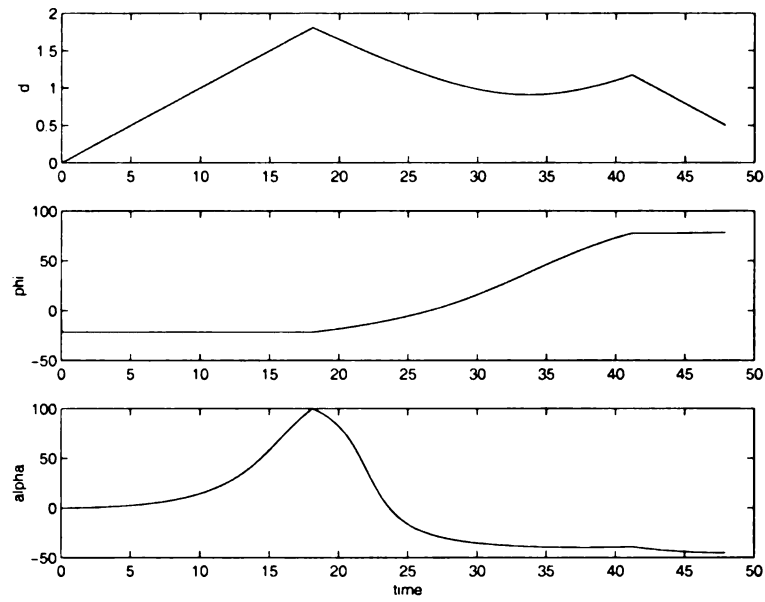


Figure 3.16. Variables d , ϕ and α from simulation 1 for open loop control of sphere.

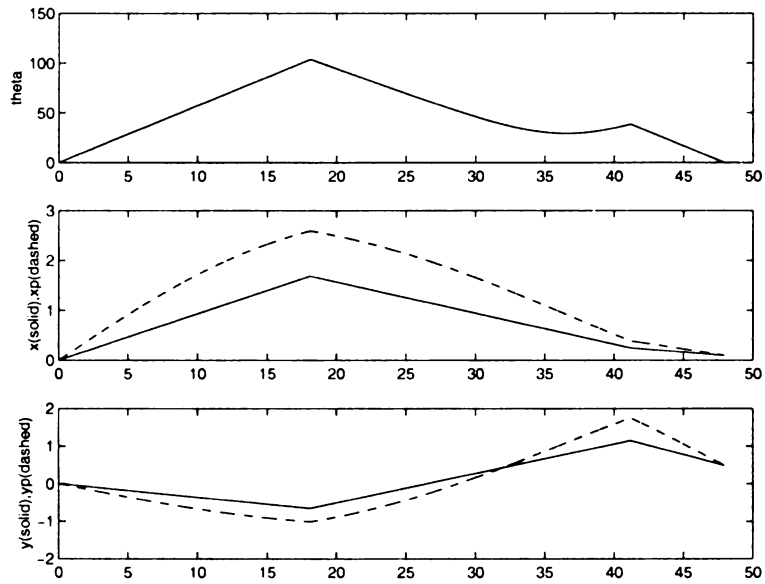


Figure 3.17. Variables θ , x , x_p , y , y_p from simulation 1 for open loop control of sphere.

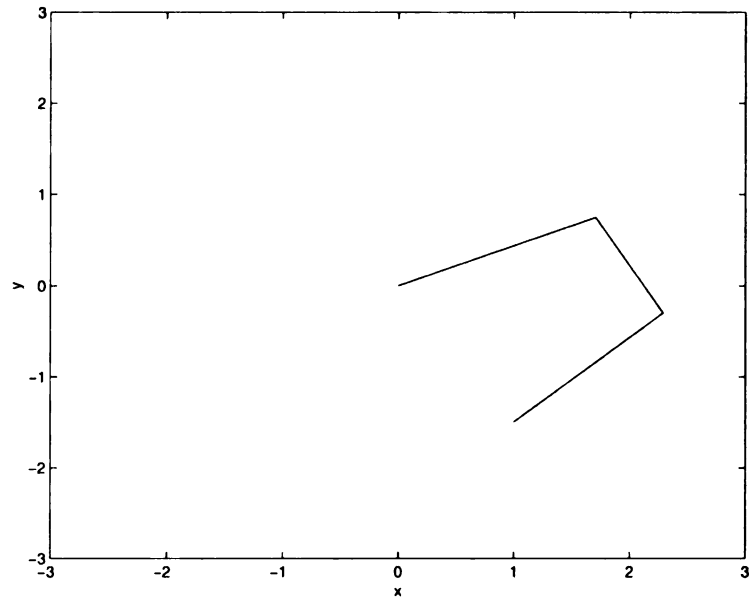


Figure 3.18. Trajectory of the sphere in simulation 2.

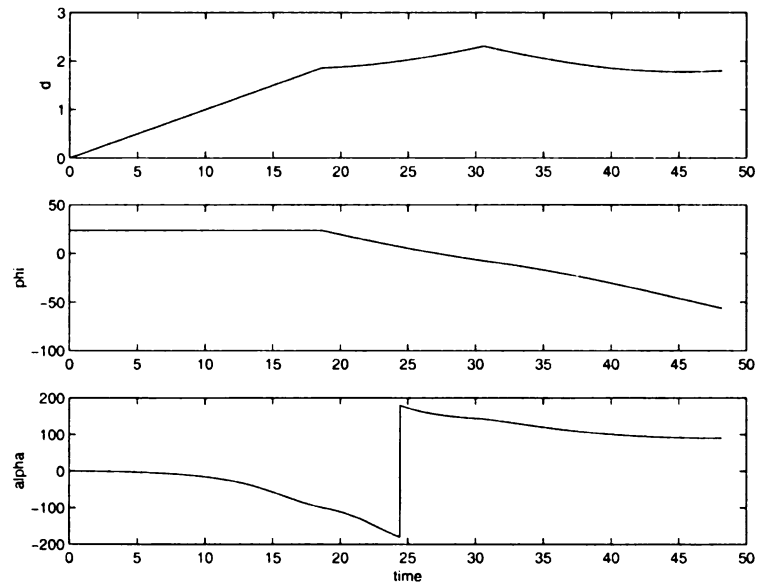


Figure 3.19. Variables d , ϕ and α from simulation 2 for open loop control of sphere.

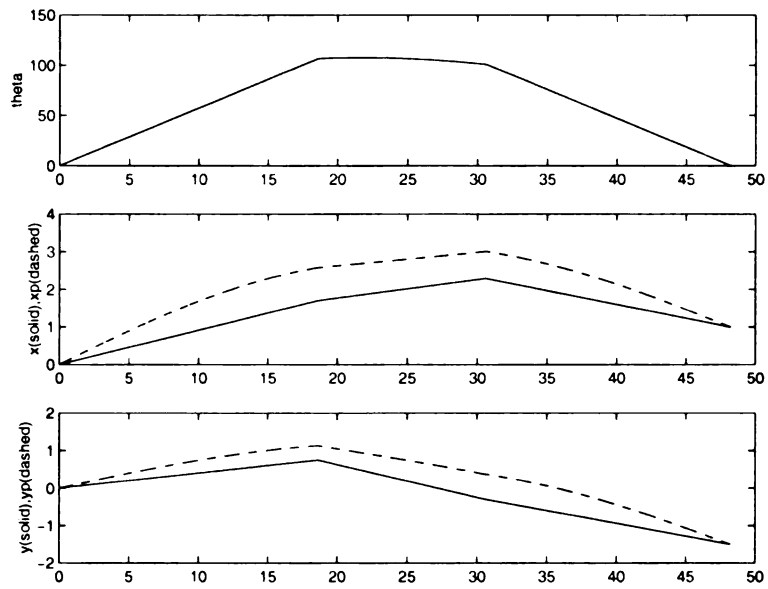


Figure 3.20. Variables θ, x, x_p, y, y_p from simulation 2 for open loop control of sphere.

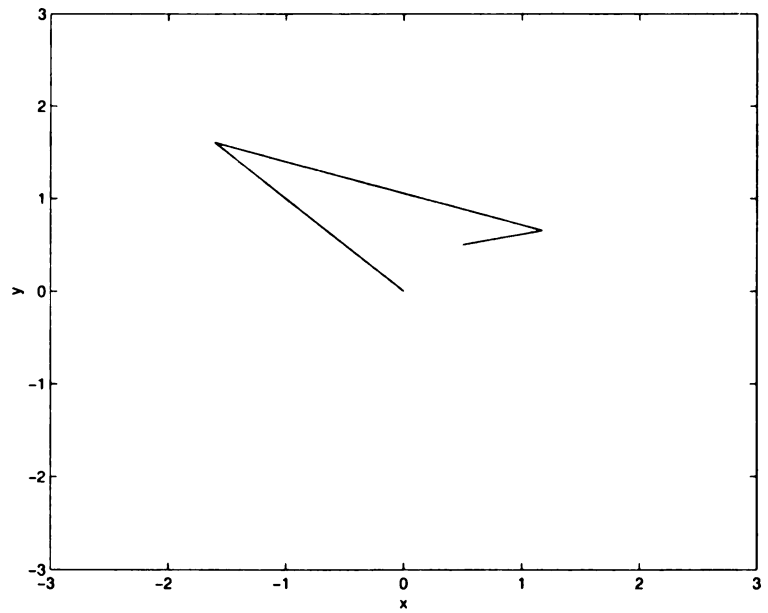


Figure 3.21. Trajectory of the sphere in simulation 3.

