

Motion Planning and Control for a Class of Underactuated Systems

白, 楊

<https://hdl.handle.net/2324/2236225>

出版情報 : Kyushu University, 2018, 博士 (工学) , 課程博士
バージョン :
権利関係 :



2019 Doctoral Thesis

MOTION PLANNING AND CONTROL FOR A
CLASS OF UNDERACTUATED SYSTEMS

Control Engineering Lab.

Department of Mechanical Engineering, Kyushu University

D3 Yang BAI

2019. 1. 10

Acknowledgements

My thesis is completed with a lot of help from many people during my doctor study at Kyushu University.

First I would like to sincerely thank Professor Mikhail Svinin for all his help in my research. His insightful guidance, kind encouragement, and patience have been great support throughout my research and strengthened my goal to be a researcher.

At the same time, I would like to thank Professor Motoji Yamamoto for his great help in my research and graduation.

Also I would like to thank Professor Kazuo Kiguchi, Professor Shinji Hokamoto, and Professor Ryo Kurazume for there suggestions and comments to my thesis.

I am also thankful to Professor Ryo Kikuuwe, Professor Kenji Tahara and Professor Yasutaka Nakashima for their valuable comments.

Finally, I am deeply thankful to my family. Their encouragement and expectation support and inspire me to work hard on my research.

Contents

Acknowledgements	i
List of figures	v
List of tables	viii
1 Introduction	1
1.1 Background	1
1.1.1 Spherical rolling robots	2
1.1.2 Motion planning for underactuated systems	5
1.1.3 Control of underactuated systems	6
1.2 Major achievements	7
1.2.1 Dynamic motion planning and control for a class of spherical rolling robots	8
1.2.2 Geometric phase based motion planning for the base-free systems	8
1.2.3 Function approximation technique based control for non-square systems	9
1.3 Organization	10
2 Spherical rolling robot	11
2.1 Background	11
2.2 Mathematical Model	13
2.2.1 Contact Kinematics	14

2.2.2	Dynamic Model	14
2.3	Motion Planning	19
2.3.1	Dynamic realizability	21
2.3.2	Virtual constraint based reduction of dynamic model	23
2.3.3	Motion planning for the reduced system	26
2.3.4	Calculation of the actual input torque	33
2.3.5	Case study	36
2.4	Trajectory tracking	44
2.4.1	Adaptive trajectory tracking controller design	45
2.4.2	Case study	52
2.5	Summary	54
3	Geometric phase based motion planning	58
3.1	Background	58
3.2	Special feature of partial differentially flat systems	59
3.3	Geometric phase based motion planning	63
3.3.1	Generation of the rest-to-rest trajectory	64
3.3.2	Feasibility of the linearized solution	65
3.3.3	Computation of the parameter $q_{\max}$	66
3.4	Case study	68
3.5	Summary	75
4	Function approximation technique based control for non-square systems	77
4.1	Background	77
4.2	Function approximation technique based control	78
4.2.1	Design of controller	79
4.2.2	Proof of stability	84
4.2.3	Selection of weight matrices	85
4.3	Case study	86
4.4	Summary	96

5 Concluding remarks	98
References	104
Appendix	116

List of figures

1.1	Inner-wheel-driven rolling robot	3
1.2	Deformable type rolling robot	4
1.3	Rotor-driven rolling robot	4
1.4	Pendulum-driven rolling robot	5
2.1	Ball-pendulum system formalization.	13
2.2	Definition of basic vectors.	15
2.3	Motion planning problem.	19
2.4	Spherical robot tracing given contact path.	23
2.5	Hoop-pendulum system.	25
2.6	Motion pattern for the base variable $\theta(t)$	29
2.7	Hoop motions resulted from geometric phase based control.	31
2.8	Pendulum motions resulted from geometric phase based control.	31
2.9	Relation between θ_m and ϕ_{des} for the hoop-pendulum system under the geometric phase based motion planning algorithm. The yellow and red portions of the curve illustrate this relation respectively when ϕ_{des} is rather small and large.	32
2.10	Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).	37
2.11	Generalized coordinates of the pendulum $q_1, \dot{q}_1$ (red curve) and $q_2, \dot{q}_2$ (blue curve).	37
2.12	Generalized accelerations and torques on the pendulum: $\ddot{q}_1, \tau_1$ (red curve) and $\ddot{q}_2, \tau_2$ (blue curve).	37

2.13	Trace of the contact point on the spherical surface (left) and on the plane (right).	38
2.14	Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).	38
2.15	Generalized coordinates of the pendulum $q_1, \dot{q}_1$ (red curve) and $q_2, \dot{q}_2$ (blue curve).	39
2.16	Generalized accelerations and torques on the pendulum: $\ddot{q}_1, \tau_1$ (red curve) and $\ddot{q}_2, \tau_2$ (blue curve).	39
2.17	Trace of the contact point on the spherical surface (left) and on the plane (right).	39
2.18	Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).	40
2.19	Generalized coordinates and angular velocities of the pendulum $q_1, \dot{q}_1$ (red curves) and $q_2, \dot{q}_2$ (blue curves).	41
2.20	Generalized accelerations and torques on the pendulum: $\ddot{q}_1, \tau_1$ (red curve) and $\ddot{q}_2, \tau_2$ (blue curve).	41
2.21	Trace of the contact point on the spherical surface (left) and on the plane (right).	41
2.22	Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).	42
2.23	Generalized coordinates and angular velocities of the pendulum $q_1, \dot{q}_1$ (red curves) and $q_2, \dot{q}_2$ (blue curves).	42
2.24	Generalized accelerations and torques on the pendulum: $\ddot{q}_1, \tau_1$ (red curve) and $\ddot{q}_2, \tau_2$ (blue curve).	43
2.25	Trace of the contact point on the spherical surface (left) and on the plane (right).	43
2.26	Motion of the ball-pendulum system under non-adaptive control.	53
2.27	Motion of the ball-pendulum system under the FAT-based adaptive control.	54
3.1	A closed loop in shape variables producing a displacement in the fiber variable.	60

3.2	Motion pattern of the based variable $q_b(t)$	65
3.3	Cart-pole system	68
3.4	Motion of the cart-pole system resulted from the geometric phase based control.	70
3.5	Motion of the cart-pole system resulted from the minimum effort optimal control.	70
3.6	Planar ballbot system.	72
3.7	Motion of the ballbot system resulted from the geometric phase based control.	74
3.8	Motion of the ballbot system resulted from the minimum effort optimal control.	74
4.1	System response of double integrator under constructed controller.	88
4.2	Input signal.	88
4.3	Nonlinear system response under constructed controller.	91
4.4	Input signal.	91
4.5	Unicycle system.	92
4.6	Nonlinear system response under constructed controller.	95
4.7	Input signal.	95
5.1	A process of the neural network aided FAT approach to controller design for robots.	102
5.2	Swarm control of spherical robots for mapping of an indoor environment.	103
5.3	Definition of basic vectors with respect to the plane P	116

List of tables

2.1	Comparison for the values of the cost function J , expressed by (2.30), between the minimum effort optimal control formulation and the geometric phase based approach for the hoop-pendulum system.	32
3.1	Comparison for the values of the cost function J , expressed by (2.30), between the minimum effort optimal control formulation and the geometric phase based approach for the cart-pole system.	71
3.2	Comparison for the values of the cost function J , expressed by (2.30), between the minimum effort optimal control formulation and the geometric phase based approach for the ballbot system.	75

Chapter 1

Introduction

1.1 Background

In recent years there has been a growing interest to the motion planning and control of underactuated mechanical systems [2–5]. These systems have fewer independent actuators than the number of degrees of freedom. The need for analysis and control of underactuated mechanical systems arises in many practical applications that include the use of ships, underwater vehicles, helicopters, aircraft, satellites, space platforms, mobile robots, flexible joint robots, hyper-redundant and snake-like manipulator, walking robots, capsule robots, and hybrid machines [2–5].

The underactuated systems are superior to the fully actuated systems in terms of energy saving, cost reduction, weight reduction, and system flexibility. Also, the simple structure of underactuated system facilitates overall kinetic analysis and planning. At the same time, due to the high nonlinearity of the system, the underactuated system is complex enough to verify the effectiveness of various algorithms. In the practical implementation, a fully actuated system, when part of the driving mechanism fails, may become an underactuated system, and the control algorithm developed for underactuated systems can play a role of fault-tolerant control.

Despite these merits, from the perspective of control theory, the limitation of the input for the underactuated system leads to challenging planning and control problems: the un-

deractuated portion of the system dynamics forms a nonholonomic constraint between the output variables. The planned and the controlled trajectories for these variables need to respect this constraint.

This thesis seeks to provide motion planning and control methods for a class of under-actuated systems. Before tackling this generic problem, I first start from a specific type of underactuated system, the spherical rolling robot, to demonstrate the features of underactuated systems, as well as the corresponding difficulties in the motion planning and control. Then, based on the study of this specific problem, I extend our methods to more generic problems, motion planning and control for a class of underactuated systems. The detailed introductions for each topic (spherical rolling robot, motion planning and control for under-actuated systems) are as follows.

1.1.1 Spherical rolling robots

Spherical rolling robots are a special type of mobile robots, the image of which can be seen in many recent movies, such the Star Wars and the Jurassic World. Different from the vast majority of mobile robots that either have legs or exploit conventional multiple wheels, the spherical rolling robots, considered as single wheel vehicles, represent a new type of mechanical systems demonstrating rich and intriguing dynamics.

The first self-propelled spherical robot can be traced back to the 1900s. The rolling robot was driven by a gyroscope inside and the dynamic model was verified under experiments. However, the main issue of the study on the spherical robot is the integrability of motion equations, and no notion of control is involved in the analysis. Only in the last two decades there appeared studies in the control area on problems of motion planning and trajectory tracking of spherical robot. Meanwhile, rolling robots have also been introduced to applications such as surveillance, inspection, guarding, and disaster mitigation. Robotic toys for the education, sport and entertainment industries are also in this group.

Motivations behind the research and applications are the advantages of spherical rolling robots compared other types of mobile robots. The advantages of spherical rolling robots can be seen from the following aspects. First is their omnidirectional mobility. They can rotate along any desired direction. This ability enables spherical robots to pass around obstacles in

minimum effort. Second is its geometric symmetry. This characteristic endows the spherical robot with the robustness that it can start or stop at any position and will never fall down or be turned upside down as other mobile robot such as wheeled robot or legged robot do. Also it should be noted that all the devices of the spherical robot are covered by its outer shell. Therefore its driving mechanisms and its sensors are very well protected when rotating, bouncing or working under special condition such as uneven terrain or under water.

Encouraged by these advantages, many designs of spherical robots have been proposed. Based on the actuating schemes the spherical rolling robot can be classified into the wheel-driven type [6–8], the rotor-driven type [9–11], the pendulum-driven type [12–14] and deformable type [15–17].

For the inner wheel driven type shown in Fig. 1.1, the power that enables the robot to move is transmitted from the driving wheels to the outer shell through the friction on the contact points. However, the friction force between wheels and shell is affected dramatically under high acceleration that causes inaccuracy in control.

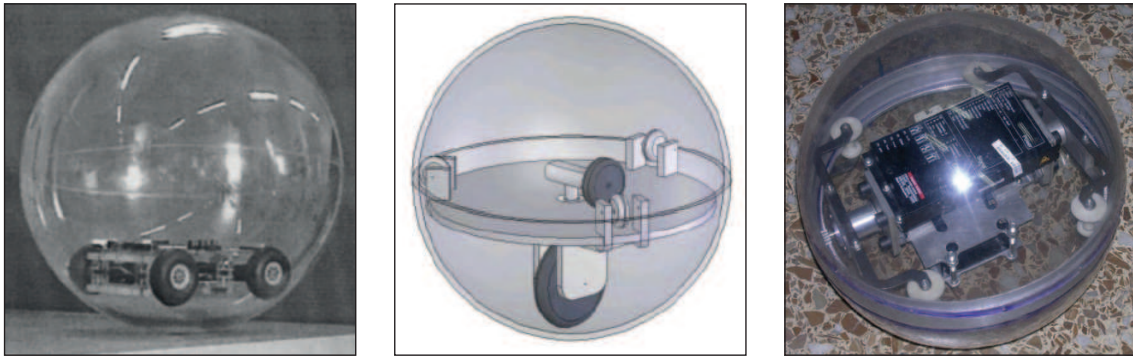


Figure 1.1: Inner-wheel-driven rolling robot

For spherical robot with deformable shape shown in Fig. 1.2, the motion of robot relies on transformation of its outer body. This can be achieved by deformation of the encompassing shell, or having environmental elements, such as wind or water, act on the body itself. However due to the complexity of the driving mechanism and difficulty of modelling, accurate motion planning and control are far from completed.

For the rotor-driven type shown in Fig. 1.3, by spinning the rotors about axis fixed in the spherical shell, the laws of conservation of angular momentum can be used to control the

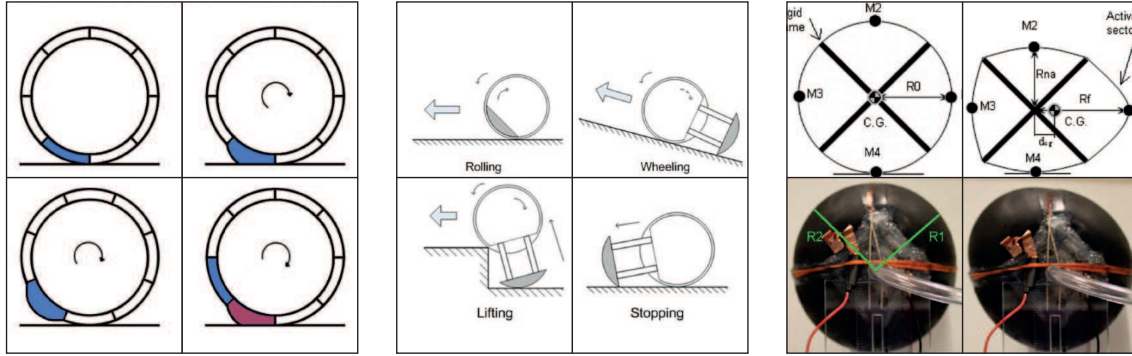


Figure 1.2: Deformable type rolling robot

movement of the robot. Under a proper placement of the rotors the center of mass of the composite system is at the geometric center of the sphere and, as a result, the gravity does not enter the motion equations. This facilitates the dynamic analysis and the design of the control system. However it should be noted that there exist certain types of motion that can not be realized by this driving mechanism.



Figure 1.3: Rotor-driven rolling robot

For the pendulum driven type shown in Fig. 1.4, consider the robot resting in equilibrium when pendulum is pointed to the ground. By swinging the pendulum, the mass distribution of the ball-pendulum will be shifted, causing the ball to roll to a new position of equilibrium. With proper timing and control, the robot can move smoothly along a given path.

While the motions of spherical rolling robots appear to be aesthetically beautiful, it is far from being clear how these motions can be planned and controlled in the full state space

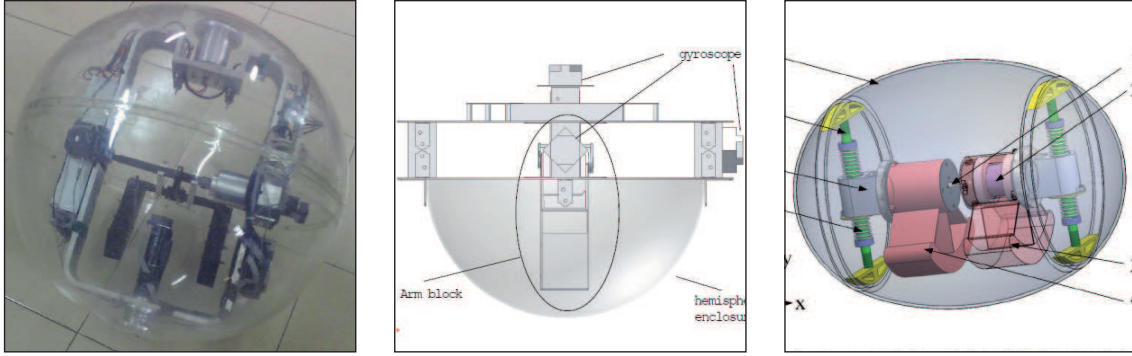


Figure 1.4: Pendulum-driven rolling robot

(including the orientation of the vehicle) if the dynamics of the actuators must be taken into account. Answering this question is very important for the construction of intelligent control systems for rolling robots.

1.1.2 Motion planning for underactuated systems

One of the important problems in the control of underactuated systems is the construction of motion planning algorithms. Typically, the motion planning problem requires to steer the system from an initial state to a given final state. Different from the fully-actuated systems, for underactuated systems, the underactuation feature forms a nonholonomic constraint between the configuration variables. This constraint makes the motion planning problem challenging due to that the generated motion trajectory needs to respect the constraint.

Due to the fact that underactuated systems are not invertible, like fully actuated systems, several approaches have been proposed, aiming at the approximate inversion. Such motion planning techniques include the nilpotent approximation [18], and a small amplitude and oscillatory control based method [19, 20]. Other than the approximation techniques, several directions have been explored focusing on the simplification of motion equations. In [21] the geometric mechanics are utilized to reduce the dynamics of a system so that it can be represented as a kinematic system. In [22] the partial feedback linearization is first applied and then a concept of natural motion is utilized to generate desired motion trajectories. Two more approaches are presented for underactuated multibody systems in [23] based on

the nonlinear coordinate transformation and the servo-constraints, respectively. Instead of approximation, other methods have been proposed for systems with special features such as the differential flatness [24,25].

Despite the achievements, motion planning techniques proposed in the literature are usually computationally expensive and thus, cannot be run in real-time on robots. Fast approaches have rarely been explored, even for planar underactuated systems. The planar underactuated systems can be classified into two types: base-fixed type and the base-free type. The base-fixed systems are those with one or more portions of the systems fixed on the ground, such as the serial robot with passive joints. Motion planning problem for this type of systems is studied in the literature [26–30]. On the other hand, the base-free type is underactuated systems whose position and orientation can be steered freely by internal shape change. It includes the cart-pole system, the planar ballbot system [31] and the hoop-pendulum system [32]. For this type of systems, which also refers to the shape-accelerated mobile robots [31,33,34], very few works on the motion planning problem exist, except for the optimal control formulation which may result to heavy computational cost. In this thesis, I propose a motion planning algorithm that provides near-optimal system performance but with less computational cost. The proposed algorithm works on a class of underactuated systems that are partially differentially flat [34].

1.1.3 Control of underactuated systems

In the literature, a great number of works on the control of underactuated mechanical systems have been proposed. Among them the most popular control methods are energy shaping [2, 36–38], partial linearization [39–41] and sliding mode control [42–47]. A review of control methods for the underactuated mechanical systems can be seen in [48]. Despite of the accomplishment however, the control of the underactuated systems is still an active research area due to that not all underactuated systems are well controlled with the above methods. Take the instance of the nonholonomic system, a special type of underactuated system which features the nonholonomic constraints.

The proposed methods have certain requirements on some special features of mechanical systems and often are not feasible for those systems that fail to satisfy the requirements.

Instead of classifying the systems by their features and developing corresponding techniques to tackle the control problem, I intend to propose a more generic method that works on a wide class of mechanical systems. This constitutes the main goal of this research.

For control systems with equal number of inputs and states, the control matrix that weights the system inputs, is square such that a state feedback law is easily defined. However in most cases, the control matrix for mechanical systems represented in the state space form, is not square due to that the number of inputs and of states are not always matched. These systems refer to the non-square systems. In this paper, I place emphasis on a class of non-square systems where number of inputs is less than that of states.

The control problem for non-square systems is difficult because the control matrix is not invertible. The most straightforward way is to square the system by applying the pseudo inverse. However, the idea of using the pseudo inverse is to approximate the original non-square system by a square system, and the performance of this approximation requires certain properties of the pseudo inverse. These properties of the pseudo inverse for its application on the control problems of non-square system are studied by [49]. Other studies on the control problem of non-square systems either apply fuzzy control [50] or are only suitable under certain situations [51, 52]. In this paper, I propose an exact control algorithm for systems expressed by non-square state space form. The control algorithm is constructed based on the function approximation technique, inspired by its application on the adaptive control [53–56].

1.2 Major achievements

Corresponding to the three topics of my research (spherical robots, motion planning and control for underactuated systems), the major achievements can be summarized as follows. Firstly, I proposed a motion planning and a control algorithms for spherical rolling robots. Secondly, I extended the proposed motion planning algorithm, based on a geometric phase approach, to a class of underactuated systems, which I call the base-free systems. Thirdly, similar to motion planning, I extended the proposed control algorithm, based on the function approximation technique (FAT), to non-square systems, a broader class of underactuated

systems. In what follows, these achievements are illustrated respectively, in details.

1.2.1 Dynamic motion planning and control for a class of spherical rolling robots

A virtual constraint based decoupling algorithm was first proposed which reduced a rest-to-rest motion planning problem for ball-pendulum system to that for the hoop-pendulum system, a planar case of the ball-pendulum. The reduced planning problem could be solved through optimal control formulation seeking for minimum effort. However the optimal control based motion planning may be sensitive to the given boundary conditions and may lead to a heavy computational cost. Instead of optimal control formulation, I proposed a geometric phase based motion planning algorithm which provides similar results compared with the optimal control formulation but with less computational cost, and thus more suitable for realtime applications.

For the dynamic control problem, an adaptive trajectory tracking controller for the ball-pendulum system was constructed under a backstepping framework. The backstepping process combined two steps. A feedback velocity control input, denoted by angular velocity of the spherical shell, was first designed for the kinematic steering system. Then the specific robot dynamics were taken into account to convert a steering command into control torques acting on the pendulum. Noting that in either step, the subsystem to be controlled is underactuated, I proposed an FAT based controller and tested it under simulations. The proposed FAT based controller was proved to be adaptive to time varying uncertainties such as un-modeled system dynamics and external perturbations.

1.2.2 Geometric phase based motion planning for the base-free systems

A rest-to-rest motion planning problem was studied for a class of underactuated systems. I first classified the planar underactuated systems into two types. One is the base-fixed systems that one or more portions of the systems are fixed on the ground, including the serial robot with passive joints. The other is the base-free type whose position and orientation can be steered freely by internal shape change, such as the cart-pole system, the ballbot system

and the hoop-pendulum system. A geometric phase based motion planning algorithm was proposed for the base-free type systems.

After the partial feedback linearization, this type of systems are commonly partially differentially flat and the base variable is the flat output. That is, the derivatives of the fiber variable of these systems can be represented purely by the base variable and its derivatives. It leads to a property that the linearized model of these systems is the n -th order integrator. Based on this property, the motion planning algorithm has been constructed as follows.

First, the derivative of Beta-function weighted by a constant parameter was selected as a candidate for the base variable due to that it is shown to be the optimal solution for the linearized system. Its feasibility was proved by showing that the corresponding fiber variable can satisfy the given boundary conditions. Finally the parameter of the derivative of Beta-function was adjusted such that a desired shift for the fiber variable can be obtained. Under the proposed motion planning algorithm, rest-to-rest trajectories for the configuration variables were generated with similar system performance but with less computational cost, compared with the optimal control formulation.

1.2.3 Function approximation technique based control for non-square systems

A generic control method has been proposed for the non-square systems where the number of system inputs is not equal to that of the states. Note that non-square systems are a broader class of underactuated systems. By introducing an auxiliary input the dimension of which equals that of the system state, the non-square system was restructured in the form of the combination of a square system and the variation from the original non-square system, which is treated as system uncertainties.

In the control process, the uncertainty term was replaced by its approximation as a chosen basis function weighted by constant parameters to be determined. These unknown plant parameters were estimated at each instant, denoted by the adjustable control parameters using a defined update law. Thus the influence to the control process caused by the variation between the auxiliary square system and the original non-square system can be eliminated.

The asymptotic stability was established for the closed loop system formulated by the

non-square system and the constructed controller. The feasibility of the proposed control method was verified under simulations for the point stabilization problem of a linear system, a nonlinear underactuated system, and a nonholonomic system, respectively.

1.3 Organization

The rest of this dissertation is organized as follows. Chapter 2 provides the motion planning and trajectory tracking control algorithms for the ball-pendulum system. Chapter 3 presents a geometric phase based motion planning algorithm for a class of underactuated systems, the base-free systems. Chapter 4 proposes a function approximation technique based control method for non-square systems. Chapter 5 provides concluding remarks and future plan.

Chapter 2

Spherical rolling robot

2.1 Background

In recent years there has been a growing interest to the study of spherical rolling robot. Different from the vast majority of mobile robots that either have legs or exploit conventional multiple wheels, the rolling robots, considered as single wheel vehicles, represent a new type of mechanical systems demonstrating rich and intriguing dynamics. Compared with traditional mobile robots, spherical rolling robots are geometrically symmetric (cannot be turned upside down), omnidirectional and more robust (all devices are covered by a solid outer shell, including driving unit, sensors, etc.).

Encouraged by these advantages, many designs of spherical robots have been proposed. Based on the actuating schemes the spherical rolling robot can be classified into the wheel-driven type [6–8], the rotor-driven type [9–11], the pendulum-driven type [12–14] and deformable type [15–17]. My work is related to the pendulum-driven type. We study a class of spherical rolling robots driven by a 2DOF pendulum, which we call the ball-pendulum systems. The ball-pendulum system is selected due to the following reasons. Firstly, the driven mechanism is simple. Secondly, this type of diving mechanism can realize motion of the sphere along any shape of smooth contact curves. Thirdly, the control for this type of spherical robot is still an open problem.

One of the important research problems for control of mobile robot is to steer the robot

from an initial configuration to the final configuration. For the spherical rolling robot we need not only to steer the position of the sphere but also the orientation of the sphere. The control for the orientation is important when it is necessary to attach some devices such as sensors or camera in or out of the shell, or when it is required to know where the robot is heading. The steering problem seems to be a normal point to point stabilization problem which can be solved by constructing a continuous feedback. However, spherical rolling robots are special because it is a nonholonomic system. According to the Brockett theory [57], there exists no continuous feedback for local asymptotic stabilization of an equilibrium point. Instead of apply direct feedback methods, we plan to solve the steering problem step by step.

The main process is as followed. One can first design a path on the ground, connecting the start and the goal configurations of the spherical shell, by using conventional kinematic planners [71–75]. With a given feasible path, one can plan the motion trajectories for both the orientation and the position of the sphere when it is tracing this path. This step is called motion planning. In the real application however, the actual motion of the ball-pendulum system may deviate from the desired trajectory due to unexpected disturbances such as the friction or uneven terrain. To deal with this problem, one needs to construct a feedback control law such that the actual motion of the spherical rolling robot will converge to the desired. We call this step the trajectory tracking. Through these two steps, one can steer the ball-pendulum system from initial to final configurations in a rest-to-rest mode even under perturbations.

The rest of this chapter is organized as follows. In Section 2.2, one provides the contact kinematics and dynamic model for the ball-pendulum system. In Section 2.3, a motion planning algorithm is proposed that enables the spherical robot to trace given contact path on the ground in a rest-to-rest mode. In Section 2.4, a feedback control method is developed such that the actual motion trajectory of the ball-pendulum system can converge to the trajectory designed in Section 2.3 under perturbation. Finally, a summary is given in Section 2.5.

2.2 Mathematical Model

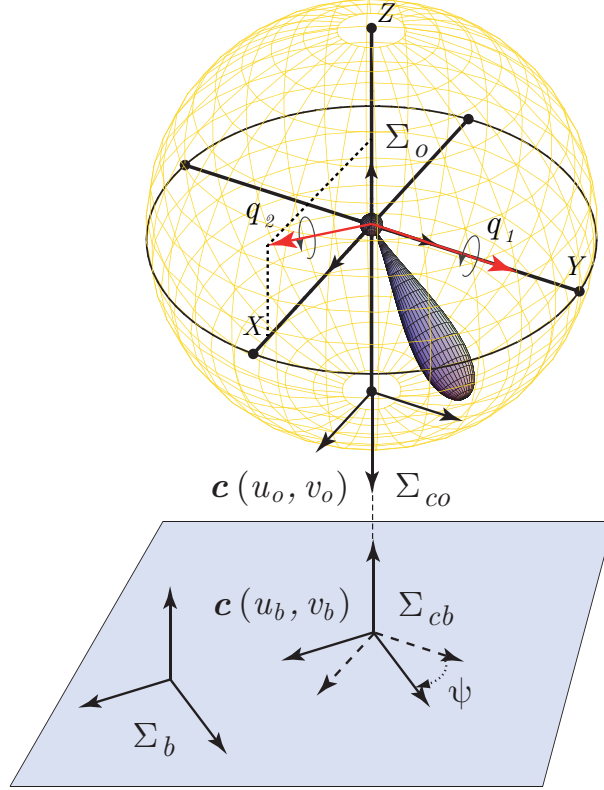


Figure 2.1: Ball-pendulum system formalization.

Consider a rolling robot composed of a homogenous spherical shell actuated by a pendulum. It is assumed that the pendulum is mounted at the geometric center of the shell as shown in Fig. 2.1 and is actuated by two motors.

To describe the contact kinematics, we introduce the following coordinate frames: Σ_o is a frame fixed at the geometric center of the sphere, Σ_b is a frame fixed at the contact plane. In addition, at the contact point we introduce the contact frame of the object Σ_{co} , and the contact frame of the plane, Σ_{cb} , which is parallel to Σ_b .

The contact coordinates are given by the angles u_o and v_o , describing the contact point on the sphere, and u_b and v_b , describing the contact point on the plane, and by the contact angle ψ which is defined as the angle between the x -axis of Σ_{co} and Σ_{cb} .

2.2.1 Contact Kinematics

The position of the contact point on the sphere, denoted by the vector ρ_c and expressed in the axis of Σ_o , is parameterized as

$${}^o\rho_c = \mathbf{R}_y(u_o)\mathbf{R}_x(v_o) \begin{bmatrix} 0 \\ 0 \\ -R \end{bmatrix} = R \begin{bmatrix} -\sin u_o \cos v_o \\ \sin v_o \\ -\cos u_o \cos v_o \end{bmatrix}, \quad (2.1)$$

where R is the radius of the sphere. In this parameterization the origin is placed at the south pole of the sphere. The parameterization defines the Gaussian frame Σ_{co} whose orientation relative to Σ_o is given by the matrix ${}^o\mathbf{R}_{co}(u_o, v_o) = \mathbf{R}_y(u_o)\mathbf{R}_x(v_o)\mathbf{R}_y(\pi)$, composed of the elementary rotations. Note that instead of the classical Euler angles and their variations we define the orientation of the sphere in terms of the contact coordinates u_o, v_o and ψ .

Let $\omega_o = (\omega_x, \omega_y, \omega_z)$ be the angular velocity of the frame Σ_o , defined in projections onto the axis of the base frame Σ_b . The evolution of the contact coordinates is described by Montana's equations [58], which in our parameterization take the following form

$$\begin{bmatrix} \dot{u}_b \\ \dot{v}_b \\ \dot{u}_o \\ \dot{v}_o \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & R & 0 \\ -R & 0 & 0 \\ -\sin \psi / \cos v_o & -\cos \psi / \cos v_o & 0 \\ -\cos \psi & \sin \psi & 0 \\ -\sin \psi \tan v_o & -\cos \psi \tan v_o & -1 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (2.2)$$

2.2.2 Dynamic Model

In what follows, the derivation of the dynamic models will be conducted with the use of D'Alembert's principle. Let r_o be the vector of the origin of the frame Σ_o (see Fig. 2.2), v_o its absolute velocity, and ω_o the angular velocity of the frame Σ_o . The virtual displacement of the origin of Σ_o is denoted by δr_o , and the vector of infinitesimal rotation of Σ_o with

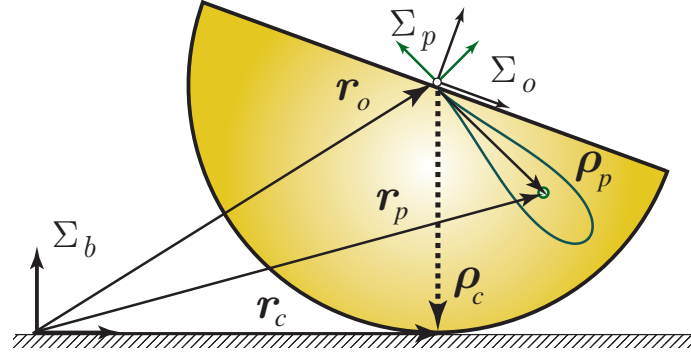


Figure 2.2: Definition of basic vectors.

respect Σ_b is denoted by $\delta\pi_o$. Unless otherwise specified all the vector and tensor quantities are assumed to be expressed in the axes of the inertial frame Σ_b .

The frame Σ_p is attached to the pendulum and its origin coincides with that of Σ_o . Let ρ_p be the vector of the center of mass of the pendulum with respect to the frame Σ_o , v_p its absolute velocity, and ω_{rel} the angular velocity of the frame Σ_p with respect to Σ_o . The vector for the infinitesimal rotation of Σ_p with respect to Σ_o is denoted by $\delta\pi_{rel}$. The kinematics of the pendulum are defined by the following relationships

$$\delta\pi_p = \delta\pi_o + \delta\pi_{rel}, \quad (2.3)$$

$$\omega_p = \omega_o + \omega_{rel}, \quad (2.4)$$

$$\dot{\omega}_p = \dot{\omega}_o + \omega_o \times \omega_{rel} + \dot{\omega}_{rel}, \quad (2.5)$$

$$v_p = v_o + \omega_p \times \rho_p, \quad (2.6)$$

$$\dot{v}_p = \dot{v}_o + \dot{\omega}_p \times \rho_p + \omega_p \times (\omega_p \times \rho_p). \quad (2.7)$$

For system under consideration free from the rolling constraint D'Alembert's principle can be written down as follows [59]

$$\begin{aligned} \delta r_o \cdot \left(m_o(\dot{v}_o - g) + m_p(\dot{v}_p - g) \right) + \delta\pi_o \cdot \left(J_o \dot{\omega}_o + \omega_o \times J_o \omega_o \right) \\ + \delta\pi_p \cdot \left(m_p \rho_p \times (\dot{v}_o - g) + J_p \dot{\omega}_p + \omega_p \times J_p \omega_p \right) = \delta W, \end{aligned} \quad (2.8)$$

where g is the gravity vector, m is the mass of the spherical shell, and $J_o = \frac{2}{3}mR^2 I$ its

inertia tensor with respect to the origin of Σ_o , m_p is the mass of the pendulum, and $\mathbf{J}_p$ is the inertia tensor of the pendulum with respect to the origin of Σ_p . The virtual work $\delta W = \delta \boldsymbol{\pi}_{\text{rel}} \cdot \mathbf{M}$, where $\mathbf{M}$ is the moment generated by the actuators.

The derivation of dynamic equations is conducted in several steps by transforming (2.8) to independent variations. First, we transform the virtual displacement $\delta \mathbf{r}_o$ in (2.8) to the point of contact, defined by the vector $\mathbf{r}_c$. The reason is to introduce variations of quasi-coordinates that would vanish by virtue of the non-holonomic constraints [60,61]. Note that $\mathbf{r}_c = \mathbf{r}_o + \boldsymbol{\rho}_c$ and therefore $\delta \mathbf{r}_c = \delta \mathbf{r}_o + \delta \boldsymbol{\pi}_o \times \boldsymbol{\rho}_c$, where $\boldsymbol{\rho}_c$ is the vector emanated from the center of mass of the sphere to the contact point. The rolling constraints, formulated in terms of the variations of the quasi-coordinates, can be written down as $\delta \mathbf{r}_c = 0$, and therefore

$$\delta \mathbf{r}_o = -\delta \boldsymbol{\pi}_o \times \boldsymbol{\rho}_c. \quad (2.9)$$

Upon substituting (2.9) into (2.8) and transforming $\delta \boldsymbol{\pi}_p$ to $\delta \boldsymbol{\pi}_o$ and $\delta \boldsymbol{\pi}_{\text{rel}}$ by using (2.3), one obtains

$$\begin{aligned} & \delta \boldsymbol{\pi}_o \cdot \left(\mathbf{J}_o \dot{\boldsymbol{\omega}}_o + \mathbf{J}_p \dot{\boldsymbol{\omega}}_p + \boldsymbol{\omega}_p \times \mathbf{J}_p \boldsymbol{\omega}_p - \boldsymbol{\rho}_c \times (m \dot{\mathbf{v}}_o + m_p \dot{\mathbf{v}}_p) + m_p \boldsymbol{\rho}_p \times (\dot{\mathbf{v}}_o - \mathbf{g}) \right) + \\ & \delta \boldsymbol{\pi}_{\text{rel}} \cdot \left(m_p \boldsymbol{\rho}_p \times (\dot{\mathbf{v}}_o - \mathbf{g}) + \mathbf{J}_p \dot{\boldsymbol{\omega}}_p + \boldsymbol{\omega}_p \times \mathbf{J}_p \boldsymbol{\omega}_p - \mathbf{M} \right) = 0. \end{aligned} \quad (2.10)$$

Next, we take into account the rolling constraint $\mathbf{v}_o + \boldsymbol{\omega}_o \times \boldsymbol{\rho}_c = \mathbf{0}$ and establish the velocity and acceleration of the spherical shell:

$$\mathbf{v}_o = -\boldsymbol{\omega}_o \times \boldsymbol{\rho}_c, \quad \dot{\mathbf{v}}_o = -\dot{\boldsymbol{\omega}}_o \times \boldsymbol{\rho}_c. \quad (2.11)$$

By expressing $\mathbf{v}_p$ and $\dot{\mathbf{v}}_p$ in (2.6, 2.7) through, respectively, $\boldsymbol{\omega}_o$, $\boldsymbol{\omega}_{\text{rel}}$, and $\dot{\boldsymbol{\omega}}_o$, $\dot{\boldsymbol{\omega}}_{\text{rel}}$ with the use of (2.4, 2.5) and (2.11), one converts (2.10) to the following form

$$\delta \boldsymbol{\pi}_o \cdot \left(\overline{\mathbf{H}}_{oo} \dot{\boldsymbol{\omega}}_o + \overline{\mathbf{H}}_{op} \dot{\boldsymbol{\omega}}_{\text{rel}} + \overline{\mathbf{h}}_o \right) + \delta \boldsymbol{\pi}_{\text{rel}} \cdot \left(\overline{\mathbf{H}}_{po} \dot{\boldsymbol{\omega}}_o + \overline{\mathbf{H}}_{pp} \dot{\boldsymbol{\omega}}_{\text{rel}} + \overline{\mathbf{h}}_p - \mathbf{M} \right) = 0, \quad (2.12)$$

where

$$\begin{aligned}\overline{\mathbf{H}}_{oo} &= \mathbf{J}_c + \mathbf{J}_p + m_p(\hat{\boldsymbol{\rho}}_p\hat{\boldsymbol{\rho}}_c + \hat{\boldsymbol{\rho}}_c\hat{\boldsymbol{\rho}}_p), \quad \overline{\mathbf{H}}_{op} = \mathbf{J}_p + m_p\hat{\boldsymbol{\rho}}_c\hat{\boldsymbol{\rho}}_p, \quad \overline{\mathbf{H}}_{po} = \overline{\mathbf{H}}_{po}^\top, \quad \overline{\mathbf{H}}_{pp} = \mathbf{J}_p, \\ \overline{\mathbf{h}}_o &= \boldsymbol{\omega}_p \times \mathbf{J}_p \boldsymbol{\omega}_p - m_p(\boldsymbol{\rho}_p \times \mathbf{g} + \boldsymbol{\rho}_c \times (\boldsymbol{\omega}_p \times (\boldsymbol{\omega}_p \times \boldsymbol{\rho}_p))), \\ \overline{\mathbf{h}}_p &= \boldsymbol{\omega}_p \times \mathbf{J}_p \boldsymbol{\omega}_p - \boldsymbol{\rho}_p \times m_p \mathbf{g}.\end{aligned}$$

Here in the equations, $\mathbf{J}_c = \mathbf{J}_o - (m + m_p)\hat{\boldsymbol{\rho}}_c\hat{\boldsymbol{\rho}}_c$ is the inertia tensor of the rolling robot with respect to the contact point, and by hat we denote the skew-symmetric operator such that $\mathbf{a} \times \mathbf{b} \equiv \hat{\mathbf{a}}\mathbf{b}$.

Note that variation $\delta\boldsymbol{\pi}_{\text{rel}}$ is not independent as the pendulum is assumed to be driven by two actuators. The relative position of the pendulum with respect to Σ_o can be described by two angles, q_1 and q_2 , mimicing the contact coordinates u_o and v_o . Under this definition, the pendulum is rotated along the axes defined (in the frame Σ_o) by the unit vectors $\mathbf{e}_1 = (0, 1, 0)$ and $\mathbf{e}_2 = (\cos q_1, 0, -\sin q_1)$, and the orientation of Σ_p with respect to Σ_o is defined by the matrix ${}^o\mathbf{R}_p = \mathbf{R}_y(q_1)\mathbf{R}_x(q_2)$.