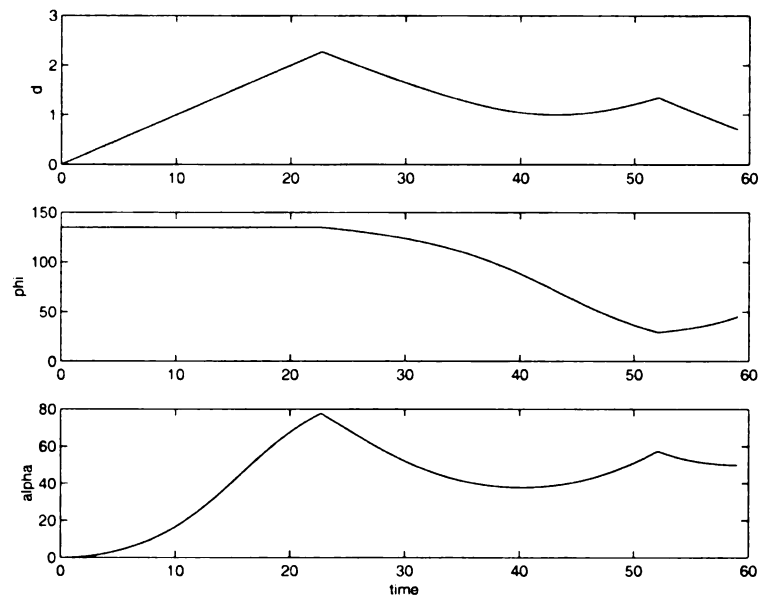


Figure 3.22. Variables d, ϕ and α from simulation 3 for open loop control of sphere.

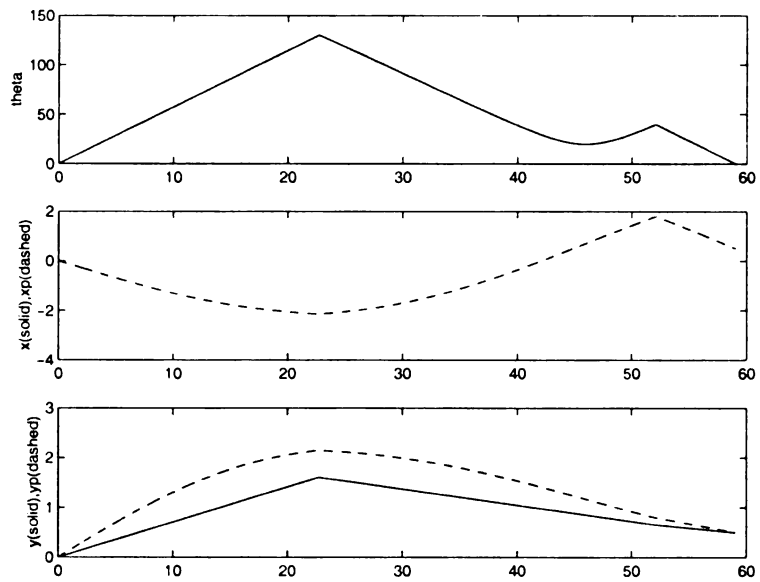


Figure 3.23. Variables θ, x, x_p, y, y_p from simulation 3 for open loop control of sphere.

$C = 26.1455^\circ$. The trajectory is shown in figure 3.21. Figures 3.22 and 3.23 show the plot of all variables with respect to time. Figure 3.22 shows that all variables converge to their desired value. Figure 3.23 shows that point p converges to the top of the sphere. The path is right-handed since the desired amount of rotation is positive.

3.8 Additional simulations supporting open-loop control strategy

In the open loop strategy, we used spherical triangles to plan the trajectory. However, any closed path on the sphere can be used to achieve the same task. While we have proved in section 3.5 that the amount of rotation of the sphere rolling along any closed curve equals to the surface area on the sphere bounded by the closed curve, we present additional simulations to support this claim. We show that the amount of rotation of the sphere rolling along any closed curve on its surface equals to the surface area bounded by the closed curve. To this end, we characterize an arbitrary closed curve by the relation of two angle, $\alpha_a(t)$ and $\theta_a(t)$ in figure 3.24 where

$$\theta_a(0) = \theta_a(t_f) \quad (3.44)$$

where t_f is the final time.

To find the angular velocity that drives the sphere along the given closed curve, we first assume that there is a pendulum attach at the center of the sphere and always hanging downward. When the sphere rolls along the closed curve, the end of this pendulum tracks the closed curve. From an observer inside the sphere, an angular velocity is needed for the pendulum to track the closed curve. This angular velocity

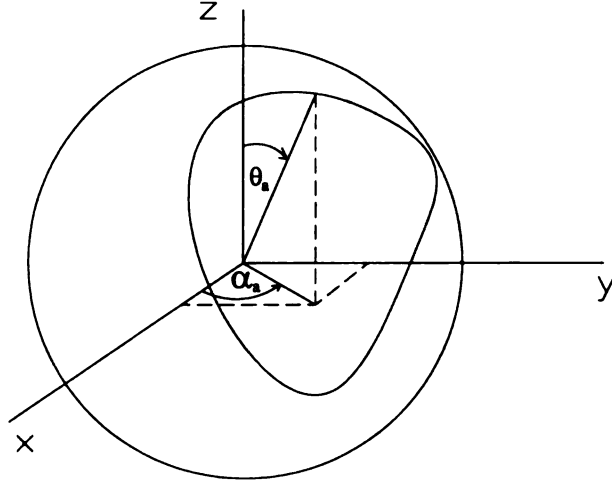


Figure 3.24. Parameters of closed curve

can be written as

$$\vec{\omega}_{pend/sphere} = -\dot{\theta}_a \sin \alpha_a \hat{i}' + \dot{\theta}_a \cos \alpha_a \hat{j}' + \dot{\alpha}_a \hat{k}' \quad (3.45)$$

In the expression above, we use primed coordinates since these coordinates are rotating with the sphere and not fixed inertially. For the sphere to roll such that the pendulum stays hanging down, the angular velocity of the sphere must be negative of that of the pendulum.

$$\vec{\omega}_{sphere} = -\omega_{pend/sphere} = \dot{\theta}_a \sin \alpha_a \hat{i}' - \dot{\theta}_a \cos \alpha_a \hat{j}' - \dot{\alpha}_a \hat{k}' \quad (3.46)$$

However, the input angular velocity of the sphere should be based on the reference coordinate system $\hat{i}\hat{j}\hat{k}$. This coordinate system is different from the coordinate $\hat{i}'\hat{j}'\hat{k}'$ since when the sphere rolls, the coordinate $\hat{i}'\hat{j}'\hat{k}'$ rotates with the sphere. We transform the angular velocity in equation (3.46) into the coordinate system $\hat{i}\hat{j}\hat{k}$ in the following manner. We keep track of coordinates \hat{k}' , with angles α and θ , and \hat{i}' , with angles α_2

and θ_2 . Thus,

$$\hat{k}' = \sin \theta \cos \alpha \hat{i} + \sin \theta \sin \alpha \hat{j} + \cos \theta \hat{k} \quad (3.47a)$$

$$\hat{i}' = \sin \theta_2 \cos \alpha_2 \hat{i} + \sin \theta_2 \sin \alpha_2 \hat{j} + \cos \theta_2 \hat{k} \quad (3.47b)$$

$$\begin{aligned} \hat{j}' &= \hat{k}' \times \hat{i}' \\ &= (\sin \theta \sin \alpha \cos \theta_2 - \sin \theta_2 \sin \alpha_2 \cos \theta) \hat{i} \\ &\quad + (\sin \theta_2 \cos \alpha_2 \cos \theta - \sin \theta \cos \alpha \cos \theta_2) \hat{j} \\ &\quad + (\sin \theta \cos \alpha \sin \theta_2 \sin \alpha_2 - \sin \theta_2 \cos \alpha_2 \sin \theta \sin \alpha) \hat{k} \end{aligned} \quad (3.47c)$$

From equation (3.46), the angular velocity of the sphere in the world coordinates is

$$\begin{aligned} \vec{\omega}_{sphere} &= [\dot{\theta}_a \sin \alpha_a \sin \theta_2 \cos \alpha_2 - \dot{\alpha}_a \sin \theta \cos \alpha \\ &\quad - \dot{\theta}_a \cos \alpha_a (\sin \theta \sin \alpha \cos \theta_2 - \sin \theta_2 \sin \alpha_2 \cos \theta)] \hat{i} \\ &\quad + [\dot{\theta}_a \sin \alpha_a \sin \theta_2 \sin \alpha_2 - \dot{\alpha}_a \sin \theta \sin \alpha \\ &\quad - \dot{\theta}_a \cos \alpha_a (\sin \theta_2 \cos \alpha_2 \cos \theta - \sin \theta \cos \alpha \cos \theta_2)] \hat{j} \\ &\quad + [\dot{\theta}_a \sin \alpha_a \cos \theta_2 - \dot{\alpha}_a \cos \theta \\ &\quad - \dot{\theta}_a \cos \alpha_a (\sin \theta \cos \alpha \sin \theta_2 \sin \alpha_2 - \sin \theta_2 \cos \alpha_2 \sin \theta \sin \alpha)] \hat{k} \end{aligned} \quad (3.48)$$

Since the sphere cannot turn while standing, the angular velocity in \hat{k}' direction is set to zero. The area enclosed by an arbitrary closed curve can also be obtained analytically. In reference to figure 3.25 the area bounded by the closed curve $\theta_a = f(\alpha_a)$ is found by the double integral

$$A = \int_{\alpha_a(0)}^{\alpha_a(t_f)} \int_{\theta_a(0)}^{f(\alpha_a)} \sin \theta_a d\theta_a d\alpha_a \quad (3.49)$$

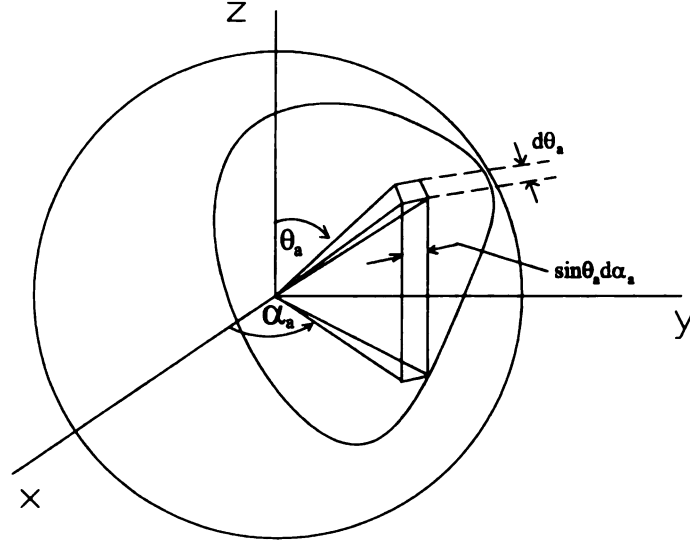


Figure 3.25. Surface integral for arbitrary closed curve on sphere

We present two simulations, 4 and 5. The closed curve in simulation 4 is given by equation

$$\theta_a(t) = \pi - \sin \alpha_a(t), \quad \alpha(0) = 0, \alpha(t_f) = \pi$$

In simulation 5, the closed curved satisfy

$$\theta_a(t) = \pi - \sin \alpha_a(t) + 0.3 \sin 5\alpha_a(t), \quad \alpha(0) = 0, \alpha(t_f) = \pi$$

The simulation results are shown in table 3.2 and figures 3.26-3.29. Figures 3.26 and 3.28 show trajectories of center point and the point initially at the top of the sphere. Figures 3.27 and 3.29 shown the orientation of the sphere.