Let $\mathbf{q} = (q_1, q_2)$ be the vector of the generalized coordinates of the pendulum and $\boldsymbol{\tau} = (\tau_1, \tau_2)$ the vector of the driving torques. The relative kinematic quantities and the moment generated by the actuators are defined as

$$\delta\boldsymbol{\pi}_{\text{rel}} = \mathbf{E}\delta\mathbf{q}, \quad \boldsymbol{\omega}_{\text{rel}} = \mathbf{E}\dot{\mathbf{q}}, \quad \dot{\boldsymbol{\omega}}_{\text{rel}} = \mathbf{E}\ddot{\mathbf{q}} + \mathbf{e}, \quad \mathbf{M} = \mathbf{E}\boldsymbol{\tau}, \quad (2.13)$$

where $\mathbf{E} = [\mathbf{e}_1 | \mathbf{e}_2]$, and $\mathbf{e} = (\mathbf{e}_1 \times \mathbf{e}_2)\dot{q}_1\dot{q}_2$. Now, by plugging (2.13) into (2.12) and treating $\delta\boldsymbol{\pi}_o$ and $\delta\mathbf{q}$ as the independent variations, one finally obtains the dynamic equations for the rolling carrier and the pendulum:

$$\begin{bmatrix} \mathbf{H}_{oo} & \mathbf{H}_{op} \\ \mathbf{H}_{po} & \mathbf{H}_{pp} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}}_o \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{h}_o \\ \mathbf{h}_p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix}, \quad (2.14)$$

where

$$\begin{aligned} \mathbf{H}_{oo} &= \overline{\mathbf{H}}_{oo}, \mathbf{H}_{op} = \overline{\mathbf{H}}_{op} \mathbf{E}, \mathbf{H}_{po} = \mathbf{H}_{op}^\top, \mathbf{H}_{pp} = \mathbf{E}^\top \mathbf{J}_p \mathbf{E}, \\ \mathbf{h}_o &= \overline{\mathbf{h}}_o + \overline{\mathbf{H}}_{op} \mathbf{e}, \mathbf{h}_p = \overline{\mathbf{h}}_p + \mathbf{E}^\top \overline{\mathbf{H}}_{pp} \mathbf{e}. \end{aligned}$$

Note that the number of equations in the dynamic model (2.14) is minimal since the reaction forces associated with the rolling constraint have been eliminated by reducing the motion equations to the independent variations. By converting the dynamic equations (2.14) to the 1st order form and combining them with the kinematic equations (2.2), one obtains the state space model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, with 12-dimensional state vector $\mathbf{x} = (u_b, v_b, u_o, v_o, \psi, \omega_x, \omega_y, \omega_z, q_1, q_2, \dot{q}_1, \dot{q}_2)$ and 2-dimensional control input $\mathbf{u} = \boldsymbol{\tau}$.

2.3 Motion Planning

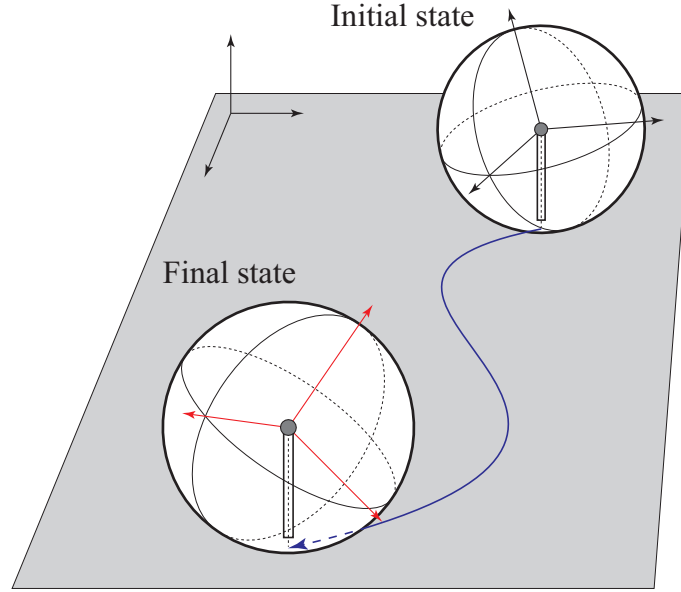


Figure 2.3: Motion planning problem.

One of the important problems in the control of spherical rolling robots is the construction of motion planning algorithms for steering the robot from an initial configuration of the spherical shell, defined by its position and orientation, to a final one. Typically, the steering is organized in the rest-to-rest mode. In literature, this motion planning problem is considered mostly for a ball-plate system, with the ball driven externally by two parallel plates, and solved in kinematic formulation [71–75]. For the rolling robots actuated by internal motors, the control system based on the kinematic planner can operate in the quasi-static mode [10, 65, 76, 77], presuming relatively slow movements.

To realize faster and more accurate movements for the ball-pendulum system, it is necessary to extend the motion planning algorithms from the kinematic to dynamic domain. A generic way is to plan motion based on the full model which takes into account both the kinematics and dynamics of the robot. The full model is described by twelve states and two control inputs. It is highly nonlinear and features both the non-holonomic rolling constraint and the gravitational drift term. As a result, only generic techniques such as optimal control

can be employed for solving the motion planning problem. However, they come with a high computational cost.

To avoid the difficulties of motion planning based on the full model, one can try to decouple the problem into kinematic and dynamic stages and solve them separately. Specifically, one can first design a feasible path, connecting the start and the goal configurations of the spherical shell, by using conventional kinematic planners [71–75]. Then, one can try to generate timing control laws, based on the dynamic model of the spherical shell and the actuators, tracing the feasible path in rest-to-rest mode. However, the decoupling approach does not necessarily work for all types of rolling robots and it depends on the type of actuation. For instance, it was shown that for rolling robots actuated by two rotors not any kinematically feasible motion is dynamically realizable [78].

It should be noted that motion control in dynamic formulation was addressed in [7, 79–82]. In [79, 82], only partial cases (rolling along a straight line or a circle) were considered. In [7, 80], the motion of the spherical robot along complex curves was decomposed into forwarding and turning. However, this decomposition is not accurate because the effect of the dynamic interaction between these two basic movements was neglected in the dynamic model. Similarly, a simplified decoupled model, ignoring dynamic interaction between rotation of the sphere along the transversal and longitudinal directions, was used in [81].

In Section 2.3.1, we show that any feasible kinematic motion of the sphere, considered in the class of pure rolling (no spinning) motion, is dynamically realizable in the ball-pendulum system. As a result, the motion planning problem can be decomposed into kinematic and dynamic stages. Moreover, by introducing a virtual constraint corresponding to pure rolling mode, the dynamic model is reduced to just two equations describing motion of an underactuated hoop-pendulum system along the contact path. We show that the hoop-pendulum system is controllable and this implies the controllability of the ball-pendulum system since the sphere rolling without spinning is controllable [71]. Next, we show that the form of the dynamic equations of the hoop-pendulum system is invariant with respect to the feasible contact path. This implies that a wide variety of kinematically feasible motions (pure rolling movements) can be dynamically realized by a single underactuated system. To plan rest-to-rest movements of the hoop-pendulum system, we use a geometric phase approach

and show its connection to and comparison with the optimal control problem where the minimum control effort is taken as the performance index.

Motion planning for the ball-pendulum system requires to steer the spherical robot from an initial configuration to a final configuration. The problem can be stated as finding a trajectory $\mathbf{x}(t)$, $t \in [0, T]$, given the start state $\mathbf{x}(0) = \mathbf{x}_s$ and the final state $\mathbf{x}(T) = \mathbf{x}_f$. It is assumed that the system is at rest at the start and final configurations, i.e. $\boldsymbol{\omega}_o(0) = \boldsymbol{\omega}_o(T) = \mathbf{0}$, $\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}(T) = \mathbf{0}$. Also, since the pendulum at the rest configurations is pointed downward, we have $q_1(0) = u_o(0) = 0$, $q_1(T) = u_o(T) = 0$, $q_2(0) = v_o(0) = 0$, $q_2(T) = v_o(T) = 0$. This reconfiguration problem is illustrated in Fig. 2.3.

Note that the kinematic (2.2) and dynamic (2.14) models are coupled through the angular velocity of the spherical shell $\boldsymbol{\omega}_o(t)$. Also note that the twelve-dimensional state space model, obtained from (2.2, 2.14), is a nonlinear control-affine system with the drift term and two control inputs. Motion planning for such a system in the general settings can be cumbersome and require the use of generic techniques such as optimal control.

2.3.1 Dynamic realizability

An alternative approach to the motion planning problem would be to decompose it and solve it separately at the kinematic and dynamic levels. At the kinematic level, one constructs a feasible path and find $\boldsymbol{\omega}_o(t)$ based on the kinematic model (2.2). At the dynamic level, one finds the trajectory of the pendulum and establishes the timing control law for the actuators tracing the kinematically feasible path.

However this decoupling approach does not always work since not any angular velocity of the spherical shell $\boldsymbol{\omega}_o(t)$ is dynamically realizable. The planned trajectory for unrealizable $\boldsymbol{\omega}_o(t)$ is invalid due to that physically unrealizable motions cannot be generated by the driving mechanisms. As far as we know, the dynamic realizability for the ball-pendulum system has not been addressed in the motion planning literature. In this chapter, we first define the dynamical realizability condition mathematically. Then, we propose a dynamically realizable decoupling technique such that the motion planning problem can be decomposed into kinematic and dynamic stages, and thus solved separately. This is the novelty of our work in the motion planning for the ball-pendulum system.

Referring to the dynamic model for the ball-pendulum system (2.14), if $\mathbf{h}_n$ is a vector parameterizing the null space of matrix $\mathbf{H}_{po}$, one has

$$\mathbf{h}_n \cdot (\mathbf{H}_{oo}\dot{\boldsymbol{\omega}}_o + \mathbf{h}_o) = 0, \quad (2.15)$$

which can be called the condition of dynamic realizability. Hence, the kinematical motion of the sphere, represented by $\boldsymbol{\omega}_o(t)$ can be accomplished by the actuators only when the dynamic realizability condition is satisfied.

A notable unrealizable case is when the ball-pendulum system is at rest: $(\boldsymbol{\omega}_o, \mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$, where the dynamic realizability condition (2.15) does not hold true unless the magnitude of the angular acceleration of the sphere $\|\dot{\boldsymbol{\omega}}_o\|$ or its z -axis component $\dot{\omega}_z$ equals to zero. This can be proved by the substitution of $(\boldsymbol{\omega}_o, \mathbf{q}, \dot{\mathbf{q}}) = \mathbf{0}$ into (2.15), where the corresponding $\mathbf{h}_n = (0, 0, 1)$. Through simple calculations, one can show that the left hand side of (2.15) equals to zero only when $\|\dot{\boldsymbol{\omega}}_o\| = 0$ or $\dot{\omega}_z = 0$. For instance, a straight line motion of the sphere, expressed by $\boldsymbol{\omega}_o = \frac{1}{\sqrt{3}}(1, 1, 1)\sin(t)$, is not realizable by the 2DOF pendulum in rest-to-rest mode, due to that at a rest configuration, the condition of the dynamic realizability (2.15) is not satisfied.

Noting that instantaneous spinning ω_z may lead to unrealizable motions of the spherical carrier, we adopt only pure rolling ($\omega_z = 0$) in our motion planning algorithm. In what follows, we show that a class of motions corresponding to pure rolling of the carrier, actuated by a two DOF pendulum, is dynamically realizable. To show the realizability, instead of checking the null space of matrix $\mathbf{H}_{po}$, we develop a dynamic model of the ball-pendulum system, reduced by the pure rolling constraint and establish its controllability. The reduced dynamic model is an underactuated system, a hoop-pendulum system, described by two differential equations with single control input. It is invariant to the shape of the contact path, and the dynamic motion planner can be constructed in a unified manner.

As a result, motion planning for the ball-pendulum system can be decoupled into kinematic and dynamic levels. In the kinematic level, one designs a path on the contact plane connecting the start and the goal points, such that if the sphere rolls along this path, it can reach the goal with a desired orientation. In the dynamic level, one constructs torques on the pendulum, the actuator, to generate the time trajectories for the rolling motion of the

ball-pendulum system along the designed path. The kinematic problem can be solved by planners available in the literature [71–75], while the dynamic problem is tackled with a decoupling based algorithm we developed, the details of which are demonstrated in the following sections.

2.3.2 Virtual constraint based reduction of dynamic model

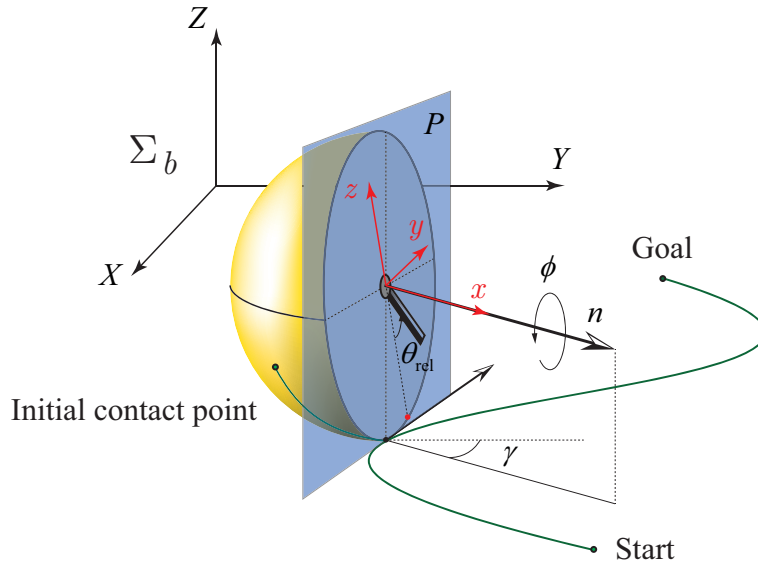


Figure 2.4: Spherical robot tracing given contact path.

Assume the spherical shell rolls without spinning along a path constructed by the kinematic planner as shown in Fig. 2.4. Let ϕ be the angular distance of the center of the sphere such that the current length of the contact path on the plane is $2\pi\phi$. Let also $\mathbf{n} = (\sin \gamma, \cos \gamma, 0)$ be the vector normal to the contact path on the plane, where γ is the angle between vector $\mathbf{n}$ and the Y -axis of the coordinate frame Σ_b . Since the angular velocity of the sphere under rolling without spinning does not have z -component, the virtual constraint corresponding to pure rolling can be formalized as

$$\delta \pi_o = \mathbf{n} \delta \phi, \quad \omega_o = \mathbf{n} \dot{\phi}, \quad \dot{\omega}_o = \mathbf{n} \ddot{\phi} + \dot{\mathbf{n}} \dot{\phi}. \quad (2.16)$$

Let P be the vertical plane instantaneously tangent to (moving along) the contact path as shown in Fig. 2.4. To realize the pure rolling motion of the sphere, the center of mass of the pendulum must (instantaneously) move in the plane P . If the pendulum is axisymmetric with respect to the line through its center of mass, the angular velocity of the pendulum must be directed along the normal vector $\mathbf{n}$. Let θ_{rel} be the angle of rotation of the pendulum in the plane P through the normal vector $\mathbf{n}$. The virtual constraint imposed on the motion of the pendulum can be formalized as

$$\delta \boldsymbol{\pi}_{\text{rel}} = \mathbf{n} \delta \theta_{\text{rel}}, \quad \boldsymbol{\omega}_{\text{rel}} = \mathbf{n} \dot{\theta}_{\text{rel}}, \quad \dot{\boldsymbol{\omega}}_{\text{rel}} = \mathbf{n} \ddot{\theta}_{\text{rel}} + \dot{\mathbf{n}} \dot{\theta}_{\text{rel}}. \quad (2.17)$$

By plugging the virtual constraints (2.16, 2.17) into (2.12) and treating $\delta \phi$ and $\delta \theta_{\text{rel}}$ as the independent variations, one obtains two equations describing the reduced dynamics:

$$\left(\mathbf{n}^\top \overline{\mathbf{H}}_{oo} \mathbf{n} \right) \ddot{\phi} + \left(\mathbf{n}^\top \overline{\mathbf{H}}_{op} \mathbf{n} \right) \ddot{\theta}_{\text{rel}} + \mathbf{n}^\top \left(\overline{\mathbf{H}}_{oo} \dot{\mathbf{n}} \dot{\phi} + \overline{\mathbf{H}}_{op} \dot{\mathbf{n}} \dot{\theta}_{\text{rel}} + \overline{\mathbf{h}}_o \right) = 0, \quad (2.18)$$

$$\left(\mathbf{n}^\top \overline{\mathbf{H}}_{po} \mathbf{n} \right) \ddot{\phi} + \left(\mathbf{n}^\top \overline{\mathbf{H}}_{pp} \mathbf{n} \right) \ddot{\theta}_{\text{rel}} + \mathbf{n}^\top \left(\overline{\mathbf{H}}_{po} \dot{\mathbf{n}} \dot{\phi} + \overline{\mathbf{H}}_{pp} \dot{\mathbf{n}} \dot{\theta}_{\text{rel}} + \overline{\mathbf{h}}_p \right) = \tau_\theta, \quad (2.19)$$

where $\tau_\theta = \mathbf{n}^\top \mathbf{M}$ represents the control torque reduced to the coordinate θ . It is the projection of the moment $\mathbf{M} = \mathbf{E} \boldsymbol{\tau}$ generated by the control torques $\boldsymbol{\tau}$ onto the line through the vector $\mathbf{n}$.

Note that equation (2.18) describes motion of the rolling carrier along the given contact path, which can be called a virtual hoop, while equation (2.19) does the motion of the pendulum in the moving plane P . To distinguish the reduced dynamic system from the main one, we call it the hoop-pendulum system and illustrate it by Fig. 2.5. The red dot on the hoop is its initial contact point to the ground. One revolution of the initial contact point implies a $2\pi R$ linear displacement for the center of the hoop. Note that at the starting configuration ($\phi(0) = 0$), the initial contact point touches the ground.

As shown in Appendix A, the calculation of terms in (2.18, 2.19) under the assumption that the pendulum is a beam with uniform mass distribution yields the following represen-

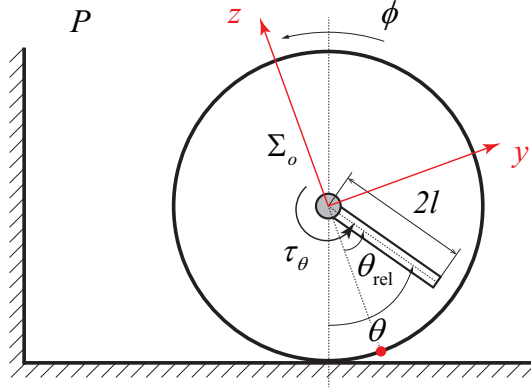


Figure 2.5: Hoop-pendulum system.

tation

$$\begin{aligned} & \left(J_c + J_p - 2m_p R l \cos(\phi + \theta_{\text{rel}}) \right) \ddot{\phi} + \left(J_p - m_p R l \cos(\phi + \theta_{\text{rel}}) \right) \ddot{\theta}_{\text{rel}} \\ & + m_p l \sin(\phi + \theta_{\text{rel}}) \left(R(\dot{\phi} + \dot{\theta}_{\text{rel}})^2 + g \right) = 0, \end{aligned} \quad (2.20)$$

$$\left(J_p - m_p R l \cos(\phi + \theta_{\text{rel}}) \right) \ddot{\phi} + J_p \ddot{\theta}_{\text{rel}} + m_p g l \sin(\phi + \theta_{\text{rel}}) = \tau_\theta, \quad (2.21)$$

with $J_c = J_o + (m + m_p)R^2$ where J_o is inertia moment of the hoop with respect to the center of the hoop, J_p is the inertia moment of the pendulum with respect to the center of the hoop, l is the half-length of the projection of the pendulum onto the plane P , and g stands for the gravitational acceleration.

To represent the hoop-pendulum dynamics in a more compact form, one can introduce the angle $\theta = \theta_{\text{rel}} + \phi$, defining the orientation of the pendulum with respect to the ground. Upon transformation of motion equations (2.20, 2.21) from θ_{rel} to θ , one obtains

$$\left(J_c - m_p R l \cos \theta \right) \ddot{\phi} + \left(J_p - m_p R l \cos \theta \right) \ddot{\theta} + m_p l \sin \theta \left(R \dot{\theta}^2 + g \right) = 0, \quad (2.22)$$

$$-m_p R l \cos \theta \ddot{\phi} + J_p \ddot{\theta} + m_p g l \sin \theta = \tau_\theta. \quad (2.23)$$

Three important remarks are in order.

1. Motion equations (2.22, 2.23) have already been derived in the literature (see [83, 84] and references therein) for the case of rolling along a straight line, which is a partial

case of pure rolling. Our derivation shows that the form (2.22, 2.23) holds true for any contact path compatible with the assumption of pure rolling.

2. Any motion of the sphere along the contact path is dynamically realizable by the pendulum (if the actuators are powerful enough), as for any $\phi(t)$ one can always find $\theta(t)$ by solving differential equation (2.22). Therefore, motion planning for the ball-pendulum system can be, in principle, decomposed to the kinematic (ball-plate system) and dynamic (hoop-pendulum system) levels.
3. Motion equations (2.22, 2.23) do not feature the angle γ defining the orientation of vector $\mathbf{n}$. Thus, the form (2.22, 2.23) is invariant to the shape of the contact curve. As a result, motion planning for the hoop-pendulum system covers a large variety of contact paths corresponding to pure rolling in a unified way.

2.3.3 Motion planning for the reduced system

The hoop-pendulum is an underactuated system where (2.22) forms a second order non-holonomic constraint. In planning the rest-to-rest movements for such a system, it is always an option to make use of a generic optimal control formulation with different performance indices. However, it comes with a higher computational cost. Instead, we employ the geometric phase approach to motion planning of underactuated systems [1] in which generation of closed-loop trajectories for base variables results in the propulsion of the fiber variables. Note from the nonholonomic constraint (2.22) that $\ddot{\phi}$ can be expressed by only θ and its derivatives. Therefore, the hoop angle ϕ and the pendulum angle θ are treated respectively as the fiber and base variables.