Table 3.2. Results from simulations 4 and 5 of the open loop control of the sphere rolling along arbitrary closed paths.

$\theta_a(\alpha_a)$	t	$\alpha_a(0)$	$\dot{\alpha}_a(t)$	Surface Area from Integration	Amount of rotation obtained through numerical integration (radian)
$\pi - \sin\alpha_a$	$0 \rightarrow 20$	0	$\pi/20$	0.7376	0.7376
$\pi - \sin\alpha_a + 0.3 \sin 5\alpha_a$	$0 \rightarrow 20$	0	$\pi/20$	0.7912	0.7912

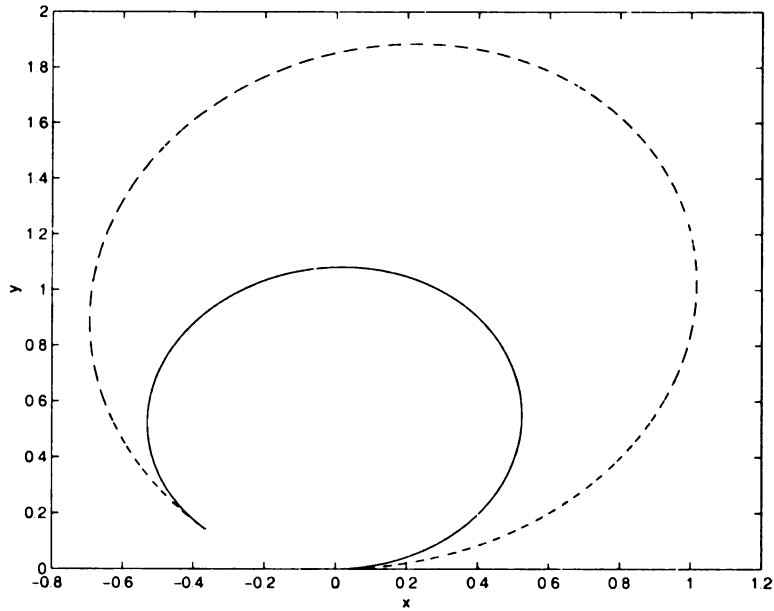


Figure 3.26. Trajectory of the sphere in simulation 4. Solid line represents trajectory of center of the sphere, dashed line represents trajectory of point P .

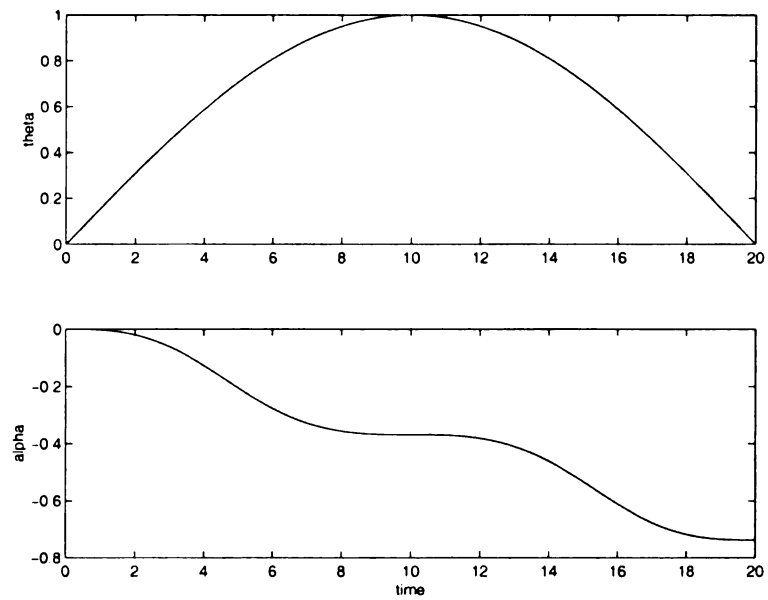


Figure 3.27. Variables θ and α from simulation 4 for open loop control of sphere.

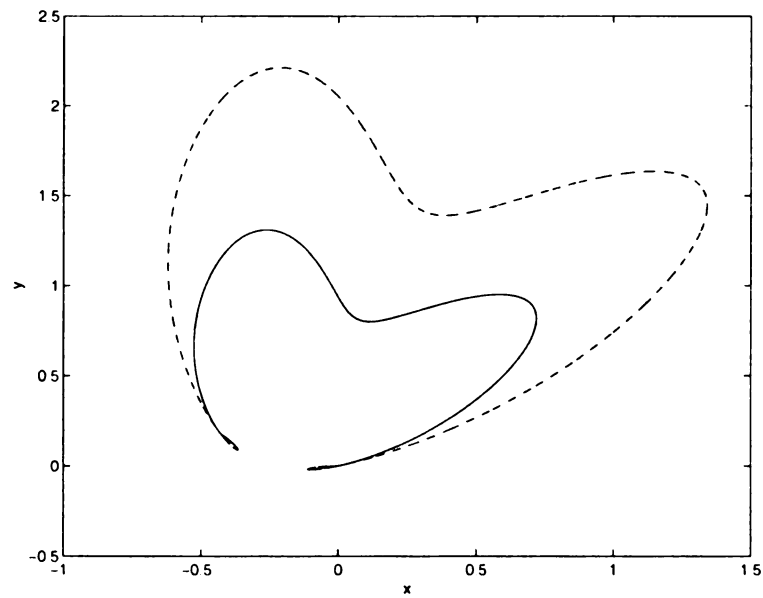


Figure 3.28. Trajectory of the sphere in simulation 5. Solid line represents trajectory of center of the sphere, dashed line represents trajectory of point P .

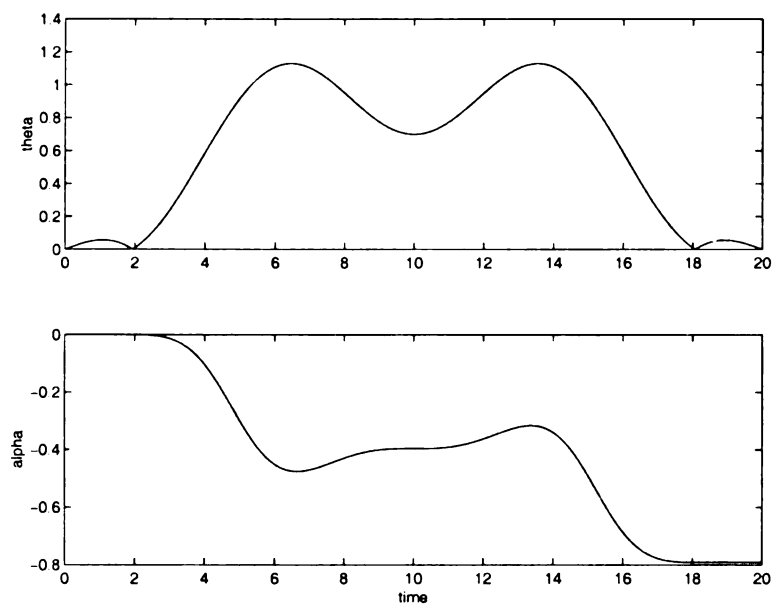


Figure 3.29. Variables θ and α from simulation 5 for open loop control of sphere.

CHAPTER 4

Conclusion

Rolling without slipping is a nonholonomic constraint that is quite common in mechanical systems. These constraints are not integrable, and can not be used to solve for a set of independent generalized coordinates. Motion planning for nonholonomic system is more difficult than that for holonomic systems, since independent generalized coordinates do not exist. Therefore only motions that satisfy the instantaneous nonholonomic constraints are feasible. Stability problems for nonholonomic systems are also interesting since there exists no smooth time-invariant static-state feedback that can asymptotically stabilize nonholonomic systems [7].

In this thesis, we present two control strategies for two nonholonomic systems, namely, the two-wheeled mobile robot and the rolling sphere. For the mobile robot, we present an almost-smooth time-invariant state feedback law that can asymptotically stabilize the mobile robot from practically any initial posture to the desired posture. The only exception is an equilibrium manifold of measure zero (containing the desired posture) from where the robot cannot move. All points on the manifold (except the desired posture) are however unstable and therefore any posture arbitrary close to, but not exactly on, the manifold can be asymptotically converged to the desired posture. This gap in the posture space is not a limitation of the controller. Our controller has an ability to turn the robot more than 360 degrees between initial

and final configurations, which will be useful for tethered mobile robots in factory automation. Our controller design, which is based on the kinematic model of mobile robot in chained form, provides a means of generalization to other nonholonomic systems. Our controller is motivated by the idea of a nonlinear oscillator, which has not been previously used for nonholonomic systems. The results presented here show that the nonlinear oscillator can be creatively used to control nonholonomic systems. The extension of the controller to higher order nonholonomic systems does not seem trivial and further investigation is required in this direction. The effect of control parameters to the trajectories of the system is needed to be investigated. Selection of the control parameters to optimize trajectories of the system still requires further studies. The implementation of the closed loop strategy to the mobile robot experiment is needed to be done.

For the case of the rolling sphere, which has higher order than the two-wheeled mobile robot, we proposed a motion planning scheme using spherical triangles. The goal is to control the sphere to the desired location while bringing a certain point on the surface of the sphere to the top. The basic idea is that when the sphere rolls along its spherical triangle, it moves to another location with a pure rotation about the vertical axis. This location, which can be found from plane trigonometry, is a function of the sides and angles of the spherical triangle. The amount of rotation is equal to the surface area of the sphere bounded by the spherical triangle divided by the radius of the sphere squared. This equality is also shown to be true for arbitrary closed curves on the sphere. The formulation along with simulation results is presented to demonstrate the control strategy. However, further studies need to be done in order to use closed curve on the sphere in open loop control. Closed loop control for rolling sphere has been studied but the effective control strategy has not been found. Control strategies for the different final configuration have to be investigated. The improvement of locomotion dynamics of rolling sphere is also

needed. The design and development of a spherical robot and its internal mechanism is required before an implementation of the control strategies.

APPENDICES

APPENDIX A

APPENDIX A

Program for Calculating Vector Rotation

The Mathematica program used to calculate the vector rotation in equation (3.16) is presented in this section. The final set of coordinates are i_3 , j_3 , and k_3 . They correspond to coordinates \hat{i}' , \hat{j}' and \hat{k}' in section 3.3, respectively.

```
In[1]:= i = {1, 0, 0};
        j = {0, 1, 0};
        k = {0, 0, 1};
        l1 = i
        i1 = (1 - Cos[a]) (l1 . i) l1 + Cos[a] i + Sin[a] (Cross[l1, i])
        j1 = (1 - Cos[a]) (l1 . j) l1 + Cos[a] j + Sin[a] (Cross[l1, j])
        k1 = (1 - Cos[a]) (l1 . k) l1 + Cos[a] k + Sin[a] (Cross[l1, k])
        l2 = Cos[CC] i - Sin[CC] j
        i2 = (1 - Cos[b]) (l2 . i1) l2 + Cos[b] i1 - Sin[b] (Cross[l2, i1])
        j2 = (1 - Cos[b]) (l2 . j1) l2 + Cos[b] j1 - Sin[b] (Cross[l2, j1])
        k2 = (1 - Cos[b]) (l2 . k1) l2 + Cos[b] k1 - Sin[b] (Cross[l2, k1])
        l3 = TrigExpand[Cos[AA + CC] i - Sin[AA + CC] j]
        i3 = (1 - Cos[c]) (l3 . i2) l3 + Cos[c] i2 + Sin[c] (Cross[l3, i2])
        j3 = (1 - Cos[c]) (l3 . j2) l3 + Cos[c] j2 + Sin[c] (Cross[l3, j2])
        k3 = (1 - Cos[c]) (l3 . k2) l3 + Cos[c] k2 + Sin[c] (Cross[l3, k2])

Out[1]= {1, 0, 0}

Out[2]= {1, 0, 0}

Out[3]= {0, Cos[a], Sin[a]}

Out[4]= {0, -Sin[a], Cos[a]}

Out[5]= {Cos[CC], -Sin[CC], 0}

Out[6]= {Cos[b] + (1 - Cos[b]) Cos[CC]^2, -(1 - Cos[b]) Cos[CC] Sin[CC],
        -Sin[b] Sin[CC]}

Out[7]= {-Cos[a] (1 - Cos[b]) Cos[CC] Sin[CC] + Sin[a] Sin[b] Sin[CC],
        Cos[a] Cos[b] + Cos[CC] Sin[a] Sin[b] + Cos[a] (1 - Cos[b]) Sin[CC]^2,
        Cos[b] Sin[a] - Cos[a] Cos[CC] Sin[b]}

Out[8]= {(1 - Cos[b]) Cos[CC] Sin[a] Sin[CC] + Cos[a] Sin[b] Sin[CC],
        -Cos[b] Sin[a] + Cos[a] Cos[CC] Sin[b] - (1 - Cos[b]) Sin[a] Sin[CC]^2,
        Cos[a] Cos[b] + Cos[CC] Sin[a] Sin[b]}

Out[9]= {Cos[AA] Cos[CC] - Sin[AA] Sin[CC], -Cos[CC] Sin[AA] - Cos[AA] Sin[CC], 0}
```

```

Out[10]= {Cos[c] (Cos[b] + (1 - Cos[b]) Cos[CC]^2) +
Sin[c] (Cos[CC] Sin[AA] Sin[b] Sin[CC] + Cos[AA] Sin[b] Sin[CC]^2) +
(1 - Cos[c]) (Cos[AA] Cos[CC] - Sin[AA] Sin[CC])
(- (1 - Cos[b]) Cos[CC] Sin[CC] (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC]) +
(Cos[b] + (1 - Cos[b]) Cos[CC]^2) (Cos[AA] Cos[CC] - Sin[AA] Sin[CC])),
- (1 - Cos[b]) Cos[c] Cos[CC] Sin[CC] +
Sin[c] (Cos[AA] Cos[CC] Sin[b] Sin[CC] - Sin[AA] Sin[b] Sin[CC]^2) +
(1 - Cos[c]) (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC])
(- (1 - Cos[b]) Cos[CC] Sin[CC] (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC]) +
(Cos[b] + (1 - Cos[b]) Cos[CC]^2) (Cos[AA] Cos[CC] - Sin[AA] Sin[CC])),
-Cos[c] Sin[b] Sin[CC] + Sin[c] (Cos[b] Cos[CC] Sin[AA] +
Cos[CC]^3 Sin[AA] - Cos[b] Cos[CC]^3 Sin[AA] + Cos[AA] Cos[b] Sin[CC] +
Cos[CC] Sin[AA] Sin[CC]^2 - Cos[b] Cos[CC] Sin[AA] Sin[CC]^2)}

Out[11]= {Sin[c] (-Cos[b] Cos[CC] Sin[a] Sin[AA] + Cos[a] Cos[CC]^2 Sin[AA] Sin[b] -
Cos[AA] Cos[b] Sin[a] Sin[CC] + Cos[a] Cos[AA] Cos[CC] Sin[b] Sin[CC]) +
Cos[c] (-Cos[a] (1 - Cos[b]) Cos[CC] Sin[CC] + Sin[a] Sin[b] Sin[CC]) +
(1 - Cos[c])
(Cos[AA] Cos[CC] - Sin[AA] Sin[CC]) ((Cos[AA] Cos[CC] - Sin[AA] Sin[CC])
(-Cos[a] (1 - Cos[b]) Cos[CC] Sin[CC] + Sin[a] Sin[b] Sin[CC]) +
(-Cos[CC] Sin[AA] - Cos[AA] Sin[CC]) (Cos[a] Cos[b] +
Cos[CC] Sin[a] Sin[b] + Cos[a] (1 - Cos[b]) Sin[CC]^2)),
Sin[c] (-Cos[AA] Cos[b] Cos[CC] Sin[a] + Cos[a] Cos[AA] Cos[CC]^2 Sin[b] +
Cos[b] Sin[a] Sin[AA] Sin[CC] - Cos[a] Cos[CC] Sin[AA] Sin[b] Sin[CC]) +
Cos[c]
(Cos[a] Cos[b] + Cos[CC] Sin[a] Sin[b] + Cos[a] (1 - Cos[b]) Sin[CC]^2) +
(1 - Cos[c]) (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC])
((Cos[AA] Cos[CC] - Sin[AA] Sin[CC])
(-Cos[a] (1 - Cos[b]) Cos[CC] Sin[CC] + Sin[a] Sin[b] Sin[CC]) +
(-Cos[CC] Sin[AA] - Cos[AA] Sin[CC]) (Cos[a] Cos[b] +
Cos[CC] Sin[a] Sin[b] + Cos[a] (1 - Cos[b]) Sin[CC]^2)),
Cos[c] (Cos[b] Sin[a] - Cos[a] Cos[CC] Sin[b]) +
Sin[c] (Cos[a] Cos[AA] Cos[b] Cos[CC] + Cos[AA] Cos[CC]^2 Sin[a] Sin[b] -
Cos[a] Cos[b] Sin[AA] Sin[CC] - Cos[a] Cos[CC]^2 Sin[AA] Sin[CC] +
Cos[a] Cos[b] Cos[CC]^2 Sin[AA] Sin[CC] +
Cos[AA] Sin[a] Sin[b] Sin[CC]^2 - Cos[a] Sin[AA] Sin[CC]^3 +
Cos[a] Cos[b] Sin[AA] Sin[CC]^3)}

```

```

Out[12]= {Cos[c] ((1 - Cos[b]) Cos[CC] Sin[a] Sin[CC] + Cos[a] Sin[b] Sin[CC]) +
  Sin[c] (-Cos[a] Cos[b] Cos[CC] Sin[AA] - Cos[CC]^2 Sin[a] Sin[AA] Sin[b] -
  Cos[a] Cos[AA] Cos[b] Sin[CC] - Cos[AA] Cos[CC] Sin[a] Sin[b] Sin[CC]) +
  (1 - Cos[c])
  (Cos[AA] Cos[CC] - Sin[AA] Sin[CC]) ((Cos[AA] Cos[CC] - Sin[AA] Sin[CC])
  ((1 - Cos[b]) Cos[CC] Sin[a] Sin[CC] + Cos[a] Sin[b] Sin[CC]) +
  (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC]) (-Cos[b] Sin[a] +
  Cos[a] Cos[CC] Sin[b] - (1 - Cos[b]) Sin[a] Sin[CC]^2)),
Sin[c] (-Cos[a] Cos[AA] Cos[b] Cos[CC] - Cos[AA] Cos[CC]^2 Sin[a] Sin[b] +
  Cos[a] Cos[b] Sin[AA] Sin[CC] + Cos[CC] Sin[a] Sin[AA] Sin[b] Sin[CC]) +
  Cos[c]
  (-Cos[b] Sin[a] + Cos[a] Cos[CC] Sin[b] - (1 - Cos[b]) Sin[a] Sin[CC]^2) +
  (1 - Cos[c]) (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC])
  ((Cos[AA] Cos[CC] - Sin[AA] Sin[CC])
  ((1 - Cos[b]) Cos[CC] Sin[a] Sin[CC] + Cos[a] Sin[b] Sin[CC]) +
  (-Cos[CC] Sin[AA] - Cos[AA] Sin[CC]) (-Cos[b] Sin[a] +
  Cos[a] Cos[CC] Sin[b] - (1 - Cos[b]) Sin[a] Sin[CC]^2)),
Cos[c] (Cos[a] Cos[b] + Cos[CC] Sin[a] Sin[b]) +
  Sin[c] (-Cos[AA] Cos[b] Cos[CC] Sin[a] + Cos[a] Cos[AA] Cos[CC]^2 Sin[b] +
  Cos[b] Sin[a] Sin[AA] Sin[CC] + Cos[CC]^2 Sin[a] Sin[AA] Sin[CC] -
  Cos[b] Cos[CC]^2 Sin[a] Sin[AA] Sin[CC] +
  Cos[a] Cos[AA] Sin[b] Sin[CC]^2 + Sin[a] Sin[AA] Sin[CC]^3 -
  Cos[b] Sin[a] Sin[AA] Sin[CC]^3)}

```

APPENDIX B

APPENDIX B

Spherical Trigonometric Identities

We summarize spherical trigonometry identities from section 3.2 to be used for simplification of the rotation matrix in equation (3.17) in section 3.3.