By solving (2.22) for $\ddot{\phi}$ and plugging it in (2.23), one establishes the static feedback control law

$$\begin{aligned} \tau_\theta = & \left(J_p + \frac{m_p R l \cos \theta (J_p - m_p R l \cos \theta)}{J_c - m_p R l \cos \theta} \right) u \\ & + m_p l \sin \theta \left(g + \frac{m_p R l \cos \theta (g + R \dot{\theta}^2)}{J_c - m_p R l \cos \theta} \right), \end{aligned} \quad (2.24)$$

transforming the hoop-pendulum dynamic equations (2.22, 2.23) to the following representation

$$\ddot{\phi} = -\frac{m_p l \sin \theta (g + R\dot{\theta}^2) + (J_p - m_p R l \cos \theta) u}{J_c - m_p R l \cos \theta}, \quad (2.25)$$

$$\ddot{\theta} = u, \quad (2.26)$$

where u is treated as a new control input. In the state-space form

$$\dot{\mathbf{x}}_{\text{hp}} = \begin{bmatrix} \dot{\phi} \\ -\frac{m_p l \sin \theta (g + R\dot{\theta}^2)}{J_c - m_p R l \cos \theta} \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{J_p - m_p R l \cos \theta}{J_c - m_p R l \cos \theta} \\ 0 \\ 1 \end{bmatrix} u, \quad (2.27)$$

where $\mathbf{x}_{\text{hp}} = (\phi, \dot{\phi}, \theta, \dot{\theta})$. Note that the collocated feedback linearization (2.24) is defined globally since $J_c > m_p R l$.

Consider first small motion of the hoop-pendulum system around an equilibrium configuration $\mathbf{x}_{\text{hp}}(0)$ or $\mathbf{x}_{\text{hp}}(T)$ where

$$\mathbf{x}_{\text{hp}}(0) = \mathbf{0}, \quad \mathbf{x}_{\text{hp}}(T) = (\phi_{\text{des}}, 0, 0, 0). \quad (2.28)$$

The linearization of (2.27) at $\mathbf{x}_{\text{hp}}(0)$ or $\mathbf{x}_{\text{hp}}(T)$ is defined as

$$\dot{\mathbf{x}}_{\text{hp}} = \mathbf{A} \mathbf{x}_{\text{hp}} + \mathbf{b} u, \quad (2.29)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ b \\ 0 \\ 1 \end{bmatrix},$$

with

$$a = -\frac{m_p g l}{J_c - m_p R l}, \quad b = -\frac{J_p - m_p R l}{J_c - m_p R l}.$$

Note that the linearized system (2.29) is controllable, which implies that the original non-linear system (2.27) is locally controllable around the equilibrium.

Under the dynamic constraints (2.29) and the boundary conditions (2.28), the optimal solution, minimizing the control effort

$$J = \frac{1}{2} \int_0^T u^2(t) \, dt, \quad (2.30)$$

is defined as [85]

$$\mathbf{x}_{\text{hp}}(t) = e^{\mathbf{A}t} \mathbf{W}(t) \mathbf{W}(T)^{-1} e^{-\mathbf{A}T} \mathbf{x}_{\text{hp}}(T), \quad (2.31)$$

where

$$\mathbf{W}(t) = \int_0^t e^{-\mathbf{A}s} \mathbf{b} \mathbf{b}^\top e^{-\mathbf{A}^\top s} \, ds. \quad (2.32)$$

It can be shown that the optimal trajectories of the hoop and the pendulum, calculated by (2.31), are expressed through the symmetric regularized incomplete Beta-function of 4-th order and its second derivative:

$$\phi(t) = B(4, 4, t/T) \phi_{\text{des}} + \frac{b}{a} \ddot{B}(4, 4, t/T) \phi_{\text{des}}, \quad (2.33)$$

$$\theta(t) = \frac{1}{a} \ddot{B}(4, 4, t/T) \phi_{\text{des}}, \quad (2.34)$$

where

$$B(4, 4, t/T) = \frac{t^4(20t^3 - 70t^2T + 84tT^2 - 35T^3)}{T^7}, \quad (2.35)$$

$$\ddot{B}(4, 4, t/T) = \frac{420t^2(t - T)^2(T - 2t)}{T^7}. \quad (2.36)$$

A profile of the shape variable $\theta(t)$ is shown by Fig. 2.6. It is an odd function with respect to the middle point $t = T/2$, with the extremal points $\theta_{\min}$ and $\theta_{\max} = -\theta_{\min}$

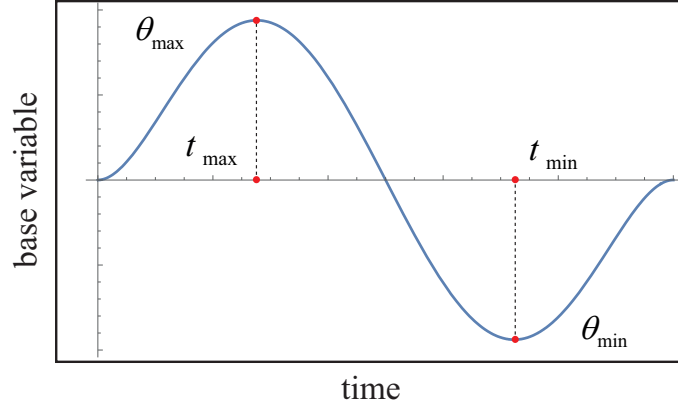


Figure 2.6: Motion pattern for the base variable $\theta(t)$

located at $t = T(5 \pm \sqrt{5})/10$. It is easy to find that

$$\phi_{\text{des}} = \frac{5\sqrt{5}aT^2}{84}\theta_{\text{m}}, \quad (2.37)$$

where θ_{m} is either maximal or minimal value of θ depending on the sign of ϕ_{des} ,

$$\theta(t) = \frac{25\sqrt{5}t^2(t-T)^2(2t-T)}{T^5}\theta_{\text{m}}. \quad (2.38)$$

One can show that the motion pattern (2.38) is a feasible candidate for generating rest-to-rest movements for the (non-linearized) hoop-pendulum system. Specifically, given the nonlinear differential equation (2.25) with the initial conditions $\phi(0) = \dot{\phi}(0) = 0$ and $\theta(t)$ defined by (2.38), one always has $\dot{\phi}(T) = 0$. Recall that $\theta(t)$ is the 2-nd derivative of Beta-function which has the property $B(t) = B(T) - B(T-t)$. Therefore, all its even derivatives are odd (asymmetric) and odd derivatives are even (symmetric) with respect to the middle point $t = T/2$. As a result, the right hand part of (2.25) is odd with respect to $t = T/2$ and so is $\ddot{\phi}(t)$. Therefore, $\int_0^T \ddot{\phi}(t) dt = \dot{\phi}(T) = 0$.

To solve the motion planning problem, one needs to find the value of θ_{m} such that $\phi(T) = \phi_{\text{des}}$. Note that (2.37) is no longer applicable as the dependence between these two parameters is not linear any more. However, one can find θ_{m} from solving a boundary value problem. Specifically, one can treat θ_{m} as an independent variable, supplement

(2.25) with one more differential equation $\dot{\theta}_m(t) = 0$ [86], and solve the resulting system of the 3-rd order with two initial conditions $\phi(0) = \dot{\phi}(0) = 0$ and one boundary condition $\phi(T) = \phi_{\text{des}}$, with the initial guess taken from the linearized model (2.33, 2.37).

It should be noted that the geometric phase technique comes with less computational cost compared with solving the minimum effort optimal control problem for the nonlinear system (2.27). Indeed, both the techniques result in solving a boundary value problem. However, in the former one there are three differential equations of the first order while in the latter one there are eight (four states and four co-states).

In the simulations, we compare the performance of the hoop-pendulum system under the proposed planner with that under the minimum effort optimal control formulation. The optimal criterion is selected as (2.30) where from (2.26), u is the angular acceleration $\ddot{\theta}$ of the pendulum. A rest-to-rest motion planning problem is studied, which is stated as finding a feasible trajectory for θ such that the corresponding ϕ satisfies the following boundary conditions

$$\phi(0) = 0, \dot{\phi}(0) = 0, \phi(T) = \phi_{\text{des}}, \dot{\phi}(T) = 0. \quad (2.39)$$

We conduct simulations with respect to various desired shift ϕ_{des} of the fiber variable ϕ . The value of ϕ_{des} changes from 0 to 20π . The numerical values used in the simulation are $m = 1[\text{kg}]$, $m_p = 1[\text{kg}]$, $R = 0.2[\text{m}]$, $l = 0.1[\text{m}]$, $g = 9.8[\text{m/s}^2]$ and $T = 10[\text{s}]$.

We plot the configuration variables ϕ and θ as functions of both ϕ_{des} and time t , as shown in Fig. 2.7 and Fig. 2.8. Compared with the minimum effort optimal control formulation, the proposed geometric phase based approach results to almost the same hoop motions, whereas the pendulum motions differ by less than $0.2[\text{rad}]$ in maximum. For both motion planners, the comparison between the values of the cost function J for the hoop-pendulum system is illustrated by Table 2.1. Note that the value of J for the geometric phase based approach is only 0.02% higher than that of the optimal control in maximum, which implies that both motion planners provide similar system performance.

The relation between the parameter θ_m , representing the amplitude of the pendulum angle θ , and the value of ϕ_{des} is illustrated in Fig. 2.9. When θ_m is relatively small (yellow part of the trajectory), the increase of which is proportional to that of ϕ_{des} , due to that in this phase the performance of the hoop-pendulum is similar to its linearized system. How-

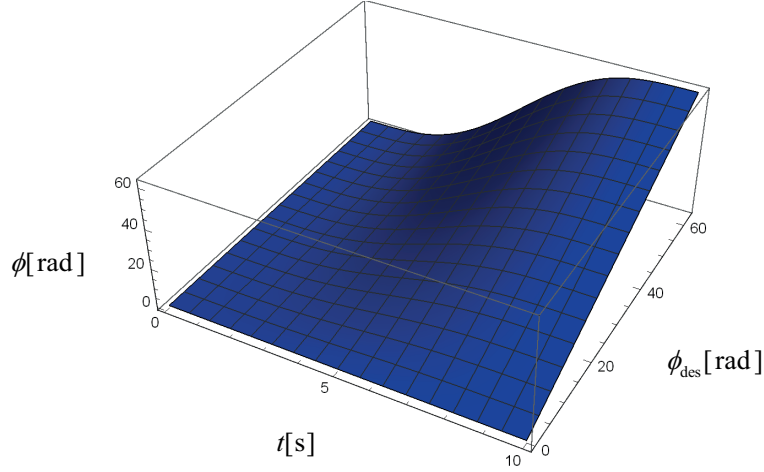


Figure 2.7: Hoop motions resulted from geometric phase based control.

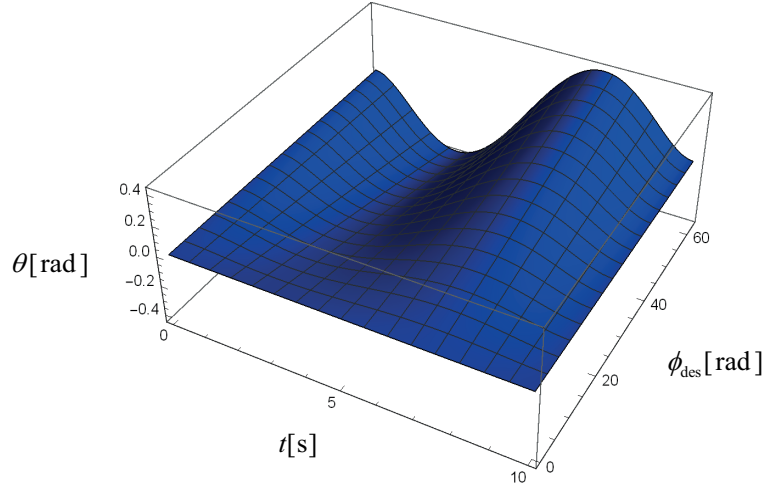


Figure 2.8: Pendulum motions resulted from geometric phase based control.

ever, when θ_m is getting larger (red part of the trajectory), this relation is no longer linearly proportional.

It should be noted that the proposed motion technique has one shortcoming appearing when the hoop is required to reach a large angular displacement ϕ_{des} within a relatively short time period T . In such situations, as shown in Fig. 2.9, the solution of the boundary value problem, implemented in the planner, may not be unique. This itself does not represent a problem—setting the initial approximation for θ_m equal to the one resulted from the lin-

ϕ_{des}	2π	4π	6π	8π	10π
Optimal control	0.0015571	0.0062403	0.0140854	0.0251534	0.0395317
Geometric phase	0.0015571	0.0062403	0.0140854	0.0251536	0.0395322
ϕ_{des}	12π	14π	16π	18π	20π
Optimal control	0.0573363	0.0787156	0.103854	0.132979	0.166365
Geometric phase	0.0573377	0.0787194	0.103863	0.132998	0.166403

Table 2.1: Comparison for the values of the cost function J , expressed by (2.30), between the minimum effort optimal control formulation and the geometric phase based approach for the hoop-pendulum system.

earized optimal control problem will single out the left side of the $\theta_m - \phi_{\text{des}}$ curve. Of more concern is the fact that for relatively large ϕ_{des} there may exist no feasible θ_m that steers ϕ to ϕ_{des} as the $\theta_m - \phi_{\text{des}}$ curve may have a point of extremum.

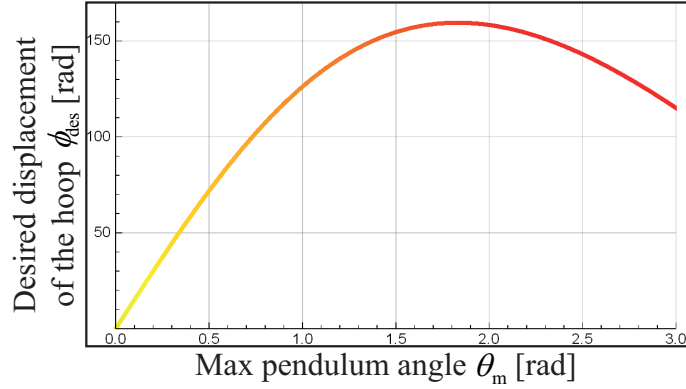


Figure 2.9: Relation between θ_m and ϕ_{des} for the hoop-pendulum system under the geometric phase based motion planning algorithm. The yellow and red portions of the curve illustrate this relation respectively when ϕ_{des} is rather small and large.

One possible (and perhaps simplest) solution to cope with this problem is to elongate the time interval T such that a large ϕ_{des} can be reached under the geometric phase based motion planning. Another solution is to assign the pendulum with a different motion pattern. Note from Fig. 2.6, the timing law of the pendulum angle $\theta(t)$ is a rest-to-rest function with one oscillation (one crest and one trough). An alternative motion pattern could be adopted with multiple oscillations—one can increase the number of oscillations having lower magnitude.

2.3.4 Calculation of the actual input torque

The input torque τ defined in (2.14) is expressed by $\phi(t)$, $\theta(t)$ as well as the general coordinates of the pendulum $q_1(t)$ and $q_2(t)$, and the normal vector $\mathbf{n}(t)$. As the expressions for $\phi(t)$ and $\theta(t)$ are already obtained from the previous section, the calculations for $q_1(t)$, $q_2(t)$ and $\mathbf{n}(t)$ are explained as follows.

Calculation of the generalized pendulum coordinates

The generalized coordinates of the pendulum, q_1 and q_2 , are calculated by applying the fact that pendulum vector is equivalent in various parametrizations. Define in the base coordinate frame Σ_b the pendulum vector $\boldsymbol{\rho}_p$, emanated from the geometric center of the spherical shell o , to the mass of center of the pendulum p , as

$${}^b\boldsymbol{\rho}_p = \begin{bmatrix} -l_p \cos \gamma \sin \theta \\ l_p \sin \gamma \sin \theta \\ -l_p \cos \theta \end{bmatrix}, \quad (2.40)$$

where angles θ and γ are illustrated in Fig 2.4. Then, the pendulum vector $\boldsymbol{\rho}_p$, defined in Σ_o (coordinate frame fixed on point o), is calculated as

$${}^o\boldsymbol{\rho}_p = {}^o\mathbf{R}_b {}^b\boldsymbol{\rho}_p = \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix}, \quad (2.41)$$

where ${}^o\mathbf{R}_b(u_o, v_o, \psi)$ is the transformation matrix between Σ_o and Σ_b and

$$\rho_x^2 + \rho_y^2 + \rho_z^2 = l_p^2, \quad (2.42)$$

because the length of the pendulum does not change.

Note that the pendulum vector ${}^o\boldsymbol{\rho}_p$ can also be parameterized by the pendulum angles q_1

and q_2 as ${}^o\boldsymbol{\rho}_p(q_1, q_2)$ defined in Σ_o , as

$${}^o\boldsymbol{\rho}_p = {}^o\mathbf{R}_p {}^p\boldsymbol{\rho}_p = \begin{bmatrix} -l_p \sin q_1 \cos q_2 \\ l_p \sin q_2 \\ -l_p \cos q_1 \cos q_2 \end{bmatrix}, \quad (2.43)$$

where ${}^p\boldsymbol{\rho}_p = (0, 0, -l_p)$ is the pendulum vector defined in the coordinate frame Σ_p fixed on the pendulum. Note that

$$(l_p \sin q_1 \cos q_2)^2 + (l_p \sin q_2)^2 + (l_p \cos q_1 \cos q_2)^2 = l_p^2, \quad (2.44)$$

because the length of the pendulum does not change. Assuming ρ_x , ρ_y , and ρ_z are given as time functions, we are to calculate the pendulum angles $q_1(t)$ and $q_2(t)$. By substituting (2.41) into (2.43), one obtains

$$\rho_x(t) = -l_p \sin q_1 \cos q_2, \quad (2.45)$$

$$\rho_y(t) = l_p \sin q_2, \quad (2.46)$$

$$\rho_z(t) = -l_p \cos q_1 \cos q_2, \quad (2.47)$$

where $\rho_x(t)$, $\rho_y(t)$, and $\rho_z(t)$ are the given time functions while q_1 and q_2 are the pendulum angles to be calculated.

Thus, one has three equations for two variables. However, note from (2.42) and (2.44), that the above equations (2.45), (2.46), and (2.47) are not independent. Specifically, if there exist $q_1(t)$ and $q_2(t)$ that satisfy (2.45) and (2.46), one finds that (2.47) also holds true. Therefore, for the calculation of $q_1(t)$ and $q_2(t)$, we can remove one of the redundant equations, i.e., (2.47), and then, solve the rest two equations with respect to q_1 and q_2 . By solving (2.45) and (2.46), one obtains

$$q_1 = -\sin^{-1} \frac{l_p \rho_x(t)}{\sqrt{\rho_x(t)^2 + \rho_z(t)^2}}, \quad q_2 = \sin^{-1} \frac{\rho_y(t)}{l_p}. \quad (2.48)$$

Note that there exists a singular point when $\rho_x^2 + \rho_z^2 = 0$. Physically, the singular point appears when the pendulum is parallel to the ground, that is $\theta = \pi/2$. It can be avoided by keeping $\theta < \pi/2$ in the motion planning.

Calculation of the normal vector

To complete the solution of the motion planning problem, in what follows the expression of $\mathbf{n}(t)$, as a time function, is determined. A curve on the contact plane can be defined as $\mathbf{c}(\phi) = (u_b(\phi), v_b(\phi), 0)$ where ϕ describes the rotation of the sphere projected on the plane P . Thus the normal vector $\mathbf{n}$ can be determined as

$$\mathbf{n} = \frac{1}{\sqrt{\dot{u}_b^2 + \dot{v}_b^2}}(\dot{v}_b, -\dot{u}_b, 0). \quad (2.49)$$

As from the proposed motion planner for the hoop-pendulum system, ϕ is derived as a function of t , and thus the normal vector $\mathbf{n}$ is also determined as a time function $\mathbf{n}(t)$.

The motion planning algorithm for tracing curves on the contact plane was completed by determining $\mathbf{n}(t)$ as a time function. However, in the motion planning for a spherical robot, especially when the control of the orientation of the spherical is considered, it is required for the robot to trace given curves on the spherical surface [87]. The motion planner allowing the robot to trace spherical contact curves is introduced as follows.

Let the position of the contact point on the sphere be parameterized by a spherical curve $\mathbf{c}(\xi) = (x(\xi), y(\xi), z(\xi))$ where ξ is a curve parameter. Since the same point is defined by (2.1), one has $\mathbf{c}(\xi) = {}^o\boldsymbol{\rho}_c(u_o, v_o)$. For pure rolling motion the contact kinematics are established as [87]

$$\dot{u}_b = \left(-\frac{\cos \psi}{\sqrt{R^2 - y^2}} \mathbf{c}_u^T \mathbf{c}_\xi + \frac{\sin \psi}{R} \mathbf{c}_v^T \mathbf{c}_\xi \right) \dot{\xi}, \quad (2.50)$$

$$\dot{v}_b = \left(-\frac{\cos \psi}{\sqrt{R^2 - y^2}} \mathbf{c}_u^T \mathbf{c}_\xi + \frac{\sin \psi}{R} \mathbf{c}_v^T \mathbf{c}_\xi \right) \dot{\xi}, \quad (2.51)$$

$$\dot{\psi} = \left(\frac{y}{R(R^2 - y^2)} \mathbf{c}_u^T \mathbf{c}_\xi \right) \dot{\xi}, \quad (2.52)$$

where $\mathbf{c}_\xi = \mathbf{c}'(\xi)$ and the partial derivatives $\mathbf{c}_u = \partial \mathbf{c} / \partial u_o$ and $\mathbf{c}_v = \partial \mathbf{c} / \partial v_o$ are defined as follows

$$\mathbf{c}_u = \begin{bmatrix} z \\ 0 \\ -x \end{bmatrix}, \quad \mathbf{c}_v = \begin{bmatrix} -xy / \sqrt{R^2 - y^2} \\ \sqrt{R^2 - y^2} \\ -yz / \sqrt{R^2 - y^2} \end{bmatrix}. \quad (2.53)$$

Note that the speed of contact point on the spherical surface, represented by $\|\mathbf{c}'(\xi)\|\dot{\xi}$, is equal to the speed of the center of the sphere, represented by $R\dot{\phi}$, which gives

$$\dot{\phi} = \frac{\|\mathbf{c}'(\xi)\|}{R} \dot{\xi}. \quad (2.54)$$

As from the proposed motion planner, the hoop angle ϕ is derived as a function of t , based on (2.54), the curve parameter ξ is determined as a time function $\xi(t)$ and thus so are $\dot{u}_b$ and $\dot{v}_b$. Then the normal vector $\mathbf{n}$ calculated from (2.49), is also a time function $\mathbf{n}(t)$.