Law of sines: In any spherical triangle ABC ,

$$\frac{\sin a}{\sin A} = \frac{\sin b}{\sin B} = \frac{\sin c}{\sin C} \quad (\text{B.1})$$

Law of cosines for sides: In any spherical triangle ABC ,

$$\cos a = \cos b \cos c + \sin b \sin c \cos A \quad (\text{B.2})$$

$$\cos b = \cos c \cos a + \sin c \sin a \cos B \quad (\text{B.3})$$

$$\cos c = \cos a \cos b + \sin a \sin b \cos C \quad (\text{B.4})$$

Law of cosines for angles: In any spherical triangle ABC ,

$$\cos A = -\cos B \cos C + \sin B \sin C \cos a \quad (\text{B.5})$$

$$\cos B = -\cos C \cos A + \sin C \sin A \cos b \quad (\text{B.6})$$

$$\cos C = -\cos A \cos B + \sin A \sin B \cos c \quad (\text{B.7})$$

And useful formulas deriving from Law of sines and Law of cosines,

$$\sin a \cos B = \sin c \cos b - \sin b \cos c \cos A \quad (\text{B.8})$$

$$\sin b \cos C = \sin a \cos c - \sin c \cos a \cos B \quad (\text{B.9})$$

$$\sin c \cos A = \sin b \cos a - \sin a \cos b \cos C \quad (\text{B.10})$$

$$\sin A \cos b = \sin C \cos B + \sin B \cos C \cos a \quad (\text{B.11})$$

$$\sin B \cos c = \sin A \cos C + \sin C \cos A \cos b \quad (\text{B.12})$$

$$\sin C \cos a = \sin B \cos A + \sin A \cos B \cos c \quad (\text{B.13})$$

$$\sin a \cos C = \sin b \cos c - \sin c \cos b \cos A \quad (\text{B.14})$$

$$\sin b \cos A = \sin c \cos a - \sin a \cos c \cos B \quad (\text{B.15})$$

$$\sin c \cos B = \sin a \cos b - \sin b \cos a \cos C \quad (\text{B.16})$$

$$\sin A \cos c = \sin B \cos C + \sin C \cos B \cos a \quad (\text{B.17})$$

$$\sin B \cos a = \sin C \cos A + \sin A \cos C \cos b \quad (\text{B.18})$$

$$\sin C \cos b = \sin A \cos B + \sin B \cos A \cos c \quad (\text{B.19})$$

And the basic trigonometric identity

$$\sin^2 A + \cos^2 A = 1 \quad (\text{B.20})$$

APPENDIX C

APPENDIX C

Simplification of Rotation Matrix

In this section, coordinate rotation is used to find the amount of rotation in equation (3.17) resulting from the sphere rolling along a spherical triangle. For simplicity, we use symbol S_a and C_a to represent $\sin a$ and $\cos a$, respectively. The number of formula used which are expressed in brackets is referred to Appendix B.

Consider the first element in i_3 , from Mathematica

$$\begin{aligned}
 i_3(1) &= C_c(C_b + (1 - C_b)C_C^2) + S_c(C_C S_A S_b S_C + C_A S_b S_C^2) + (1 - C_c) \\
 &\quad (C_A C_C - S_A S_C)[- (1 - C_b)C_C S_C(-C_C S_A - C_A S_C) + (C_b + \\
 &\quad (1 - C_b)C_C^2)(C_A C_C - S_A S_C)] \\
 &= C_c C_C^2 + C_c C_b S_C^2 + S_c C_C S_A S_b S_C + S_c C_A S_b S_C^2 + \\
 &\quad (1 - C_c)(C_A C_C - S_A S_C)[C_C^2 S_C S_A + C_C S_C^2 C_A - \\
 &\quad C_b C_C^2 S_C S_A - C_b C_C S_C^2 C_A + C_b C_A C_C - C_b S_A S_C + \\
 &\quad C_C^3 C_A - C_C^2 S_A S_C - C_b C_C^3 C_A + C_b C_C^2 C_A + C_b C_C^2 S_A S_C]
 \end{aligned}$$

Cancel and rearrange terms

$$\begin{aligned}
 i_3(1) &= C_c C_C^2 + S_c C_C S_A S_b S_C + C_c C_b S_C^2 + S_c C_A S_b S_C^2 + \\
 &\quad (1 - C_c)(C_A C_C - S_A S_C)[C_C S_C^2 C_A + C_C^3 C_A - C_b S_A S_C - \\
 &\quad C_b C_C S_C^2 C_A - C_b C_C^3 C_A + C_b C_A C_C]
 \end{aligned}$$

Since $C_C^2 + S_C^2 = 1$ and use identity (B.2),

$$\begin{aligned}
 i_3(1) &= C_c C_C^2 + S_c C_C S_A S_b S_C + C_a S_C^2 + (1 - C_c)(C_A C_C - S_A S_C) \\
 &\quad [C_C C_A - C_b S_A S_C - C_b C_C C_A + C_b C_A C_C]
 \end{aligned}$$

Cancel terms and use identity (B.6),

$$i_3(1) = C_c C_C^2 + S_c C_C S_A S_b S_C + C_a S_C^2 + (1 - C_c)(C_A C_C - S_A S_C)[-C_B]$$

Expand all terms,

$$\begin{aligned}
 i_3(1) &= C_c C_C^2 + S_c C_C S_A S_b S_C + C_a S_C^2 - C_A C_C C_B + S_A S_C C_B + \\
 &\quad C_c C_A C_C C_B - C_c S_A S_C C_B
 \end{aligned}$$

Use identities (B.2), (B.7), and (B.19),

$$\begin{aligned} i_3(1) = & C_c C_C^2 + S_c C_C S_A S_b S_C + S_C^2 C_b C_c + S_C^2 S_b S_C C_A - \\ & C_A C_C C_B + S_A S_C C_B + C_c^2 C_C S_A S_B - C_c C_C^2 - C_c C_b S_C^2 + \\ & C_c^2 S_C S_B C_A \end{aligned}$$

Cancel and rearrange terms,

$$\begin{aligned} i_3(1) = & S_C^2 S_b S_C C_A + C_c^2 S_C S_B C_A + S_c C_C S_A S_b S_C + C_c^2 C_C S_A S_B - \\ & C_A C_C C_B + S_A S_C C_B \end{aligned}$$

Use identity (B.1),

$$\begin{aligned} i_3(1) = & S_C S_B C_A + C_c S_A S_B - C_A C_C C_B + S_A S_C C_B \\ = & -C_{A+B+C} \end{aligned}$$

Consider the second element in i_3 , from Mathematica

$$\begin{aligned} i_3(2) = & -(1 - C_b) C_c C_C S_C + S_c [C_A C_C S_b S_C - S_A S_b S_C^2] + \\ & (1 - C_c) (-C_C S_A - C_A S_C) [- (1 - C_b) C_C S_C (-C_C S_A - C_A S_C) + (C_b + \\ & (1 - C_b) C_C^2) (C_A C_C - S_A S_C)] \end{aligned}$$

Expand 1st two terms and all terms in square bracket

$$\begin{aligned} i_3(2) = & -C_c C_C S_C + C_b C_c C_C S_C + S_c C_A C_C S_b S_C - S_C S_A S_b S_C^2 + \\ & (1 - C_c) (-C_C S_A - C_A S_C) [C_C^2 S_C S_A + C_C S_C^2 C_A - C_b C_C^2 S_C S_A - \\ & C_b C_C S_C^2 C_A + C_b C_A C_C - C_b S_A S_C + C_C^3 C_A - C_C^2 S_A S_C - \\ & C_b C_C^3 C_A + C_b C_C^2 S_A S_C] \end{aligned}$$

Cancel terms and use identity (B.2) in square bracket

$$\begin{aligned} i_3(2) = & -C_c C_C S_C + C_C S_C C_a - S_C S_A S_b S_C^2 + (1 - C_c) (-C_C S_A - C_A S_C) \\ & [C_C S_C^2 C_A + C_C^3 C_A - C_b S_A S_C - C_b C_C S_C^2 C_A - C_b C_C^3 C_A + \\ & C_b C_A C_C] \end{aligned}$$

Use (B.20) and (B.6) and cancel terms, then expand terms

$$i_3(2) = -C_c C_C S_C + C_C S_C C_a - S_c S_A S_b S_C^2 + C_C S_A C_B + C_A S_C C_B -$$

$$C_c C_C S_A C_B - C_c C_A S_C C_B$$

Use (B.13) with second and third terms, substitute (B.1) for third term, substitute (B.7) for last term, then cancel terms

$$i_3(2) = C_C S_B C_A - S_C^2 S_A S_B S_C + C_C S_A C_B + C_A S_C C_B - C_c^2 S_C S_A S_B$$

Use (B.20) with second and fifth terms

$$\begin{aligned} i_3(2) &= C_C S_B C_A + C_C S_A C_B + C_A S_C C_B - S_A S_B S_C \\ &= S_{A+B+C} \end{aligned}$$

Consider the third element in i_3 , from Mathematica

$$\begin{aligned} i_3(3) &= -C_c S_b S_C + S_c [C_b C_C S_A + C_C^3 S_A - C_b C_C^3 S_A + C_A C_b S_C + \\ &\quad C_C S_A S_C^2 - C_b C_C S_A S_C^2] \end{aligned}$$

Use (B.20) and cancel terms

$$i_3(3) = -C_c S_b S_C + S_c [C_C S_A + C_A C_b S_C]$$

Use (B.12)

$$i_3(3) = -C_c S_b S_C + S_c S_B C_c$$

Use (B.1)

$$i_3(3) = 0$$

Consider the first element in j_3 , from Mathematica

$$\begin{aligned} j_3(1) &= S_c (-C_b C_C S_a S_A + C_a C_C^2 S_A S_b - C_A C_b S_a S_C + \\ &\quad C_a C_A C_C S_b S_C) + C_c (-C_a (1 - C_b) C_C S_C + S_a S_b S_C) + (1 - C_c) \\ &\quad (C_A C_C - S_A S_C) [(C_A C_C - S_A S_C) (-C_a (1 - C_b) C_C S_C + S_a S_b S_C) + \\ &\quad (-C_C S_A - C_A S_C) (C_a C_b + C_C S_a S_b + C_a (1 - C_b) S_C^2)] \end{aligned}$$

Use (B.16) and then expand terms,

$$\begin{aligned} j_3(1) &= S_c (-S_c C_C S_A C_B - C_A S_C S_c C_B) + C_c (-C_a C_C S_C + C_a C_b C_C S_C + \\ &\quad S_a S_b S_C) + (1 - C_c) (C_A C_C - S_A S_C) [-C_A C_C^2 C_a S_C + C_A C_C^2 C_a C_b S_C + \\ &\quad C_A C_C S_a S_b S_C + S_A S_C^2 C_a C_C - S_A S_C^2 C_a C_b C_C - S_A S_C^2 S_a S_b - \end{aligned}$$

$$C_C S_A C_a C_b - C_C^2 S_A S_a S_b + C_C S_A C_a S_C^2 + C_C S_A C_a C_b S_C^2 - \\ C_A S_C C_a C_b - C_A S_C C_C S_a S_b - C_A S_C^3 C_a + C_A S_C^3 C_a C_b]$$

Cancel and rearrange terms

$$j_3(1) = S_c(-S_c C_C S_A C_B - C_A S_C S_c C_B) + C_c(-C_a C_C S_C + C_a C_b C_C S_C + \\ S_a S_b S_C) + (1 - C_c)(C_A C_C - S_A S_C)[-C_A C_C^2 C_a S_C - C_A S_C^3 C_a - \\ C_C S_A C_a C_b - S_A S_C^2 S_a S_b - C_C^2 S_A S_a S_b + C_A C_C^2 C_a C_b S_C + \\ C_A S_C^3 C_a C_b - C_A S_C C_a C_b]$$

Use (B.20)

$$j_3(1) = S_c(-S_c C_C S_A C_B - C_A S_C S_c C_B) + C_c(-C_A C_C S_C + C_a C_b C_C S_C + \\ S_a S_b S_C) + (1 - C_c)(C_A C_C - S_A S_C)[-C_A C_a S_C - C_C S_A C_a C_b - S_A S_a S_b + \\ C_A C_a C_b S_C - C_A S_C C_a C_b]$$

Inside square bracket, cancel terms, use (B.18), (B.1), and (B.20), we left with $[-S_B]$.
Then expand all

$$j_3(1) = -S_c^2 C_C S_A C_B - S_c^2 C_A S_C C_B - C_c C_a C_C S_C + C_c C_a C_b C_C S_C + \\ C_c S_a S_b S_C - C_A C_C S_B + S_A S_C S_B + C_c C_A C_C S_B - C_c S_A S_C S_B$$

Use (B.20) with the first and second terms, use (B.4) for fourth term, and then rearrange terms

$$j_3(1) = -C_C S_A C_B + C_C S_A C_B C_c^2 - C_A S_C C_B + C_A S_C C_B C_c^2 - \\ C_c C_a C_C S_C + C_c^2 C_C S_C - C_c C_C^2 S_C S_a S_b + C_c S_a S_b S_C - \\ C_A C_C S_B + S_A S_C S_B + C_c C_A C_C S_B - C_c S_A S_C S_B$$

Use (B.13) for the second and third terms, use (B.20) in the 8th term,

$$j_3(1) = -C_C S_A C_b + C_C C_c S_C C_a - C_A S_C C_B + C_A S_C C_B C_c^2 - \\ C_c C_a C_C S_C + C_c^2 C_C S_C - C_c S_a S_b S_C + C_c S_C^3 S_a S_b + \\ C_c S_a S_b S_C - C_A C_C S_B + S_A S_C S_B - C_c S_A S_C S_B$$

Use (B.1) in 8th, cancel 2nd and 5th, cancel 7th and 9th,

$$j_3(1) = -C_C S_A C_b - C_A S_C C_B + C_A S_C C_B C_c^2 + C_c^2 C_C S_C + \\ C_c S_A S_B S_C^2 S_C - C_A C_C S_B + S_A S_C S_B - C_c S_A S_C S_B$$

Use (B.20) in 8th, then cancel terms

$$j_3(1) = -C_C S_A C_b - C_A S_C C_B + C_A S_C C_B C_c^2 + C_c^2 C_C S_C - \\ C_c S_A S_B S_C C_c^2 - C_A C_C S_B + S_A S_C S_B$$

Use (7) with 4th and 8th, then cancel with 6th,

$$j_3(1) = -C_C S_A C_b - C_A S_C C_B - C_A C_C S_B + S_A S_C S_B \\ = -S_{A+B+C}$$

Consider the second element in j_3 , from Mathematica

$$j_3(3) = S_c(-C_A C_b C_C S_a + C_a C_A C_C^2 S_b + C_b S_a S_A S_C - \\ C_a C_C S_A S_b S_C) + C_c(C_a C_b + C_C S_a S_b + C_a(1 - C_b)S_C^2) + \\ (1 - C_c)(-C_C S_A - C_A S_C)[(C_A C_C - S_A S_C)(-C_a(1 - C_b)C_c S_C + \\ S_a S_b S_C) + (-C_C S_A - C_A S_C)(C_a C_b + C_C S_a S_b + C_a(1 - C_b)S_C^2)]$$

Consider terms in square bracket, when expand all terms,

$$[...] = [-C_A C_C^2 C_a S_C + C_A C_C^2 C_a C_b S_C + C_A C_C S_a S_b S_C + \\ S_A S_C^2 C_a C_C - S_A S_C C_a C_b C_C S_C - S_A S_C^2 S_a S_b - C_C S_A C_a C_b - \\ C_C^2 S_A S_a S_b - C_C S_A C_a S_C^2 + C_C S_A C_a C_b S_C^2 - C_A S_C C_a C_b - \\ C_A S_C C_C S_a S_b - C_A S_C^3 C_a + C_A S_C C_a C_b S_C^2]$$

Cancel terms, 3rd and 12th, 4th and 9th, 5th and 10th. Pairs 1st-13th, 2nd-14th, and 6th-8th can be combined.

$$[...] = [-C_a S_C C_a + C_A C_a C_b S_C - S_A S_a S_b - C_C S_A C_a C_b - C_A S_C C_a C_b]$$

The second and fifth terms are canceled. The first and forth terms can be combined using (18), therefore,

$$[...] = [-C_a^2 S_B - S_A S_a S_b]$$

Return to $j_3(3)$, use (B.16) in the first and (B.20) in the second brackets, then expand terms.

$$\begin{aligned} j_3(3) = & S_c^2 C_A C_C C_B - S_c^2 S_A S_C C_B - C_c C_a C_b C_C^2 - C_c C_C S_a S_b - \\ & C_c C_a S_C^2 - C_C S_A C_a^2 S_B - C_C S_A^2 S_a S_b - C_A S_C C_a^2 S_B - \\ & C_A S_C S_A S_a S_b - C_c C_C S_A C_a^2 S_B + C_c C_C S_A^2 S_a S_b + \\ & C_c C_A S_C C_a^2 S_B + C_c C_A S_C S_A S_a S_b \end{aligned}$$

Expand the 5th terms using (B.13), 6th using (B.11). Combine 10th and 11th, 8th and 9th.

$$\begin{aligned} j_3(3) = & S_c^2 C_A C_C C_B - S_c^2 S_A S_C C_B - C_c C_a C_b C_C^2 - C_c C_C S_a S_b - \\ & C_c S_C S_B C_A - C_C^2 S_C S_A C_B - C_a S_A^2 C_b + C_a S_A S_C C_B - C_C S_A^2 S_a S_b - \\ & C_A S_C S_B - C_c C_C S_A S_B + C_c C_A S_C C_a^2 S_B + C_c C_A S_C S_A S_a S_b \end{aligned}$$

The 5th term combine with the 13th equal to 12th. Combine 2nd and 6th using (B.20), 7th and 9th using (4), and expand 8th using (17).

$$\begin{aligned} j_3(3) = & S_c^2 C_A C_C C_B - S_A S_C C_B - C_c C_a C_b C_C^2 - C_c C_C S_a S_b - S_A^2 C_c + \\ & S_A^2 C_c - S_A S_B C_C - C_A S_C S_B - C_c C_C S_A S_B \end{aligned}$$

The fifth term cancel with sixth term. The third, fourth, and ninth add up with first terms.

$$\begin{aligned} j_3(3) = & C_A C_C C_B - S_A S_C C_B - S_A S_B C_C - S_C S_B C_A \\ = & -C_{A+B+C} \end{aligned}$$

Consider the third element in j_3 , from Mathematica

$$\begin{aligned} j_3(3) = & C_c(C_b S_a - C_a C_C S_b) + S_c(C_a C_A C_b C_C + C_A C_C^2 S_a S_b - \\ & C_a C_b S_A S_C - C_a C_C^2 S_A S_C + C_a C_b C_C^2 S_A S_C + C_A S_a S_b S_C^2 - \\ & C_a S_A S_C^3 + C_a C_b S_A S_C^3) \end{aligned}$$

Use (B.20), cancel terms, and expand

$$\begin{aligned} j_3(3) = & C_c C_b S_a - C_c C_a C_C S_b + S_c C_a C_A C_b C_C + S_c C_A S_a S_b - \\ & S_c C_a S_A S_C \end{aligned}$$

Use (B.14) in the third term

$$j_3(3) = C_c C_b S_a - C_c C_a C_C S_b + C_a C_C S_b C_c - C_a C_C^2 S_a + \\ S_c C_A S_a S_b - S_c C_a S_A S_C$$

Use (B.20) in 4th term

$$j_3(3) = C_c C_b S_a - C_c C_a C_C S_b + C_a C_C S_b C_c - C_a S_a + C_a S_a S_C^2 + \\ S_c C_A S_a S_b - S_c C_a S_A S_C$$

Use (B.2) in 1st and 6th, then cancel with 4th. Use (B.1) in 5th. Cancel 2nd and 3rd.