2.3.5 Case study

The proposed motion planning algorithm is tested by simulations for tracing contact curves on the plane and spherical surface, respectively. Specifically four cases are studied, tracing straight line, circular curve on the contact plane and tracing Viviani's curve, the Loxodrome on the spherical surface. The numerical values used in the simulations are the same as in 2D case.

Straight line

Here we consider the problem of tracing a straight line on the contact plane. A straight line on the plane is defined by

$$\mathbf{c}(\phi) = \begin{bmatrix} \phi + x_o \\ k\phi + y_o \\ 0 \end{bmatrix}, \quad (2.55)$$

where k , x_o and y_o are constant parameters. In this simulation, the desired path to be traced on the plane is specified as $k = \sqrt{3}$ and $x_o = y_o = 0$. The simulation results are shown from Fig. 2.10 to Fig. 2.13. It can be seen from the simulation results that the spherical robot traces the straight line on the contact plane in rest-to-rest mode.

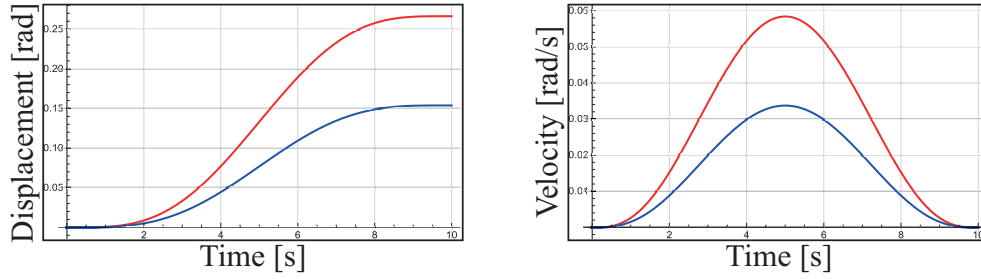


Figure 2.10: Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).

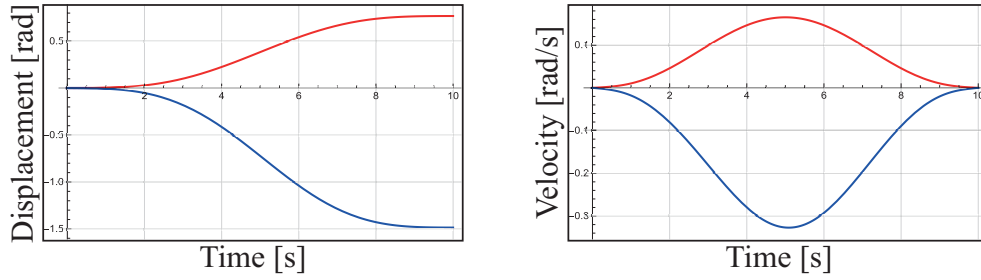


Figure 2.11: Generalized coordinates of the pendulum q_1 , $\dot{q}_1$ (red curve) and q_2 , $\dot{q}_2$ (blue curve).

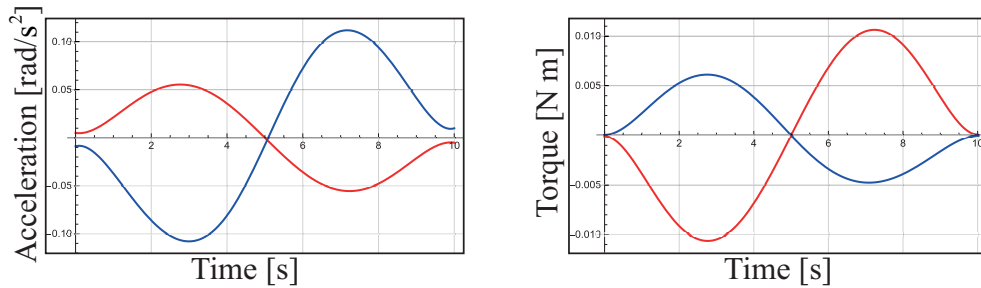


Figure 2.12: Generalized accelerations and torques on the pendulum: $\ddot{q}_1$, τ_1 (red curve) and $\ddot{q}_2$, τ_2 (blue curve).

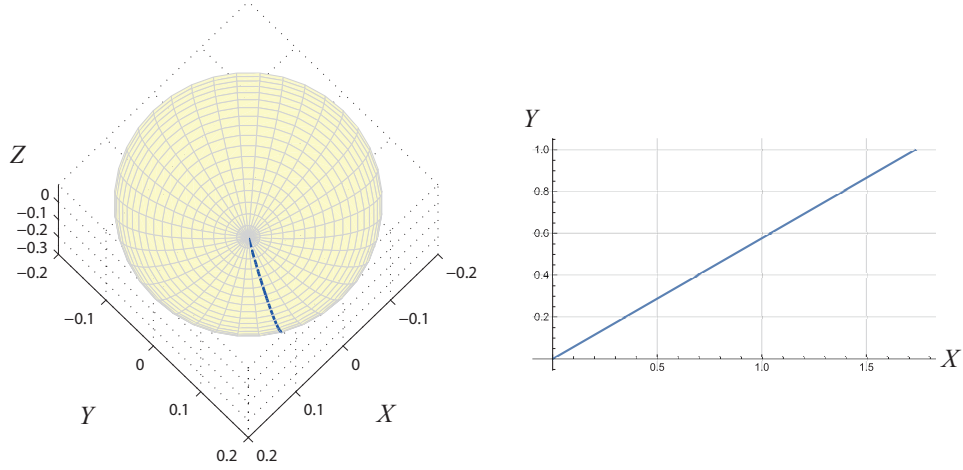


Figure 2.13: Trace of the contact point on the spherical surface (left) and on the plane (right).

Circular curve

Here we consider the problem of tracing a circular curve on the contact plane. A circular curve on the plane is defined by

$$\mathbf{c}(\phi) = \begin{bmatrix} L \cos \phi + x_o \\ L \sin \phi + y_o \\ 0 \end{bmatrix}, \quad (2.56)$$

where L , x_o and y_o are constant parameters. In this simulation, $x_o = 0$, $y_o = -0.5$ and $L = 0.5$.

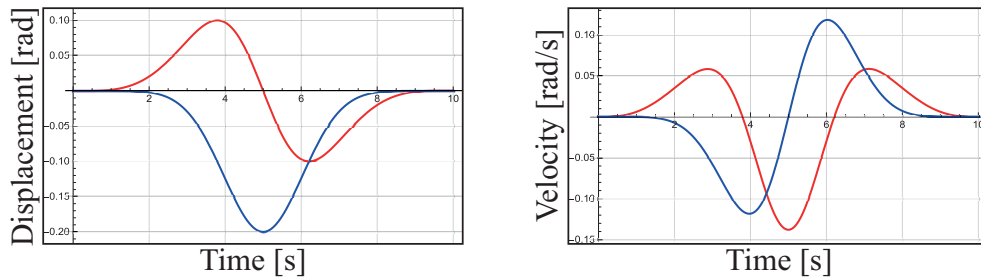


Figure 2.14: Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).

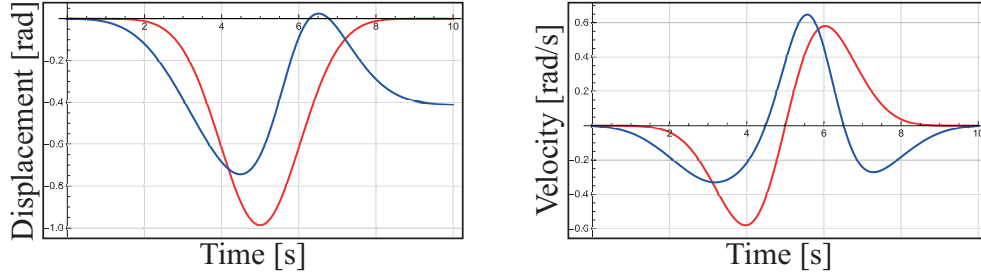


Figure 2.15: Generalized coordinates of the pendulum q_1 , $\dot{q}_1$ (red curve) and q_2 , $\dot{q}_2$ (blue curve).

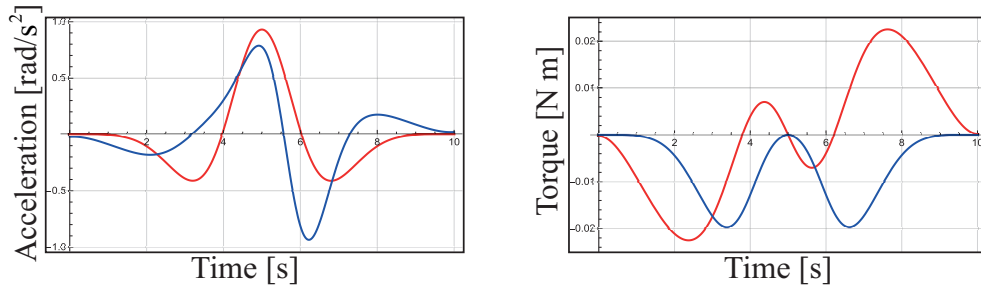


Figure 2.16: Generalized accelerations and torques on the pendulum: $\ddot{q}_1$, τ_1 (red curve) and $\ddot{q}_2$, τ_2 (blue curve).

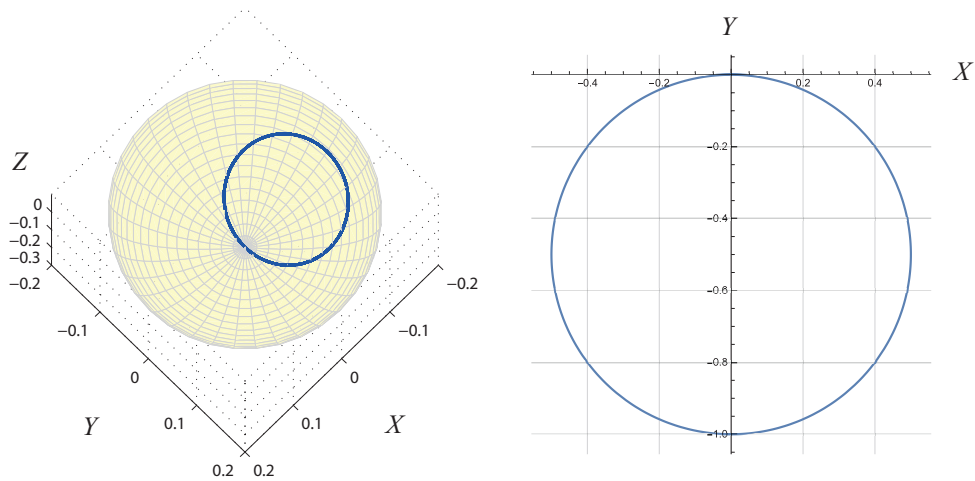


Figure 2.17: Trace of the contact point on the spherical surface (left) and on the plane (right).

The simulation results are shown from Fig. 2.14 to Fig. 2.17. It can be seen from the

simulation results that the spherical robot traces the circular curve on the contact plane in rest-to-rest mode.

Viviani's curve

Here we consider the problem of tracing Viviani's curve on the sphere. For Viviani's curve we have

$$\mathbf{c}(\xi) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2\sqrt{d(R-d)} \sin \xi \\ (d-R) \sin 2\xi \\ -d + (d-R) \cos 2\xi \end{bmatrix}, \quad (2.57)$$

where d is a constant parameter and

$$\dot{\phi} = \frac{2\sqrt{(R-d)[R+2(d-R)(\cos 2\xi - 1)]} \cos \xi}{R} \dot{\xi}. \quad (2.58)$$

Assume that we want to trace this curve three times so that $\xi(0) = 0$, and $\xi(T) = 6\pi$. Set $d = 0.1467768$ so that the geometric center of the spherical shell is displaced along the Y -axis of Σ_b to the distance 0.3.

The simulation results are shown from Fig. 2.18 to Fig. 2.21. It can be seen from the simulation results that the spherical robot traces the Viviani's curve on the spherical surface in rest-to-rest mode.

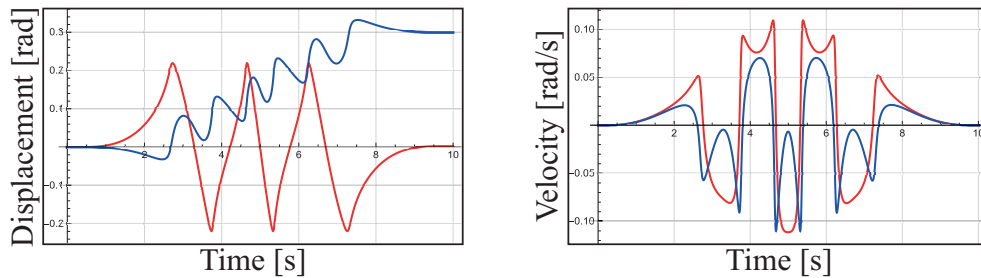


Figure 2.18: Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).

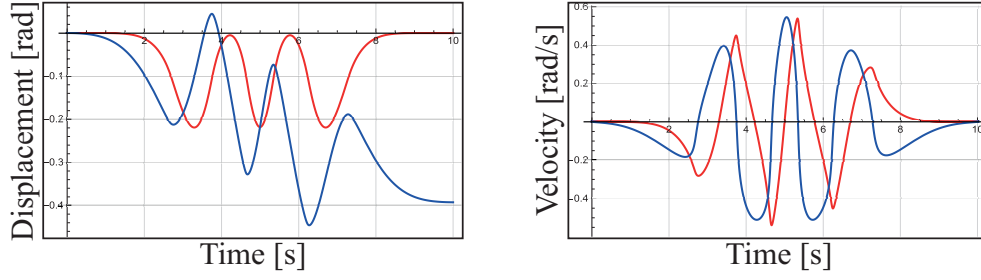


Figure 2.19: Generalized coordinates and angular velocities of the pendulum $q_1, \dot{q}_1$ (red curves) and $q_2, \dot{q}_2$ (blue curves).

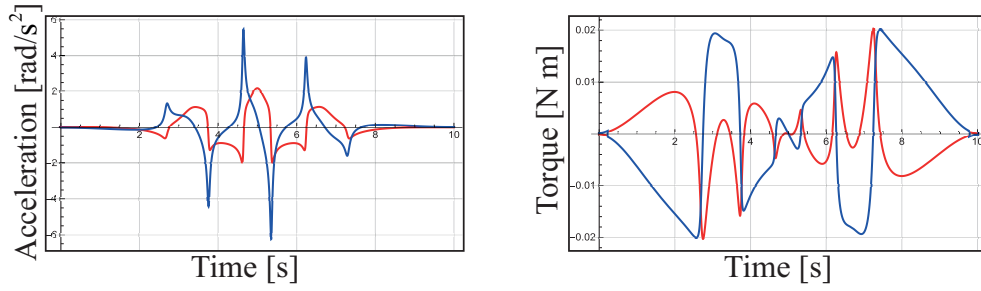


Figure 2.20: Generalized accelerations and torques on the pendulum: $\ddot{q}_1, \tau_1$ (red curve) and $\ddot{q}_2, \tau_2$ (blue curve).

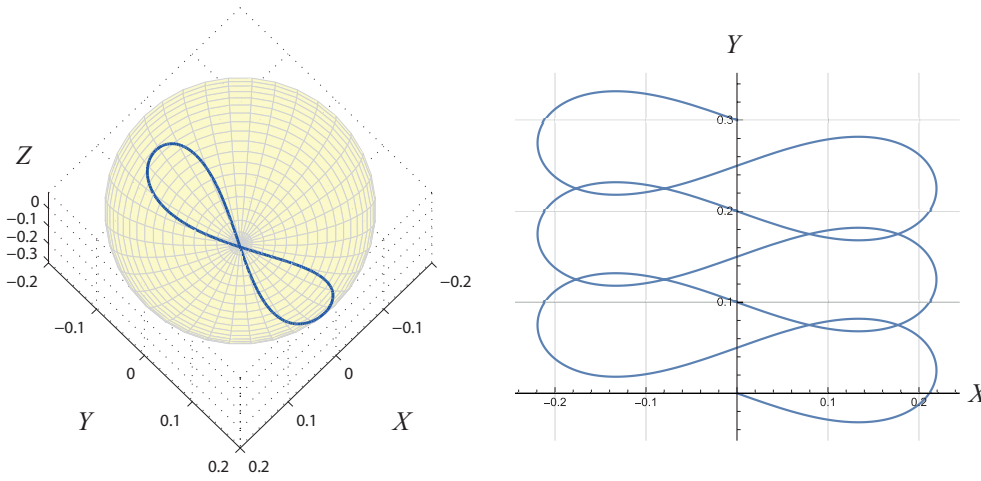


Figure 2.21: Trace of the contact point on the spherical surface (left) and on the plane (right).

Loxodrome

Here we consider the problem of tracing the Loxodrome on the sphere. For the Loxodrome we have

$$\mathbf{c}(\xi) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R \cos \xi / \cosh(\cot \alpha (\xi - \xi_o)) \\ R \sin \xi / \cosh(\cot \alpha (\xi - \xi_o)) \\ R \tanh(\cot \alpha (\xi - \xi_o)) \end{bmatrix}, \quad (2.59)$$

where α and ξ_o are constant parameters and

$$\dot{\phi} = \csc \alpha \operatorname{sech}(\xi \cot \alpha) \dot{\xi}. \quad (2.60)$$

Let $\xi_o = 0$ and $\alpha = 15\pi/32$. The simulation results are shown from Fig. 2.22 to Fig. 2.25.

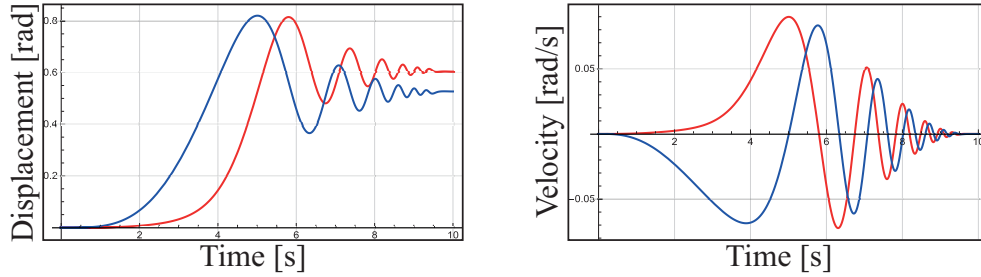


Figure 2.22: Displacement and velocity of the shell along X -axis (red curve) and along Y -axis (blue curve).

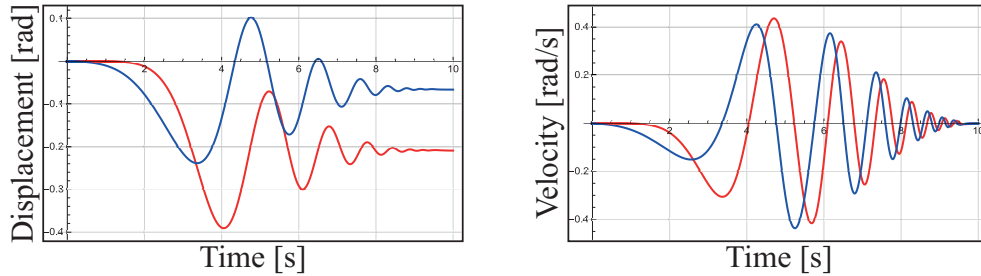


Figure 2.23: Generalized coordinates and angular velocities of the pendulum $q_1, \dot{q}_1$ (red curves) and $q_2, \dot{q}_2$ (blue curves).

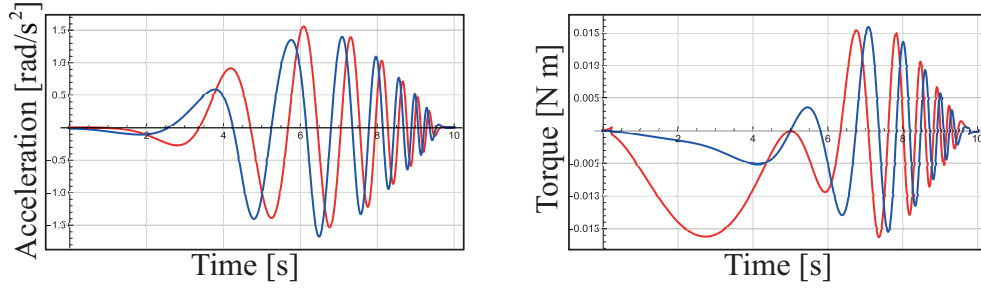


Figure 2.24: Generalized accelerations and torques on the pendulum: $\ddot{q}_1, \tau_1$ (red curve) and $\ddot{q}_2, \tau_2$ (blue curve).

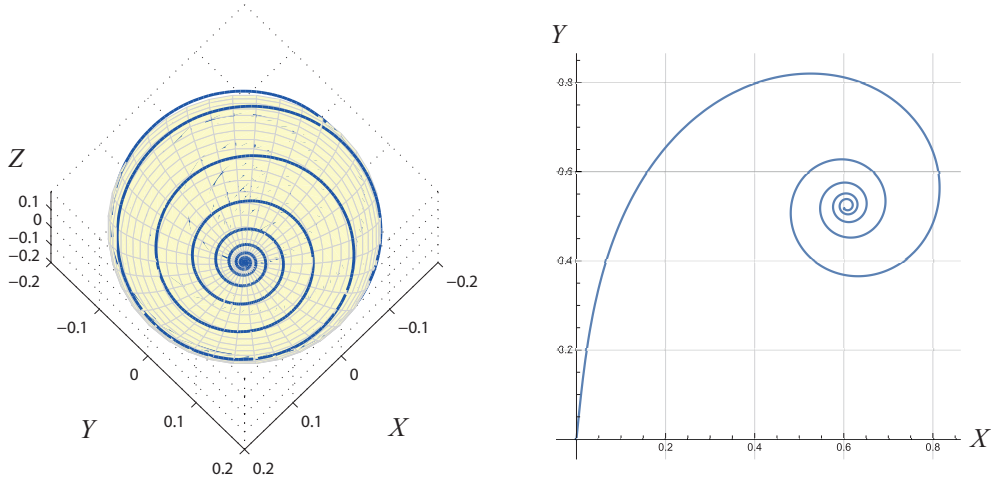


Figure 2.25: Trace of the contact point on the spherical surface (left) and on the plane (right).

It can be seen from the simulation results that the spherical robot traces the Loxodrome on the spherical surface in rest-to-rest mode.

2.4 Trajectory tracking

Research on spherical rolling robots has been drawing increasing attention in recent years. Depending on the driving principles, the spherical robots can be classified into three types depending on the driving principles: changing the mass center [10, 64–66], changing the angular momentum [13, 14, 63, 68, 88] and changing the outer shape [15]. A review of the design based on these driving principles can be found in [70]. In this section we explore a trajectory tracking problem for a class of spherical rolling robots actuated by a pendulum with two degrees of mobility since two is the minimal number of the control inputs necessary to move the spherical shell.

The control problem can be stated as follows. Given a reference motion trajectory of the spherical robot, construct a tracking control law such that the error between the ideal and actual configurations of the spherical rolling robot converges to zero. This problem is difficult because the ball-pendulum system is nonholonomic, underactuated, and has non-zero drift term. Due to these difficulties, the approaches reported in the robotics literature [82, 89–91] only solved the tracking problem partially. They either track the motion trajectory for only the orientation or only the position. In our previous work [92], we proposed a backstepping framework for the control process of tracking the motion trajectory for the full configuration of the sphere, including both its position and orientation. However, the constructed input torque contains acceleration term which is difficult to measure in the application. We improve it by restructuring the backstepping process such that acceleration term will be excluded from the input law. Also it should be noted that in [92], we assumed that the mathematical model perfectly described the relation between the input and the output. However there are always errors between the mathematical model and the actual mechanical system due to inaccurately measured system parameters, un-modeled dynamics or unexpected external perturbation. These errors refer to the general uncertainties [93], which may cause deviations of the system trajectory from the desired one when the control inputs are constructed from the ideal model. To correct these deviations, an adaptive trajectory tracking controller is necessary. The development of such a controller constitutes the main goal of Section 2.4.

The adaptation problem is commonly addressed by the well-known model reference

adaptive control schemes (MRAC). The controller is formed by combining on-line parameter estimators, which provide estimates of unknown parameters at each instant, with a control law motivated from the known parameter case. However, in practical applications, there always exist non-parametric, or time-varying uncertainties such as un-modeled dynamics or external disturbance. The traditional MRAC method is not feasible for dealing with these conditions. To control a rolling robot with time-varying uncertainties, we adopt a new adaptive controller design method based on function approximation technique (FAT) [53–56,94,95]. Compared with the traditional MRAC, the advantage of the FAT-based control is in the representation of the general uncertainties by a set of given basis functions weighted by a set of unknown constant parameters [93]. Thus the problem of eliminating the influence of general uncertainties is transformed to the estimation of parametric errors. Then Lyapunov designs are applied to derive proper update laws adjusting the estimates of the unknown parameters.

The adaptive trajectory tracking controller for the ball-pendulum system is constructed under backstepping framework and the adaptation mechanism is designed via the FAT. The backstepping process combines two steps. A feedback velocity control input, denoted by angular velocity of the spherical shell, is first designed for the kinematic steering system. Then the specific robot dynamics are taken into account to convert a steering command into control torque acting on the pendulum. Note that in either step the subsystem to be controlled is underactuated and the FAT cannot be directly applied. Therefore, the main contribution of this work is in the modification of the FAT-based adaptive control method such that it becomes applicable to the class of underactuated systems corresponding to rolling robots.

2.4.1 Adaptive trajectory tracking controller design

We construct the adaptive tracking controllers the ball-pendulum system with time varying uncertainties in a backstepping framework. To follow the backstepping technique, it is required to rewrite the motion equations in the pure feedback form [93] form

$$\dot{x} = f_x(x, z), \quad (2.61a)$$

$$\dot{z} = f_z(x, z, u). \quad (2.61b)$$

The trajectory tracking problem is posed as follows. Given a reference motion trajectory of the spherical shell $\mathbf{x}_r(t)$, construct an input torque $\boldsymbol{\tau}$ such that $\lim_{t \rightarrow \infty} (\mathbf{x}_r - \mathbf{x}) = \mathbf{0}$. Here in this equation, $\mathbf{x}_r(t) = (u_b^*, v_b^*, u_o^*, v_o^*, \psi^*)$ is the reference configuration of the spherical shell where (u_b^*, v_b^*) and (u_o^*, v_o^*, ψ^*) represent respectively the desired smooth trajectories for its position of the geometric center and for its orientation. We construct the adaptive tracking controller for the ball-pendulum system expressed by (2.2) and (2.14) with time-varying uncertainties, based on the backstepping techniques, mimicking the planar case. To apply backstepping techniques, it is required to rewrite the motion equations in form of (2.61a) and (2.61b). Apparently, the contact kinematics (2.2) of the ball-pendulum system satisfy the form of (2.61a). The following steps are made to convert the system dynamics (2.14) into the form of (2.61b). Rewrite the system dynamics (2.14) as

$$\dot{\boldsymbol{\omega}} = -\mathbf{H}_{oo}^{-1} \mathbf{H}_{op} \ddot{\mathbf{q}} - \mathbf{H}_{oo}^{-1} \mathbf{h}_o, \quad (2.62a)$$

$$\mathbf{H}_{pp} \ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{H}_{po} \dot{\boldsymbol{\omega}} - \mathbf{h}_p, \quad (2.62b)$$

and from (2.62a) and (2.62b), one obtains the second derivative for the pendulum coordinates

$$\ddot{\mathbf{q}} = (\mathbf{H}_{pp} - \mathbf{H}_{po} \mathbf{H}_{oo}^{-1} \mathbf{H}_{op})^{-1} (\boldsymbol{\tau} + \mathbf{H}_{po} \mathbf{H}_{oo}^{-1} \mathbf{h}_o - \mathbf{h}_p).$$

Substituting the expression of $\ddot{\mathbf{q}}$ into (2.62a) and together with the kinematics, one obtains the mathematical model for the ball-pendulum system in the backstepping form (2.61a) and (2.61b)

$$\dot{\mathbf{x}} = \mathbf{A} \boldsymbol{\omega}, \quad (2.63a)$$

$$\begin{aligned} \dot{\boldsymbol{\omega}} = & -\mathbf{H}_{oo}^{-1} \mathbf{H}_{op} (\mathbf{H}_{pp} - \mathbf{H}_{po} \mathbf{H}_{oo}^{-1} \mathbf{H}_{op})^{-1} \boldsymbol{\tau} \\ & - \mathbf{H}_{oo}^{-1} (\mathbf{H}_{op} (\mathbf{H}_{po} \mathbf{H}_{oo}^{-1} \mathbf{h}_o - \mathbf{h}_p) + \mathbf{h}_o). \end{aligned} \quad (2.63b)$$

The input torque $\boldsymbol{\tau}$ is constructed in two steps, following the backstepping technique. First, find a control law for the velocity input $\boldsymbol{\omega}$ to the kinematic steering system (2.63a) such that $\lim_{t \rightarrow \infty} (\mathbf{x}_r - \mathbf{x}) = \mathbf{0}$. The control law is specified as $\boldsymbol{\omega}_d$. Then, taking into account the robot dynamics (2.63b), construct an input torque $\boldsymbol{\tau}$ such that $\lim_{t \rightarrow \infty} (\boldsymbol{\omega}_d - \boldsymbol{\omega}) = \mathbf{0}$. Let us discuss

the construction in detail.

Step 1 In this step we examine the kinematics (2.63a) only. Define error between real and reference configuration of the sphere, including both its position and orientation, as $\mathbf{e} = \mathbf{x}_r - \mathbf{x} = (e_1, e_2, e_3, e_4, e_5)$. Assuming the angular velocity of the spherical shell can be perfectly realized, that is $\boldsymbol{\omega} = \boldsymbol{\omega}_d$, the error dynamics can be described by

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_r - \dot{\mathbf{x}} = \dot{\mathbf{x}}_r - \mathbf{A}\boldsymbol{\omega}_d. \quad (2.64)$$

We are to find a feasible input $\boldsymbol{\omega}_d$ such that the output $\mathbf{e}$ converges to $\mathbf{0}$.

Note that the major difference between the velocity controller design process for the ball-pendulum and for the hoop-pendulum is (2.64) is an underactuated system such that $\mathbf{A}^{-1}$ does not exist and the FAT-based control algorithm is not directly applicable. To deal with this problem, by introducing an auxiliary input $\boldsymbol{\omega}^* \in \mathbb{R}^5$ such that

$$\boldsymbol{\omega}_d = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \boldsymbol{\omega}^* = \mathbf{A}^+ \boldsymbol{\omega}^*, \quad (2.65)$$

we reformulate (2.64) as

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_r - \mathbf{A}\mathbf{A}^+ \boldsymbol{\omega}^* + \boldsymbol{\omega}^* - \boldsymbol{\omega}^* = \dot{\mathbf{x}}_r - \boldsymbol{\omega}^* + \mathbf{a}, \quad (2.66)$$

where $\mathbf{a} = (\mathbf{I} - \mathbf{A}\mathbf{A}^+) \boldsymbol{\omega}^*$. Note that $\mathbf{a}$ can be viewed as the time-varying uncertainty of system (2.66). Then the control problem is restated as constructing $\boldsymbol{\omega}^*$ such that $\lim_{t \rightarrow \infty} \mathbf{e} = \mathbf{0}$, with the variation $\mathbf{a}$ unknown. Here we use a polynomial function weighted by constant parameters $\mathbf{a}_i$ to approximate the unknown function $\mathbf{a}$ as

$$\mathbf{a} \approx \sum_{i=0}^N \mathbf{a}_i t^i. \quad (2.67)$$

To eliminate the influence of $\mathbf{a}$ to the control process, the unknown parameters $\mathbf{a}_i$, referring to the plant parameters, are required to be identified. These plant parameters $\mathbf{a}_i$ are estimated at each time t denoted by the adjustable control parameters $\hat{\mathbf{a}}_i(t)$ using an update law that

we are to define from the following process.

To construct a feedback velocity control ω^* that steers e to zero and an update law that defines $\hat{\mathbf{a}}_i(t)$, a feasible Lyapunov candidate function would be

$$V_k = \frac{1}{2} \mathbf{e}^\top \mathbf{e} + \frac{1}{2} \sum_{i=0}^N \left(\hat{\mathbf{a}}_i(t) - \mathbf{a}_i \right)^\top \left(\hat{\mathbf{a}}_i(t) - \mathbf{a}_i \right), \quad (2.68)$$

which incorporates both the state error $\mathbf{e}$ and the estimation error $\hat{\mathbf{a}}_i(t) - \mathbf{a}_i$ between the plant parameters $\mathbf{a}_i$ and their estimates $\hat{\mathbf{a}}_i(t)$. Note that in (2.68), the parameters $\mathbf{a}_i$ are constants, which implies $\dot{\mathbf{a}}_i = \mathbf{0}$. Therefore the derivative of the Lyapunov function candidate is

$$\dot{V}_k = \mathbf{e}^\top \left(\dot{\mathbf{x}}_r - \dot{\omega}^* \right) + \sum_{i=0}^N \left[\hat{\mathbf{a}}_i(t)^\top \dot{\hat{\mathbf{a}}}_i(t) + \mathbf{a}_i^\top \left(\mathbf{e} t^i - \dot{\hat{\mathbf{a}}}_i(t) \right) \right]. \quad (2.69)$$

As the constructed control law cannot contain unmeasurable elements, the unknown parameters $\mathbf{a}_i$ in the derivative of Lyapunov function need to be excluded. To cancel the terms with $\mathbf{a}_i$, define the update law

$$\dot{\hat{\mathbf{a}}}_i(t) = \mathbf{e} t^i, \quad (2.70)$$

which leads to

$$\dot{V}_k = \mathbf{e}^\top \left(\dot{\mathbf{x}}_r + \sum_{i=0}^N \hat{\mathbf{a}}_i(t) t^i - \dot{\omega}^* \right).$$

Constructing the auxiliary velocity input

$$\omega^* = \dot{\mathbf{x}}_r + \mathbf{K} \mathbf{e} + \sum_{i=0}^N \hat{\mathbf{a}}_i(t) t^i, \quad (2.71)$$

where $\mathbf{K}$ is a positive definite matrix, yields

$$\dot{V}_k = -\mathbf{e}^\top \mathbf{K} \mathbf{e} \leq 0,$$

which implies the convergence of $\mathbf{e}$. Hence the velocity input ω_d can be calculated from

(2.65) as

$$\boldsymbol{\omega}_d = \mathbf{A}^+ \left(\dot{\mathbf{x}}_r + \mathbf{K}\mathbf{e} + \sum_{i=0}^N \hat{\mathbf{a}}_i(t) t^i \right). \quad (2.72)$$

It should be noted that in Step 1, we assume that the angular velocity of the sphere can be perfectly realized, namely $\boldsymbol{\omega} = \boldsymbol{\omega}_d$. However, in the real implementation, as we cannot directly control the angular velocity of the spherical robot, there exists a deviation of $\boldsymbol{\omega}$ from the desired angular velocity input $\boldsymbol{\omega}_d$. In the next step we design input torque $\boldsymbol{\tau}$ such that the real $\boldsymbol{\omega}$ converges to $\boldsymbol{\omega}_d$.

Step 2 Construct torque $\boldsymbol{\tau}$ such that $\lim_{t \rightarrow \infty} (\boldsymbol{\omega}_d - \boldsymbol{\omega}) = \mathbf{0}$. Define the deviation of actual angular velocity of the sphere from the desired as $\mathbf{e}_\omega = \boldsymbol{\omega}_d - \boldsymbol{\omega}$, such that $\dot{\mathbf{e}}_\omega = \dot{\boldsymbol{\omega}}_d - \dot{\boldsymbol{\omega}}$. Then the error state of the total system is expressed as $\mathbf{x}_e = (\mathbf{e}, \mathbf{e}_\omega)$. Despite of the state error, however, note that there are also errors between the mathematical model and the actual ball-pendulum system, which may cause deviations of the system performance from the desired under control inputs derived from the ideal model. This type of error refers to the model uncertainty. By involving the consideration of the model uncertainty, the error dynamics for the actual mechanical system are formulated as

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_r - \mathbf{A}\boldsymbol{\omega}_d + \mathbf{A}\mathbf{e}_\omega, \quad (2.73a)$$

$$\dot{\mathbf{e}}_\omega \equiv \mathbf{f}_{model} + \mathbf{d} = \mathbf{H}\boldsymbol{\tau} + \mathbf{h} + \mathbf{d}, \quad (2.73b)$$

where

$$\begin{aligned} \mathbf{H} &= \mathbf{H}_{oo}^{-1} \mathbf{H}_{op} (\mathbf{H}_{pp} - \mathbf{H}_{po} \mathbf{H}_{oo}^{-1} \mathbf{H}_{op})^{-1}, \\ \mathbf{h} &= \dot{\boldsymbol{\omega}}_d + \mathbf{H}_{oo}^{-1} \left(\mathbf{H}_{op} (\mathbf{H}_{po} \mathbf{H}_{oo}^{-1} \mathbf{h}_o - \mathbf{h}_p) + \mathbf{h}_o \right), \end{aligned}$$

and $\mathbf{d}$ represents for the effects caused by the inaccuracy of the mathematical model and the external disturbance. Due to that (2.73b) is underactuated and $\mathbf{H}^{-1}$ does not exist, we follow the same restructuring process conducted in Step 1. By introducing an auxiliary input $\boldsymbol{\tau}^* \in \mathbb{R}^3$ such that

$$\boldsymbol{\tau} = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top \boldsymbol{\tau}^* = \mathbf{H}^+ \boldsymbol{\tau}^*, \quad (2.75)$$

we reformulate (2.73b) as

$$\dot{e}_\omega = \mathbf{h} + \mathbf{d} + \mathbf{H}\mathbf{H}^+ \boldsymbol{\tau}^* + \boldsymbol{\tau}^* - \boldsymbol{\tau}^* = \boldsymbol{\tau}^* + \mathbf{h} + \boldsymbol{\delta}, \quad (2.76)$$

where $\boldsymbol{\delta} = (\mathbf{H}\mathbf{H}^+ - \mathbf{I})\boldsymbol{\tau}^* + \mathbf{d}$. Note that $\boldsymbol{\delta}$ represents the total influence caused by the general uncertainties of system (2.76). Then the control problem is restated as constructing $\boldsymbol{\tau}^*$ such that $\lim_{t \rightarrow \infty} e_\omega = \mathbf{0}$, with $\boldsymbol{\delta}$ unknown. Here we use a polynomial function to estimate $\boldsymbol{\delta}$ as

$$\boldsymbol{\delta} = (\delta_x, \delta_y, \delta_z) \approx \sum_{i=0}^N \boldsymbol{\delta}_i t^i. \quad (2.77)$$

These constant plant parameters $\boldsymbol{\delta}_i$ are estimated at each time t denoted by adjustable $\hat{\boldsymbol{\delta}}_i(t)$, referring to the control parameters, using an update law that we are to define from the following process. To incorporate both kinematics and dynamics, a feasible Lyapunov candidate function would be

$$\begin{aligned} V &= \frac{1}{2} \mathbf{e}^\top \mathbf{e} + \frac{1}{2} \sum_{i=0}^N \left(\hat{\mathbf{a}}_i(t) - \mathbf{a}_i \right)^\top \left(\hat{\mathbf{a}}_i(t) - \mathbf{a}_i \right) \\ &+ \frac{1}{2} \mathbf{e}_\omega^\top \mathbf{e}_\omega + \frac{1}{2} \sum_{i=0}^N \left(\hat{\boldsymbol{\delta}}_i(t) - \boldsymbol{\delta}_i \right)^\top \left(\hat{\boldsymbol{\delta}}_i(t) - \boldsymbol{\delta}_i \right), \end{aligned} \quad (2.78)$$

where the first line of (2.78) has the same structure the Lyapunov function candidate (2.68) selected for the kinematic steering system. As the plant parameters $\mathbf{a}_i$ and $\boldsymbol{\delta}_i$ in (2.78) are constant such that $\dot{\mathbf{a}}_i = \mathbf{0}$ and $\dot{\boldsymbol{\delta}}_i = \mathbf{0}$, the derivative of the Lyapunov function candidate is calculated as

$$\begin{aligned} \dot{V} &= \mathbf{e}^\top \left(\dot{\mathbf{x}}_r - \mathbf{A}\boldsymbol{\omega}_d + \mathbf{A}\mathbf{e}_\omega \right) + \sum_{i=0}^N \left(\hat{\mathbf{a}}_i(t) - \mathbf{a}_i \right)^\top \dot{\hat{\mathbf{a}}}_i(t) \\ &+ \mathbf{e}_\omega^\top \left(\boldsymbol{\tau}^* + \mathbf{h} + \sum_{i=0}^N \boldsymbol{\delta}_i t^i \right) + \sum_{i=0}^N \left(\hat{\boldsymbol{\delta}}_i(t) - \boldsymbol{\delta}_i \right)^\top \dot{\hat{\boldsymbol{\delta}}}_i(t). \end{aligned} \quad (2.79)$$