$$j_3(3) = C_a S_A S_c S_C - S_c C_a S_A S_C = 0$$

Consider the first element in k_3 , from Mathematica

$$k_3(1) = C_c((1 - C_b)C_C S_a S_C + C_a S_b S_C) + S_c(-C_a C_b C_C S_A - C_C^2 S_a S_A S_b - \\ C_a C_A C_b S_C - C_A C_C S_a S_b S_C) + (1 - C_c)(C_A C_C - S_A S_C)[(C_A C_C - S_A S_C) \\ ((1 - C_b)C_C S_a S_C + C_a S_b S_C) + (-C_C S_A - C_A S_C)(-C_b S_a + C_a C_C S_b - \\ (1 - C_b)S_a S_C^2)]$$

Use (B.10) in the first bracket, (B.4) in second. Use (B.10) (B.16) and (B.20) in square bracket.

$$k_3(1) = C_c(C_C S_a S_C + S_C S_c C_A) + S_c(-C_C S_A C_c - C_A S_C C_c) + \\ (1 - C_c)(C_A C_C - S_A S_C)[(C_A C_C - S_A S_C)(C_C S_a S_C + S_C S_c C_A) + \\ (-C_C S_A - C_A S_C)(C_C S_c C_A - S_a S_C^2)]$$

Expand terms,

$$k_3(1) = C_c C_C S_a S_C + C_c S_C S_c C_A - S_c C_C S_A C_c - S_c C_A S_C C_c + \\ (1 - C_c)(C_A C_C - S_A S_C)[C_A C_C^2 S_a S_C + C_A^2 C_C S_C S_c - \\ S_A S_C^2 C_C S_a - S_A S_C^2 S_c C_A - C_C^2 S_A S_c C_A + C_C S_A S_a S_C^2 - \\ C_A^2 S_C C_C S_c + C_A S_C^3 S_a]$$

The first and third terms can be canceled using (B.1). Second and fourth are canceled.

In square bracket, use (B.1) and (B.20), all terms can be canceled. Therefore,

$$k_3(1) = 0$$

Consider the second element in k_3 , from Mathematica

$$\begin{aligned} k_3(2) = & S_c(-C_a C_A C_b C_C - C_A C_C^2 S_a S_b + C_a C_b S_A S_C + \\ & C_C S_a S_A S_b S_C) + C_c(-C_b S_a + C_a C_C S_b - (1 - C_b) S_a S_C^2) + \\ & (1 - C_c)(-C_C S_A - C_A S_C)[(C_A C_C - S_A S_C)((1 - C_b) C_C S_a S_C + \\ & C_a S_b S_C) + (-C_C S_A - C_A S_C)(-C_b S_a + C_a C_C S_b - (1 - C_b) S_a S_C^2)] \end{aligned}$$

Use (B.4) in the first bracket, (B.10) and (B.20) in second bracket. Use (B.10) and (B.20) in square bracket.

$$\begin{aligned} k_3(2) = & S_c(-C_A C_C C_c + S_A S_C C_c) + C_c(C_C S_c C_A - S_a S_C^2) + \\ & (1 - C_c)(-C_C S_A - C_A S_C)[(C_A C_C - S_A S_C)(C_C S_a S_C + S_c C_A S_C) + \\ & (-C_C S_A - C_A S_C)(C_C S_c C_A - S_a S_C^2)] \end{aligned}$$

Expand terms,

$$\begin{aligned} k_3(2) = & -S_c C_A C_C C_c + S_c S_A S_C C_c + C_c C_C S_c C_A - C_c S_a S_C^2 + \\ & (1 - C_c)(-C_C S_A - C_A S_C)[C_A C_C^2 S_a S_C + S_c C_A^2 S_C C_C - \\ & S_A S_C^2 C_C S_a - S_A S_C^2 S_c C_A - C_C^2 S_c C_A S_A + C_C S_A S_a S_C^2 - \\ & C_A^2 S_C C_C S_c + C_A S_C^3 S_a] \end{aligned}$$

The first and third terms can be canceled using (B.1). Second and fourth are canceled. In square bracket, use (B.1) and (B.20), all terms can be canceled. Therefore,

$$k_3(2) = 0$$

Consider the third element in k_3 , from Mathematica

$$\begin{aligned} k_3(3) = & C_c(C_a C_b + C_C S_a S_b) + S_c(-C_A C_b C_C S_a + C_a C_A C_C^2 S_b + \\ & C_b S_a S_A S_C + C_C^2 S_a S_A S_C - C_b C_C^2 S_a S_A S_C + C_a C_A S_b S_C^2 + \\ & S_a S_A S_C^3 - C_b S_a S_A S_C^3) \end{aligned}$$

Use (B.4) in the first bracket. Use (B.10) and (B.20) in the second bracket.

$$k_3(3) = C_c^2 + S_c(S_a S_A S_C + C_A S_c C_A)$$

By using (B.1),

$$k_3(3) = C_c^2 + S_c^2 = 0$$

This completes the simplification of all entries of rotation matrix in equation (3.17). After simplification we can write the final coordinates of the sphere $\hat{i}', \hat{j}', \hat{k}'$, or i_3, j_3, k_3 in this appendix, as

$$\begin{aligned}\hat{i}' &= -\cos(A + B + C)\hat{i} + \sin(A + B + C)\hat{j} \\ \hat{j}' &= -\sin(A + B + C)\hat{i} - \cos(A + B + C)\hat{j} \\ \hat{k}' &= \hat{k}\end{aligned}$$

or, in matrix form

$$\begin{bmatrix} \hat{i}' \\ \hat{j}' \\ \hat{k}' \end{bmatrix} = \begin{bmatrix} -\cos(A + B + C) & \sin(A + B + C) & 0 \\ -\sin(A + B + C) & -\cos(A + B + C) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{bmatrix}$$

APPENDIX D

APPENDIX D

Fortran program for numerical root finding

This section presents the fortran routine used for numerically root-finding of equation (3.41), (3.42) and (3.43) to find two sides and the included angle of spherical triangle. We follow the instruction of root-finding in chapter 9 of the book by W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery [29]. Some necessary modification is made in order to apply to our set of equations.

Main program : sphtri.f

```
INTEGER n
REAL x(3),f(3)
PARAMETER (pi=3.1415926535897)
COMMON /desire/ dd,phid,alphad
n=3
write(*,*) 'guess a,b,CC(rad)'
read(*,*) x(1),x(2),x(3)
write(*,*) 'desired d,phi(rad),alpha(rad)'
read(*,*) dd,phid,alphad
call newt(x,n,check)
write(*,*) 'a,b,CC(deg),check'
write(*,*) x(1),x(2),x(3)*180/pi,check
call funcv(n,x,f)
write(*,*) 'f(1),f(2),f(3)'
write(*,*) f(1),f(2),f(3)
END
```

Subroutine : newt.f

```
SUBROUTINE newt(x,n,check)
INTEGER n,nm,NP,MAXITS
LOGICAL check
REAL x(n),fvec,TOLF,TOLMIN,TOLX,STPMX
PARAMETER (NP=40,MAXITS=200,TOLF=1.e-6,TOLMIN=1.e-8,TOLX=1.e-8,
* STPMX=100.)
COMMON /newtv/ fvec(NP),nm
SAVE /newtv/
INTEGER i,its,j,indx(NP)
REAL d,den,f,fold,stpmax,sum,temp,test,fjac(NP,NP),g(NP),p(NP),
* xold(NP),fmin
EXTERNAL fmin
nn=n
f=fmin(x)
test=0.
do i=1,n
  if(abs(fvec(i)).gt.test)test=abs(fvec(i))
enddo
if(test.lt..01*TOLF)then
  check=.false.
  return
```

```

endif
sum=0.
do i=1,n
    sum=sum+x(i)**2
enddo
stpmax=STPMX*max(sqrt(sum),float(n))
do its=1,MAXITS
    call fdjac(n,x,fvec,NP,fjac)
    do i=1,n
        sum=0.
        do j=1,n
            sum=sum+fjac(j,i)*fvec(j)
        enddo
        g(i)=sum
    enddo
    do i=1,n
        xold(i)=x(i)
    enddo
    fold=f
    do i=1,n
        p(i)=-fvec(i)
    enddo
    call ludcmp(fjac,n,NP,indx,d)
    call lubksb(fjac,n,NP,indx,p)
    call lsrch(n,xold,fold,g,p,x,f,stpmax,check,fmin)
    test=0.
    do i=1,n
        if(abs(fvec(i)).gt.test)test=abs(fvec(i))
    enddo
    if(test.lt.TOLF)then
        check=.false.
        return
    endif
    if(check)then
        test=0.
        den=max(f,.5*n)
        do i=1,n
            temp=abs(g(i))*max(abs(x(i)),1.)/den
            if(temp.gt.test)test=temp
        enddo
        if(test.lt.TOLMIN)then
            check=.true.
        else
            check=.false.
        endif
        return
    endif
endif
test=0.
do i=1,n
    temp=(abs(x(i)-xold(i)))/max(abs(x(i)),1.)
    if(temp.gt.test)test=temp
enddo
if(test.lt.TOLX)return
enddo
pause 'MAXITS exceeded in newt'
END

```

Subroutine : ludcmp.f

```

SUBROUTINE ludcmp(a,n,np,indx,d)
INTEGER n,np,indx(n),NMAX
REAL d,a(np,np),TINY

```

```

PARAMETER (NMAX=500,TINY=1.0e-20)
INTEGER i,imax,j,k
REAL aamax,dum,sum,vv(NMAX)
d=1.
do i=1,n
  aamax=0.
  do j=1,n
    if(abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))
  enddo
  if(aamax.eq.0.) pause 'singular matrix in ludcmp'
  vv(i)=1./aamax
enddo
do j=1,n
  do i=1,j-1
    sum=a(i,j)
    do k=1,i-1
      sum=sum-a(i,k)*a(k,j)
    enddo
    a(i,j)=sum
  enddo
  aamax=0.
  do i=j,n
    sum=a(i,j)
    do k=1,j-1
      sum=sum-a(i,k)*a(k,j)
    enddo
    a(i,j)=sum
    dum=vv(i)*abs(sum)
    if(dum.ge.aamax)then
      imax=i
      aamax=dum
    endif
  enddo
  if(j.ne.imax)then
    do k=1,n
      dum=a(imax,k)
      a(imax,k)=a(j,k)
      a(j,k)=dum
    enddo
    d=-d
    vv(imax)=vv(j)
  endif
  indx(j)=imax
  if(a(j,j).eq.0.)a(j,j)=TINY
  if(j.ne.n)then
    dum=1./a(j,j)
    do i=j+1,n
      a(i,j)=a(i,j)*dum
    enddo
  endif
enddo
return
END