Note the first line of (2.79) varies from $\dot{V}_k$ expressed by (2.69), that its contains a factor $\mathbf{e}^\top \mathbf{A}\mathbf{e}_\omega$, which is a scalar function such that $\mathbf{e}^\top \mathbf{A}\mathbf{e}_\omega = \mathbf{e}_\omega^\top \mathbf{A}^\top \mathbf{e}$. Hence the derivative of the

Lyapunov function candidate (2.79) can be rewritten as

$$\dot{V} = \dot{V}_k + \mathbf{e}_\omega^\top \left(\mathbf{A}^\top \mathbf{e} + \boldsymbol{\tau}^* + \mathbf{h} \right) + \sum_{i=0}^N \hat{\boldsymbol{\delta}}_i(t)^\top \dot{\hat{\boldsymbol{\delta}}}_i(t) + \sum_{i=0}^N \boldsymbol{\delta}_i^\top \left(\mathbf{e}_\omega t^i - \dot{\hat{\boldsymbol{\delta}}}_i(t) \right). \quad (2.80)$$

Adopting the velocity input ω_d in the form of (2.72) constructed in Step 1, yields

$$\dot{V}_k = -\mathbf{e}^\top \mathbf{K} \mathbf{e}.$$

Then, as the controller $\boldsymbol{\tau}^*$ to be constructed cannot contain unmeasurable term, to eliminate unknown plant parameters $\boldsymbol{\delta}_i$ by bringing its coordinate $\mathbf{e}_\omega t^i - \dot{\hat{\boldsymbol{\delta}}}_i(t)$ to zero, define the update law

$$\dot{\hat{\boldsymbol{\delta}}}_i(t) = \mathbf{e}_\omega t^i, \quad (2.81)$$

which leads to

$$\dot{V} = -\mathbf{e}^\top \mathbf{K} \mathbf{e} + \mathbf{e}_\omega^\top \left(\boldsymbol{\tau}^* + \mathbf{h} + \mathbf{A}^\top \mathbf{e} + \sum_{i=0}^N \hat{\boldsymbol{\delta}}_i(t) t^i \right).$$

Constructing the auxiliary input

$$\boldsymbol{\tau}^* = -\left(\mathbf{A}^\top \mathbf{e} + \mathbf{h} + \mathbf{K}_\omega \mathbf{e}_\omega + \sum_{i=0}^N \hat{\boldsymbol{\delta}}_i(t) t^i \right),$$

where $\mathbf{K}_\omega$ is a positive definite matrix, yields

$$\dot{V} = \mathbf{e}^\top \dot{\mathbf{e}} + \mathbf{e}_\omega^\top \dot{\mathbf{e}}_\omega = -\mathbf{e}^\top \mathbf{K} \mathbf{e} - \mathbf{e}_\omega^\top \mathbf{K}_\omega \mathbf{e}_\omega \leq 0, \quad (2.82)$$

which implies the convergence of $\mathbf{e}$ and $\mathbf{e}_\omega$. The asymptotic stability of the closed-loop system can be proved similarly to the hoop-pendulum case. The actual input torque is calculated as

$$\boldsymbol{\tau} = -\mathbf{H}^+ \left(\mathbf{A}^\top \mathbf{e} + \mathbf{h} + \mathbf{K}_\omega \mathbf{e}_\omega + \sum_{i=0}^N \hat{\boldsymbol{\delta}}_i(t) t^i \right). \quad (2.83)$$

It should be noted that the error between the plant parameters and the control parameters $\hat{\delta}_i(t) - \delta_i$ are not necessarily asymptotically stable. However, according to (2.82) that the derivative of the Lyapunov function of the closed loop system is negative semi-definite, $\hat{\delta}_i(t) - \delta_i$ is bounded.

2.4.2 Case study

Note from (2.83) that the constructed adaptive controller varies from a conventional non-adaptive one that it contains an adaption mechanism

$$\tau_a = -\mathbf{H}^+ \sum_{i=0}^N \hat{\delta}_i(t) t^i,$$

that deals with the un-modeled factors for the ball-pendulum system. To demonstrate the effectiveness of the adaption mechanism, we conduct simulations for the perturbed hoop-pendulum system under the control law with and without τ_a respectively. We present simulation results for the ball-pendulum system tracking circular motion.

Let the reference trajectory for the displacement and orientation of the spherical robot be obtained by solving the following group of differential equations

$$\dot{\mathbf{x}}_r = \begin{bmatrix} 0 & R & 0 \\ -R & 0 & 0 \\ -\sin \psi^* / \cos v_o^* & -\cos \psi^* / \cos v_o^* & 0 \\ -\cos \psi^* & \sin \psi^* & 0 \\ -\sin \psi^* \tan v_o^* & -\cos \psi^* \tan v_o^* & -1 \end{bmatrix} \boldsymbol{\omega}_r,$$

mimicking the form of (2.2), where the reference angular velocity of the spherical robot $\boldsymbol{\omega}_r$ is specified as

$$\boldsymbol{\omega}_r = \left(\cos \frac{\pi t}{10}, -\sin \frac{\pi t}{10}, 0 \right),$$

that realizes a circular motion. Note that $\boldsymbol{\omega}_r$ is different from $\boldsymbol{\omega}_d$ which is the velocity input

to be designed. The vector that demonstrates the influence caused by the modeled system dynamics and the external disturbance, is assumed to be

$$\mathbf{d} = (50 \cos(3\pi t), 50 \sin(2\pi t), 0), \quad (2.84)$$

Set the initial state error as the position error $e_1(0) = 1.5\text{m}$ and $e_2(0) = 1.5\text{m}$, the orientation error $e_3(0) = e_4(0) = e_5(0) = \frac{\pi}{6}\text{rad}$ and the velocity error $e_6(0) = e_7(0) = e_8(0) = 0\text{rad/s}$. The numerical values used in the simulation are $m = 0.5\text{kg}$, $m_p = 1\text{kg}$, $R = 0.2\text{m}$, $l = 0.15\text{m}$, $g = 9.8\text{m/s}^2$, and the gain matrices $\mathbf{K} = \text{diag}(35, 20, 0.1, 1, 0.8)$ and $\mathbf{K}_\omega = \text{diag}(6, 5, 1)$.

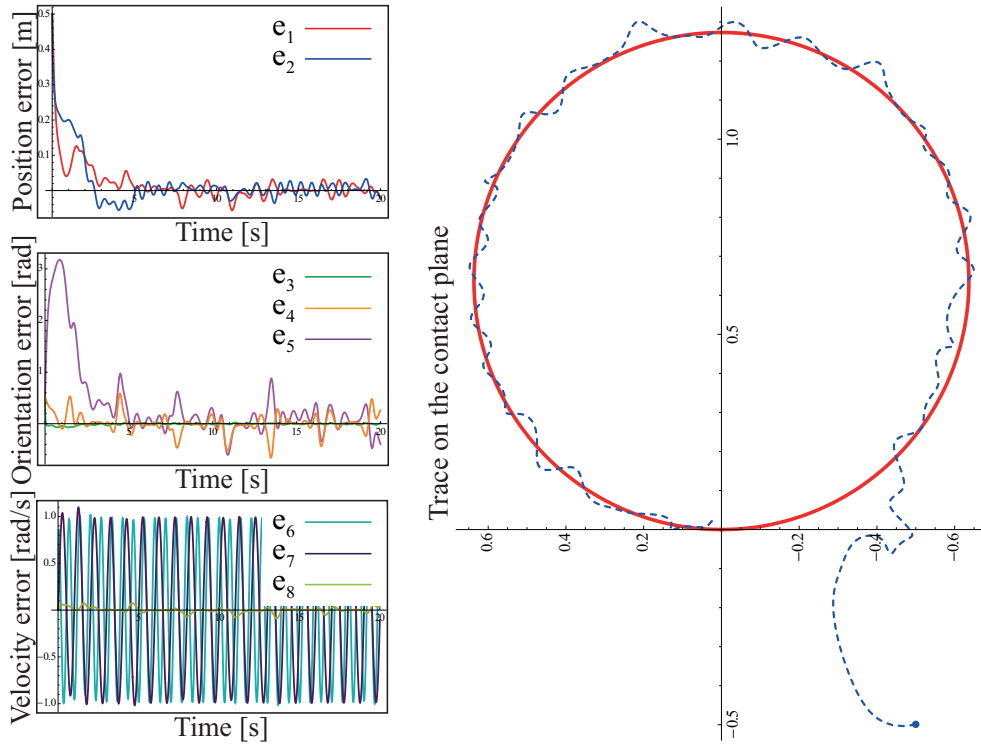


Figure 2.26: Motion of the ball-pendulum system under non-adaptive control.

The motion of the ball-pendulum system under non-adaptive and adaptive control laws are illustrated by Fig. 2.26 and Fig. 2.27. In the parametric plots, the blue dashed curve and the red curve represent for the actual and reference traces on the contact plane respectively. It can be seen from the following graphs that under the adaptive controller, even with un-

modeled dynamics and external disturbance, the system output still converges to the desired. The order of the polynomial basis functions is chosen to be two since a higher order may lead to overfitting problems. Note that a flat contact plane is assumed in the simulation which is impractical for the real application. The effect to the control system led by uneven terrain, needs to be taken into consideration in the future work.

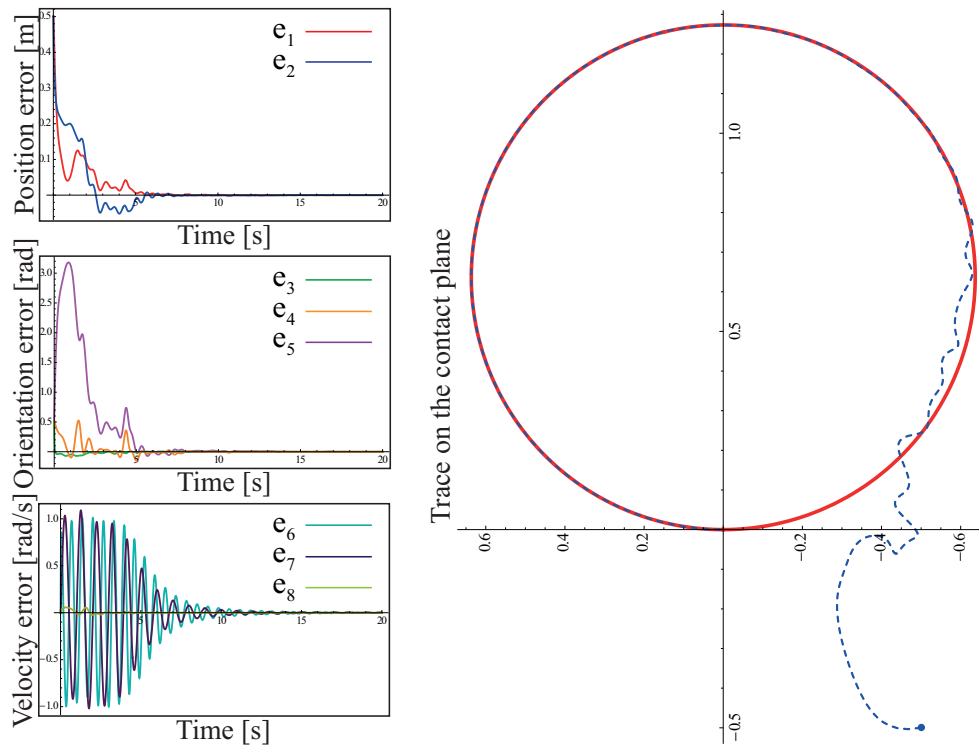


Figure 2.27: Motion of the ball-pendulum system under the FAT-based adaptive control.

2.5 Summary

One of the important research problems for control of mobile robot is to steer the robot from an initial configuration to the final configuration. For the spherical rolling robot we need not only to steer the position of the sphere but also the orientation of the sphere. The control for the orientation is important when it is necessary to attach some devices such as sensors or camera in or out of the shell, or when it is required to know where the robot is heading.

The steering problem seems to be a normal point to point stabilization problem which can be solved by constructing a continuous feedback. However, our rolling system is special because it is a nonholonomic system. According to the Brockett theory [57], there exists no continuous feedback for local asymptotic stabilization of an equilibrium point. The direct method does not work. However we can solve the steering problem step by step.

The main process is as followed. First we design a path connecting the start and the goal. A good path needs to satisfy the following condition, that if the sphere traces this path, it can reach the final destination with desired orientation. Second, with a given feasible path, we need to plan the motion trajectories for both the orientation and the position of the sphere when it is tracing this path. Here motion trajectory are the expression of the change of orientation and position of sphere with respect to time that describes how the sphere moves along the given path. This step is called motion planning. Third, we have obtained the desired motion trajectory from motion planning, but during the motion of sphere, there are always unexpected disturbances due to the friction or uneven terrain which cause the deviation of motion of robot from what we want. Therefore we need to construct a more robust control law such that the real motion of the sphere will always converge to the desired one. That is the error between the real and reference trajectories will converge to zero. We call this step the trajectory tracking.

It should be noted that path design problem has already been solved. In the robotics literature, many path planners have been proposed to generate a feasible path [71–75]. Therefore the rest of the work left for us are motion planning and trajectory tracking. These are also the two main parts of my master study.

Motion planning

A motion planning problem of steering a spherical robot from one configuration to another in rest-to-rest mode, has been addressed in Section 2.3. The problem was extended from kinematic level, designing a path that leads the robot to the goal configuration, to dynamic level that requires a desired motion with respect to time realized by an inner pendulum as the actuator. By assuming the contact path was given by a conventional planner, we proposed feedforward control law for the pendulum, which drives the spherical robot moving along

the contact path in rest-to-rest mode.

To develop feasible control laws for the pendulum, the dynamic model of the ball-pendulum system has been developed. It has been shown that, in general, the system kinematics and dynamics are coupled together, which makes the motion planning problem challenging. A notable exception is when the spherical robot undergoes pure rolling motion. Under the assumption that the sphere traces a given path in pure rolling mode, a reduced dynamic model has been developed. The reduced dynamic model, known as the hoop-pendulum system, was proved to be invariant to the shape of the contact curve. This property facilitates the construction of motion planning algorithms due to that the problem is simplified as planning motion for a planar system.

A rest-to-rest motion planning problem for the hoop-pendulum system could be solved through optimal control formulation seeking for minimum effort. However the optimal control based motion planning may be sensitive to the given boundary conditions and may lead to a heavy computational cost. Inspired by the geometric phase approach, a simpler feedforward motion planning algorithm, based on the parameterization of the pendulum angle by the 2-nd order derivative of the 4-th order Beta-function, has been constructed. The algorithm has lower computational cost and is thus more suitable for real-time applications. The algorithm has been verified under simulation for four cases, tracing straight line, circular curve, Viviani's curve and the Loxodrome.

It should be noted that in Section 2.3 we dealt only with pure rolling motion. The development of control algorithms for motion with instantaneous spinning will be addressed in a separate section. Also the effect of friction at the contact point was not considered. We plan to bring it into consideration in future work.

Trajectory tracking

A trajectory tracking problem for the ball-pendulum system has been considered in Section 2.4. An adaptive controller based on the backstepping technique was first constructed for the planar hoop-pendulum system with time-varying uncertainties and then extended to the three dimensional ball-pendulum system. The convergence of system motion to the desired trajectories was proved and verified by simulations for tracking circular motions under

perturbation.

In Section 2.4, the ball-pendulum system, of which both the kinematics and dynamics are underactuated, played an important role as the testbed for our modified FAT-based adaptive control algorithm. In the controller design process, there are two noteworthy points. First, we restructured the kinematics and dynamics of the rolling robot respectively, as the combination of a fully-actuated system and the variation from it. The variation part was then included into the general uncertainty term that represents for the total influence to the control process caused by mismeasured system parameters, un-modeled dynamics and external perturbation. Second, by parameterizing the general uncertainty term with selected basis functions weighted by unknown parameters, we simplified an adaptation problem for system with time-varying uncertainties to a parameter identification problem.

Note that the selection of basis functions depends on the types of mechanical systems and system uncertainties. To widen the range of application of our modified FAT-based algorithm on a class of underactuated systems, in the future study we plan to analyze the advantages and disadvantages of polynomial basis functions which we chose in Section 2.4, and other possible candidates. It may give rise to an interesting topic of how to select or design basis functions based on the consideration of the accuracy of their parametrization of the uncertainty term and the computational cost. On the practical side, we plan to conduct the experimental verification of the developed controller.

Chapter 3

Geometric phase based motion planning

3.1 Background

In recent years there has been a growing interest to the control of underactuated mechanical systems [2–5]. These systems have fewer independent actuators than the number of degrees of freedom. The need for analysis and control of underactuated mechanical systems arises in many practical applications that include the use of ships, underwater vehicles, helicopters, aircraft, satellites, mobile robots, space platforms, flexible joint robots, hyper-redundant and snake-like manipulator, walking robots, capsule robots, and hybrid machines [2–5].

One of the important problems in the control of underactuated systems is the construction of motion planning algorithms. Typically, the motion planning problem requires to steer the system from an initial state to a given final state. Different from the fully-actuated systems, for underactuated systems, the underactuation feature forms a nonholonomic constraint between the configuration variables. This constraint makes the motion planning problem challenging due to that the generated motion trajectory needs to respect the constraint.

In this chapter we propose a motion planning technique for a class of base-free underactuated systems that are partially differentially flat [34]. Our work is based on the concept of fiber and base variables, the definition of which is as follows [35]. The configuration variables of mechanical systems include: the base variables which describe the internal shape changes of the system and the fiber space that represents the systems position and orientation

change.

Our motion planning algorithm is constructed as follows. First, through partial feedback linearization with respect to the base variable, the role the dynamic constraint plays in the motion planning process is highlighted. The base variable is treated as the input while the fiber variable as the output. Then the rest-to-rest motion planning problem is reformulated as finding a time trajectory for the base variable, such that the corresponding fiber variable satisfies the rest-to-rest boundary condition. Next, a Beta-function weighted by a constant parameter is selected as a candidate function for the base variable, which is shown to be an optimal solution for the linearized system. Its feasibility for the nonlinear system is then proved by showing that the corresponding fiber variable can satisfy the given boundary conditions. Finally the parameter of the Beta-function is adjusted such that a desired shift for the fiber variable can be obtained. Under the proposed motion planning algorithm, rest-to-rest trajectories for the configuration variables are generated with less computational cost compared with the optimal control formulation.

The rest of this chapter is organized as follows. First, in Section 3.2 we formulate the rest-to-rest motion planning problem for a class of underactuated systems and provide corresponding dynamic equations. In Section 3.3, we propose a geometric phase based algorithm and test its feasibility for generating rest-to-rest motion trajectories for the configuration variables. Then in Section 3.4 the proposed motion planner is verified under simulations for several underactuated systems. Discussion is conducted in and finally, conclusions are drawn in Section 3.5.

3.2 Special feature of partial differentially flat systems

The motion equations of the 2DOF underactuated systems with one input τ are

$$M(q)\ddot{q} + h(q, \dot{q}) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau, \quad (3.1)$$

where

$$\mathbf{q} = \begin{bmatrix} q_f \\ q_b \end{bmatrix}, \mathbf{M} = \begin{bmatrix} m_{ff} & m_{fb} \\ m_{bf} & m_{bb} \end{bmatrix}, \mathbf{h} = \begin{bmatrix} h_f \\ h_b \end{bmatrix}.$$

Here in the equations, $\mathbf{q}$ is the vector for the generalized coordinates where q_f and q_b are respectively the fiber and the base variables. The base variable describes the internal shape changes of the system and the fiber variable represents the systems position and orientation change.

In this chapter, we consider motion planning for a class of systems for which $\mathbf{M}$ and $\mathbf{h}$ contain only base variable q_b and its time derivative $\dot{q}_b$. This special class of systems includes some well-known underactuated systems such the cart-pole, the ballbot system [34] and segway [96], etc.

Note that the underactuated portion of the motion equations (first line of (3.1)) forms a nonholonomic constraint. For systems with nonholonomic constraint, a closed loop phase change of the base variable can lead to a non-zero phase change of the fiber variable, as illustrated in Fig. 3.1. In this figure, F is the set of fiber variables and the B is the manifold for the base space. A trivial principle fiber bundle with B and F consists of the configuration manifold $Q = B \times F$ to which $\mathbf{q}$ belongs [1].

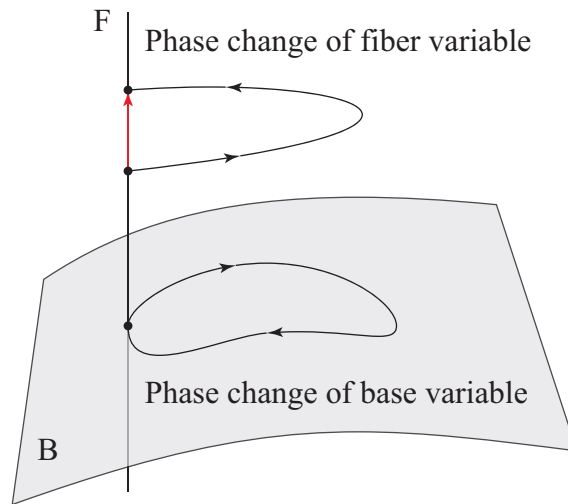


Figure 3.1: A closed loop in shape variables producing a displacement in the fiber variable.