```

Subroutine : lubksb.f

```

SUBROUTINE lubksb(a,n,np,indx,b)
INTEGER n,np,indx(n)
REAL a(np,np),b(n)
INTEGER i,ii,j,ll
REAL sum
ii=0

```

```

do i=1,n
  ll=indx(i)
  sum=b(ll)
  b(ll)=b(i)
  if(ii.ne.0)then
    do j=ii,i-1
      sum=sum-a(i,j)*b(j)
    enddo
  else if (sum.ne.0.)then
    ii=i
  endif
  b(i)=sum
enddo
do i=n,1,-1
  sum=b(i)
  do j=i+1,n
    sum=sum-a(i,j)*b(j)
  enddo
  b(i)=sum/a(i,i)
enddo
return
END

```

Subroutine : Insrch.f

```

SUBROUTINE Insrch(n,xold,fold,g,p,x,f,stpmax,check,func)
INTEGER n
LOGICAL check
REAL f,fold,stpmax,g(n),p(n),x(n),xold(n),func,ALF,TOLX
PARAMETER (ALF=1.e-4,TOLX=1.e-7)
EXTERNAL func
INTEGER i
REAL a,alam,alam2,alamin,b,disc,f2,rhs1,rhs2,slope,sum,temp,
* test,tmplam
check=.false.
sum=0.
do i=1,n
  sum=sum+p(i)*p(i)
enddo
sum=sqrt(sum)
if(sum.gt.stpmax)then
  do i=1,n
    p(i)=p(i)*stpmax/sum
  enddo
endif
slope=0.
do i=1,n
  slope=slope+g(i)*p(i)
enddo
if(slope.ge.0.) pause 'roundoff problem in Insrch'
test=0.
do i=1,n
  temp=abs(p(i))/max(abs(xold(i)),1.)
  if(temp.gt.test)test=temp
enddo
alamin=TOLX/test
alam=1.
1 continue
do i=1,n
  x(i)=xold(i)+alam*p(i)
enddo
f=func(x)

```

```

    if(alam.lt.alamin)then
      do i=1,n
        x(i)=xold(i)
      enddo
      check=.true.
      return
    else if(f.le.fold+ALF*alam*slope)then
      return
    else
      if(alam.eq.1.)then
        tmplam=-slope/(2.*(f-fold-slope))
      else
        rhs1=f-fold-alam*slope
        rhs2=f2-fold-alam2*slope
        a=(rhs1/alam**2-rhs2/alam2**2)/(alam-alam2)
        b=(-alam2*rhs1/alam**2+alam*rhs2/alam2**2)/
*      (alam-alam2)
    if(a.eq.0.)then
      tmplam=-slope/(2.*b)
    else
      disc=b*b-3.*a*slope
      if(disc.lt.0.)then
        tmplam=.5*alam
      else if(b.le.0.)then
        tmplam=(-b+sqrt(disc))/(3.*a)
      else
        tmplam=-slope/(b+sqrt(disc))
      endif
    endif
    if(tmplam.gt..5*alam)tmplam=.5*alam
  endif
endif
alam2=alam
f2=f
alam=max(tmplam,.1*alam)
goto 1
END

```

Subroutine: fdjac.f

```

SUBROUTINE fdjac(n,x,fvec,np,df)
  INTEGER n,np,NMAX
  REAL df(np,np),fvec(n),x(n),EPS
  PARAMETER (NMAX=40,EPS=1.e-4)
  INTEGER i,j
  REAL h,temp,f(NMAX)
  do j=1,n
    temp=x(j)
    h=EPS*abs(temp)
    if(h.eq.0.)h=EPS
    x(j)=temp+h
    h=x(j)-temp
    call funcv(n,x,f)
    x(j)=temp
    do i=1,n
      df(i,j)=(f(i)-fvec(i))/h
    enddo
  enddo
  return
END

```

Subroutine : fmin.f

```
FUNCTION fmin(x)
INTEGER n,NP
REAL fmin,x(*),fvec
PARAMETER (NP=40)
COMMON /newtv/ fvec(NP),n
SAVE /newtv/
INTEGER i
REAL sum
call funcv(n,x,fvec)
sum=0.
do i=1,n
    sum=sum+fvec(i)**2
enddo
fmin=0.5*sum
return
END
```


BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] M. Aicardi, G. Casalino, A. Balestrino, and A. Bicchi. Closed-loop smooth steering of unicycle-like vehicles. *IEEE Conference on Decision and Control*, pages 2455–2458, 1994.
- [2] A. Astolfi. On the stabilization of nonholonomic systems. *IEEE Conference on Decision and Control*, pages 3481–3486, 1994.
- [3] Frank Ayres, Jr. *Schaum's Outline of Theory and Problems of Plane and Spherical Trigonometry*. McGraw Hill, New York, NY, 1954.
- [4] A. M. Bloch and S. Drakunov. Stabilization of a nonholonomic system via sliding modes. *IEEE Conference on Decision and Control*, pages 2961–2963, 1994.
- [5] A. M. Bloch, N. H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. *Proceedings of 29th IEEE International Conference on Decision and Control, Honolulu, Hawaii*, pages 1312–1314, 1990.
- [6] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch. Control and stabilization of nonholonomic dynamical systems. *IEEE Transactions on Automatic Control*, 37(11):1746–1757, 1992.
- [7] R. W. Brockett. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory, Editors. R. W. Brockett, R. S. Millman, and H. F. Sussman, Birkhauser*, pages 181–191, 1983.
- [8] J. M. Coron. Global asymptotic stabilization for controllable systems without drift. *Mathematics of Control, Signals, and Systems*, 5:295–312, 1992.
- [9] C. Canudas de Wit and C. Samson. Path following of a 2-dof wheeled mobile robot under kinematic constraints. *Proceedings of First European Control Conference, Grenoble, France*, pages 2084–2088, 1991.
- [10] C. Canudas de Wit and O. J. Sordalen. Exponential stabilization of mobile robots with nonholonomic constraint. *IEEE Transactions on Automatic Control*, 37(1):1791–1797, 1992.

- [11] J. Guldner and V. I. Utkin. Stabilization of nonholonomic mobile robots using lyapunov functions for navigation and sliding mode control. *IEEE Conference on Decision and Control*, pages 2967–2972, 1994.
- [12] L. Gurvits and Z. X. Li. Smooth time-periodic feedback solutions for nonholonomic motion planning. *Progress in Nonholonomic Motion Planning*, Z. X. Li and J. Canny, editors, Kluwer Academic Press, 1992.
- [13] A. Isidori. *Nonlinear Control Systems*. Springer Verlag, New York, NY, third edition, 1995.
- [14] J. L. Junkins and J. D. Turner. *Optimal Spacecraft Rotational Maneuvers*. Elsevier, New York, NY, 1986.
- [15] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. *Proceedings of IEEE International Conference on Robotic and Automation, Cincinnati, Ohio*, pages 384–389, 1990.
- [16] L. M. Kells, W. F. Kern, and J. R. Bland. *Plane and Spherical Trigonometry*. McGraw Hill, New York, NY, second edition, 1940.
- [17] H. Khenous and C. Canudas de Wit. On the construction of stabilizing discontinuous controllers for nonholonomic systems. *Nonlinear Control Systems Design Symposium (NOLCOS), IFAC Preprints, Tahoe City*, pages 747–752, 1995.
- [18] I. Kolmanovsky and N. H. McClamroch. Developments in nonholonomic control problems. *IEEE Control Systems*, 15(6):20–36, 1995.
- [19] G. A. Lafferriere. A general strategy for computing steering controls of systems without drift. *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1115–1120, 1991.
- [20] G. A. Lafferriere and H. Sussmann. A differential geometric approach to motion planning. *Nonholonomic Motion Planning*, Z. Li and J. F. Canny, editors, Kluwer, pages 235–270, 1993.
- [21] N. E. Leonard and P. S. Krishnaprasad. Averaging for attitude control and motion planning. *Proceedings of the 32nd Conference on Decision and Control*, pages 3098–3104, 1993.
- [22] A. Lewis, J. P. Ostrowski, R. Murray, and J. Burdick. Nonholonomic mechanics and locomotion: The snakeboard example. *Proceedings of the International Conference on Robotics and Automation*, pages 2391–2397, 1994.

- [23] Z. Li and J. F. Canny. *Nonholonomic Motion Planning*. Kluwer, 1993.
- [24] R. T. M'Closkey and R. M. Murray. Convergence rates for nonholonomic systems in power form. *American Control Conference, San Francisco, CA*, 1993.
- [25] R. Mukherjee and D. P. Anderson. A surface integral approach to the motion planning of nonholonomic systems. *Journal of Dynamic Systems, Measurement and Control*, 116:315–324, 1994.
- [26] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Ann Arbor, 1993.
- [27] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic control*, 38(5):700–716, 1993.
- [28] J. B. Pomet. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *System and Control Letters*, 18:147–158, 1992.
- [29] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.
- [30] C. Rui and N. H. McClamroch. Stabilization and asymptotic path tracking of a rolling disk. *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 4294–4299, 1995.
- [31] C. Samson. Velocity and torque feedback control of a nonholonomic cart. *International Workshop in Adaptive and Nonlinear Control: Issues in Robotics, Proceedings in Advanced Robot Control, Springer Verlag*, 162, 1990.
- [32] C. Samson. Time-varying feedback stabilization of a car-like wheeled mobile robots. *International Journal of Robotics Research*, 12(1):55–66, 1993.
- [33] H. J. Sussmann and W. Liu. Limits of highly oscillatory controls and approximation of general paths by admissible trajectories. *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 437–442, 1991.
- [34] I. Todhunter. *Spherical Trigonometry*. Macmillan, London, fourth edition, 1878.
- [35] G. C. Walsh, D. Tilbury, S. Sastry, and R. Murray and J. P. Laumond. Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, 39:216–222, 1994.

MICHIGAN STATE UNIV. LIBRARIES



31293017710868