In the locomotion problem, this property implies that a periodic internal shape change of underactuated system steers its position and orientation. Therefore, one can select the base variable q_b to be the input in the motion planning problem. The selection of q_b as input is implemented by the partial feedback linearization for (3.1) with respect to q_b conducted below. From first line of (3.1), one can obtain

$$\ddot{q}_f = -m_{ff}^{-1}(m_{fb}\ddot{q}_b + h_f), \quad (3.2)$$

and by substituting (3.2) into (3.1), the second line of (3.1) is rewritten as

$$\ddot{q}_b = -(m_{bb} - m_{bf}m_{ff}^{-1}m_{fb})^{-1}(\tau + h_b). \quad (3.3)$$

Note that during this process, a singular point may appear when $m_{ff} = 0$, which needs to be avoided when generating the motion trajectories. Selecting

$$\tau = -(m_{bb} - m_{bf}m_{ff}^{-1}m_{fb})u + h_b, \quad (3.4)$$

and substituting (3.4) back into (3.1), gives

$$\ddot{q}_f = -m_{ff}^{-1}h_f - m_{ff}^{-1}m_{fb}u, \quad (3.5)$$

$$\ddot{q}_b = u. \quad (3.6)$$

where u is the auxiliary input. Then by choosing a state vector $\mathbf{x} = (q_f, \dot{q}_f, q_b, \dot{q}_b)$, (3.5) and (3.6) can be written in the state space form as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \quad (3.7)$$

where

$$\mathbf{f}(\cdot) = \begin{bmatrix} \dot{q}_f \\ -m_{ff}^{-1}h_f \\ \dot{q}_b \\ 0 \end{bmatrix}, \quad \mathbf{g}(\cdot) = \begin{bmatrix} 0 \\ -m_{ff}^{-1}m_{fb} \\ 0 \\ 1 \end{bmatrix}.$$

Note that for a class of 2DOF base-free systems (including cart-pole, hoop-pendulum, planar ballbot and segway), which also refer to the shape-accelerate systems [34], the 2-nd order derivative of the fiber variable q_f can be expressed purely by q_b and its derivatives as

$$\ddot{q}_f = h(q_b, \dot{q}_b, u). \quad (3.8)$$

Therefore, the underactuated system (3.7) is partially differentially flat [34] where the base variable q_b is the flat output. This property of (3.7) facilitates the construction of motion planning algorithms because it can be shown that after the linearization around $q_b = 0$ and $\dot{q}_b = 0$, the controllable canonical form of (3.7) is a 4-th order integrator. Indeed, the linearization of (3.7) around $q_b = \dot{q}_b = 0$ is expressed as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u, \quad (3.9)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ b \\ 0 \\ 1 \end{bmatrix},$$

and a_{23} , a_{24} , b are constants. According to (3.8), the second row of matrix $\mathbf{A}$ has two non-zero items a_{23} and a_{24} due to that $\ddot{q}_f$ contains no q_f or $\dot{q}_f$ terms.

To transform the linearized model (3.9) to the controllable canonical form, one defines

a transformation matrix P by

$$P = \begin{bmatrix} A^3b & A^2b & Ab & b \end{bmatrix}. \quad (3.10)$$

By setting $x = P\hat{x}$, one obtains the controllable canonical form of (3.7) as

$$\dot{\hat{x}} = P^{-1}APx + P^{-1}bu = \hat{A}\hat{x} + \hat{b}u. \quad (3.11)$$

where

$$\hat{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \hat{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

This special form (3.11) is later on shown to be helpful in the construction of our planning algorithm.

3.3 Geometric phase based motion planning

Based on (3.7), the rest-to-rest motion planning problem is stated as defining u such that the corresponding trajectories for the state x satisfy the given boundary conditions

$$x(0) = (0, 0, 0, 0), \quad x(T) = (q_{\text{des}}, 0, 0, 0). \quad (3.12)$$

where T is the movement time.

Note the optimal control formulation provides a generic way to plan the rest-to-rest motion, but it comes with a high computational cost and the algebraic relation between the configuration variables is not analytically known. In this chapter, we propose a systematic way for capturing the optimal motion pattern by a polynomial function that satisfies the given boundary conditions. The resulting motion planner is less computationally expensive and thus more suitable for real-time application.

3.3.1 Generation of the rest-to-rest trajectory

To construct a less computational expensive but feasible motion planner for system (3.7) satisfying the rest-to-rest boundary conditions (3.12), we use the minimum effort optimal solution (the optimal criterion is $\frac{1}{2} \int u^2 dt$) for the linearized system, which is shown later on to be a polynomial, as a candidate function for the base variable $q_b(t)$. Then we prove its feasibility by showing that the corresponding trajectories of $\mathbf{x}(t)$ satisfy the given boundary conditions.

The analytical solution for the minimum effort optimal control problem is well established in the control literature, which is represented as

$$\mathbf{x}(t) = e^{\hat{\mathbf{A}}t} \left((\mathbf{I} - \mathbf{W}(t)\mathbf{W}(T)^{-1})\mathbf{x}(0) + \mathbf{W}(t)\mathbf{W}(T)^{-1}e^{-\hat{\mathbf{A}}T}\mathbf{x}(T) \right), \quad (3.13)$$

where

$$\mathbf{W}(t) = \int_0^t e^{-\hat{\mathbf{A}}s} \hat{\mathbf{b}}\hat{\mathbf{b}}^\top e^{-\hat{\mathbf{A}}^\top s} ds. \quad (3.14)$$

As $\hat{\mathbf{A}}$ is a nilpotent matrix for base-free systems, a finite solution for $\hat{\mathbf{x}}(t)$ can be obtained. For the rest-to-rest motion, boundary conditions are as (3.12) and $e^{-\hat{\mathbf{A}}T}\mathbf{x}(T) = \mathbf{x}(T)$. Setting the nominal time $\hat{t} = t/T$ and with binomial expansion for $(1 - \hat{t})^3$, one obtains

$$q_f(\hat{t}) = \frac{\int_0^T p^3(1-p)^3 dp}{B(4,4)} = \frac{B(\hat{t}; 4, 4)}{B(4, 4)}, \quad (3.15)$$

where $B(\hat{t}; 4, 4)$ is the regularized incomplete Beta-function. In what follows, as we deal with the 4-th order systems only, $B(\hat{t}; 4, 4)$ will be written as $B(\hat{t})$ unless otherwise specified.

The base variable $q_b(t)$ is the 2-nd derivative of $q_f(t)$ shown by Fig. 3.2, because the linearized system is a 4-th order integrator. Note that $q_b(t)$ is an odd function with respect to the middle point where $t = T/2$. It implies $|q_{\max}| = |q_{\min}|$ where $q_{\max}$ and $q_{\min}$ are constants indicating the maximum and minimum values of $q_b(t)$.

For the sake of simplicity, we represent $q_b(t)$ in the form of a polynomial whose maximum value is 1, weighted by $q_{\max}$. One way to do so is to find the location $t_{\max}$ of the

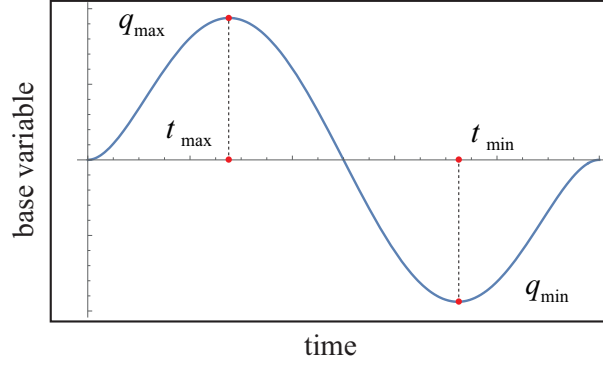


Figure 3.2: Motion pattern of the based variable $q_b(t)$

maximum point, calculating the corresponding value $q_b(t_{\max})$, and rendering it to 1. Note the zero-velocity points of $q_b(t)$ are found at $t = 0$, $t = t_{\max}$, $t = t_{\min}$ and $t = T$ where

$$t_{\max} = \left(\frac{5 - \sqrt{5}}{10}\right)T, \quad t_{\min} = \left(\frac{5 + \sqrt{5}}{10}\right)T. \quad (3.16)$$

Thus, the value of $q_{\max} = q_b(t_{\max})$ can be easily calculated and the base variable $q_b(t)$ can be expressed as

$$q_b(t) = q_{\max} \left(\frac{25\sqrt{5}t^2(t-T)^2(2t-T)}{T^5} \right). \quad (3.17)$$

Then the auxiliary input u is calculated as the second order derivative of $q_b(t)$.

3.3.2 Feasibility of the linearized solution

We now prove that (3.17) is a feasible candidate function for the base variable $q_b(t)$ to realize a rest-to-rest motion for the original nonlinear underactuated system (3.7). Given the nonlinear differential equation (3.7), with the initial conditions $q_f(0) = \dot{q}_f(0) = 0$, we prove that $\dot{q}_f(T) = 0$. The proof is sketched as follows.

Note that $q_b(t)$ is the 2-nd derivative of Beta-function, and Beta-function has the property that $B(t) = B(T) - B(T - t)$. Therefore, all its even derivatives are odd (asymmetric) and odd derivatives are even (symmetric) with respect to the middle point $t = T/2$. By

introducing the new time coordinate $\tilde{t} = t - T/2$, one obtains

$$q_b(\tilde{t}) = q_b(t - \frac{T}{2}), \quad \dot{q}_b(\tilde{t}) = \dot{q}_b(t - \frac{T}{2}), \quad \ddot{q}_b(\tilde{t}) = \ddot{q}_b(t - \frac{T}{2}),$$

and one gets

$$q_b(-\tilde{t}) = -q_b(\tilde{t}), \quad \dot{q}_b(-\tilde{t}) = \dot{q}_b(\tilde{t}), \quad \ddot{q}_b(-\tilde{t}) = -\ddot{q}_b(\tilde{t}). \quad (3.18)$$

By substituting (3.18) into (3.5), one obtains the 2-nd derivative of the fiber variable $\ddot{q}_f$ as a function of $\tilde{t}$. If

$$\ddot{q}_f(-\tilde{t}) = -\ddot{q}_f(\tilde{t}), \quad (3.19)$$

holds true, which implies that $\ddot{q}_f(t)$ is odd with respect to $t = T/2$, one obtains that with the initial condition $\dot{q}_f(0) = 0$,

$$\int_0^T \ddot{q}_f(t) dt = \dot{q}_f(T) - \dot{q}_f(0) = 0, \quad (3.20)$$

which leads to the conclusion that $\dot{q}_f(T) = 0$. Therefore, the selected $q_b(t)$ is a feasible candidate for the rest-to-rest motion planning for the underactuated system (3.7).

It is important to notice that only when $\ddot{q}_f(-\tilde{t}) = -\ddot{q}_f(\tilde{t})$ holds true, the geometric phase based algorithm works. Although a generic proof of this property for all base-free systems is yet to be proposed, one can easily prove it for common base-free systems such as cart-pole, hoop-pendulum [32] and planar ballbot system [34].

3.3.3 Computation of the parameter $q_{\max}$

Choose the base variable $q_b(t)$ as (3.17), where the parameter $q_{\max}$ is a constant left unknown. Correspondingly, with the initial conditions of the fiber variable $q_f(0) = 0$ and $\dot{q}_f(0) = 0$, the final velocity $\dot{q}_f(T) = 0$ has been proved in the previous subsection. The rest of the boundary condition of $q_f(t)$ that $q_f(T) = q_{\text{des}}$, needs to be satisfied by identifying a suitable $q_{\max}$. The identification problem can be formulated as follows. For (3.8) and (3.17), determine $q_{\max}$ such that $q_f(t)$ satisfies $q_f(T) = q_{\text{des}}$.

To determine the value of $q_{\max}$, one sets it as an independent variable and add one more

differential equation [86]

$$\dot{q}_{\max}(t) = 0. \quad (3.21)$$

It should be noted that (3.8) is a 2-nd order differential equation that can carry two boundary conditions: $(q_f(0), q_f(T))$ or $(q_f(0), \dot{q}_f(0))$, while (3.21) can carry only one. However, there are four required boundary conditions to satisfy: $(q_f(0), q_f(T), \dot{q}_f(0), \dot{q}_f(T))$. To match the number of equations for solving the boundary value problem, we introduce new variable $z(t)$, where

$$\dot{z}(t) = q_f(t), \quad (3.22)$$

by which (3.8) is transformed into a third order differential equation that can carry three boundary conditions. Choosing $z(t)$ and $q_{\max}(t)$ as new independent variables, we restructure the system of differential equations as

$$\ddot{z}(t) = h(q_b(t), \dot{q}_b(t), \ddot{q}_b(t)), \quad \dot{q}_{\max}(t) = 0, \quad (3.23)$$

with the following four boundary conditions

$$\dot{z}(0) = 0, \quad \ddot{z}(0) = 0, \quad \dot{z}(T) = q_{\text{des}}, \quad \ddot{z}(T) = 0. \quad (3.24)$$

Solving the boundary value problem is equivalent to solving the rest-to-rest motion planning problem. Thus, through the geometric phase based motion planning, rest-to-rest trajectories for the underactuated system (3.7) are generated explicitly with less computational cost compared with the optimal control formulation.

Note that under the geometric phase based method, $q_b(t)$ is a polynomial expressed by (3.17), and $q_f(t)$ is obtained through the integration of (3.8). Under the optimal control formulation, on the other hand, for the 4-th order systems, a group of 8 differential equations, including the 4 state equations and 4 costate equations, need to be solved. Compared with the optimal control formulation, the geometric phase based method tends to be less computationally expensive while the motions of the cart-pole system resulted from both planners are very similar. Upon this fact, the geometric phase based method is more suitable for the real-time applications.

3.4 Case study

Here we provide examples of the application of the proposed motion planning algorithm on several underactuated systems. The comparison of the system performance between the proposed motion planner and the optimal control formulation is also conducted.

Cart-pole system

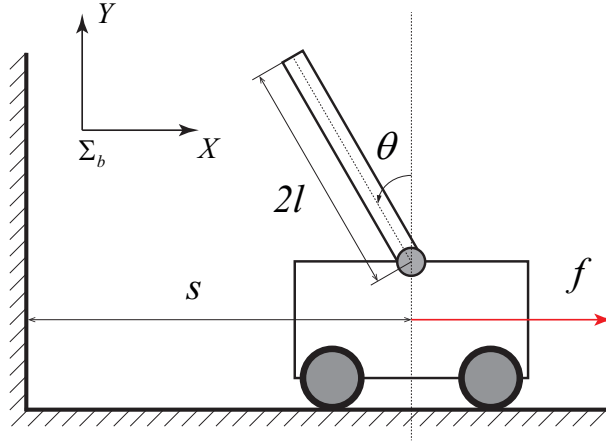


Figure 3.3: Cart-pole system

The cart-pole system, as shown in Fig. 3.3, is a well-known test bed for nonlinear control techniques. It is combined by a sliding cart, on which a inverted pendulum is mounted, connecting by a passive pin joint. The displacement of the cart is denoted by s and the pendulum angle by θ , increasing counterclockwise. The input force f is acting on the cart. The dynamic model of the cart-pole system is obtained [32] as

$$\begin{bmatrix} H_{cc} & H_{cp} \\ H_{pc} & H_{pp} \end{bmatrix} \begin{bmatrix} \ddot{s} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} h_c \\ h_p \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} f, \quad (3.25)$$

where

$$\begin{aligned} H_{cc} &= m + m_p, \quad H_{cp} = -m_p l \cos \theta, \quad h_c = m_p l \sin \theta \dot{\theta}^2, \\ H_{pc} &= -m_p l \cos \theta, \quad H_{pp} = J_p, \quad h_p = m_p g l \sin \theta, \end{aligned}$$

and m, m_p stand for the mass of the cart and of the pendulum and g for the standard gravitational acceleration. Here one assumes that the pendulum to be a uniform beam of length $2l$. Thus its inertia momentum with respect to the pin joint $J_p = \frac{1}{12}m_p(2l)^2 + m_p l^2 = \frac{4}{3}m_p l^2$.

Then following the procedure from (3.2) to (3.4), one conducts the partial linearization for the cart-pole system driven by cart. By selecting the driving force

$$f = \frac{J_p(m + m_p) - (m_p l \cos \theta)^2}{m_p l \cos \theta} u + m_p g l \sin \theta \dot{\theta}^2, \quad (3.26)$$

and through partial feedback linearization, one obtains

$$\ddot{s} = h(\theta, \dot{\theta}, \ddot{\theta}) = \frac{J_p}{m_p l \cos \theta} \ddot{\theta} + g \tan \theta, \quad (3.27)$$

$$\ddot{\theta} = u. \quad (3.28)$$

As the passive pendulum angle θ is the base variable, the linearization is non-collocated. Note a singularity appears at $\theta = \pi/2$ (see (3.27)). Then the amplitude of θ should be limited with the range from 0 to $\pi/2$ in the motion planning.

For (3.27), the condition (3.19) is satisfied if the base variable θ is chosen as (3.17). Hence, the proposed geometric phase based algorithm can be applied. The rest-to-rest motion planning problem is stated as finding a feasible trajectory for θ such that the corresponding s satisfies the following boundary conditions

$$s(0) = 0, \quad \dot{s}(0) = 0, \quad s(T) = s_{\text{des}}, \quad \dot{s}(T) = 0. \quad (3.29)$$

The candidate function for the input based variable θ is chosen in the form (3.17) with the constant parameter q_{max} to be determined. By introducing the new variable $\dot{z}(t) = s(t)$, one determines q_{max} through the process shown in Section 3.3.3 with the following four

boundary conditions

$$\dot{z}(0) = 0, \ddot{z}(0) = 0, \dot{z}(T) = s_{\text{des}}, \ddot{z}(T) = 0. \quad (3.30)$$

Solving the above boundary value problem is equivalent to solving the motion planning problem for the cart-pole system undergoing rest-to-rest motion to the desired state.

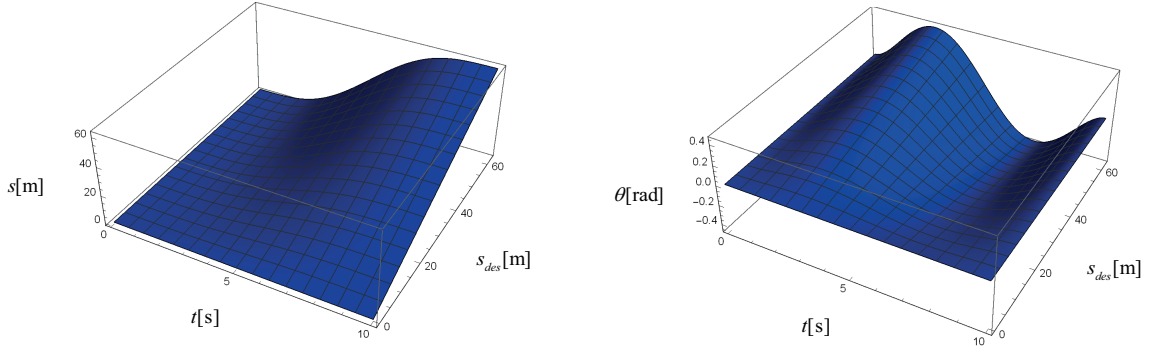


Figure 3.4: Motion of the cart-pole system resulted from the geometric phase based control.

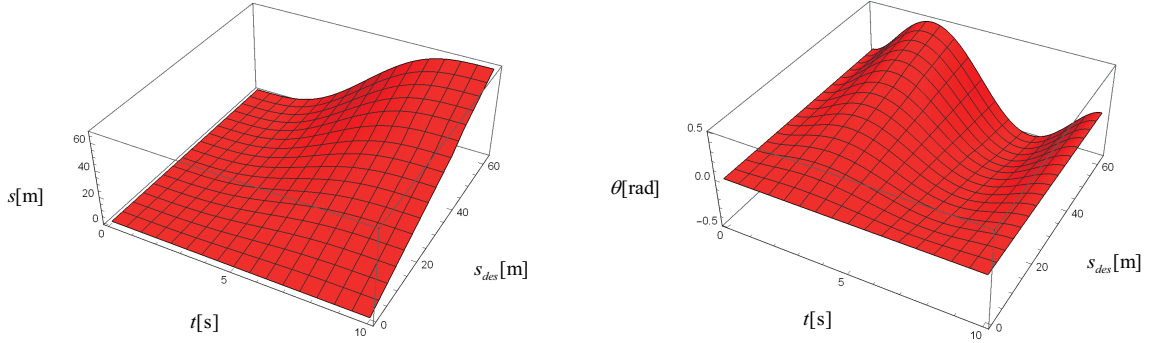


Figure 3.5: Motion of the cart-pole system resulted from the minimum effort optimal control.

To analyze the performance of the cart-pole system under proposed planner compared with that under optimal control formulation, we conduct simulations with respect to various desired shift s_{des} of the fiber variable s . The value of s_{des} changes from 0 to 20π . We plot the configuration variables s and θ as functions of both s_{des} and time t . System motion under the geometric phase based planner is shown in Fig. 3.4 by blue surfaces whereas motion under

optimal control in Fig. 3.5 by red surfaces. The numerical values used in the simulation are $m = 1\text{kg}$, $m_p = 1\text{kg}$, $l = 0.1\text{m}$, $g = 9.8\text{m/s}^2$ and the time interval $T = 10\text{s}$.

By comparing the simulation results of the optimal control and the geometric phase based approach, one finds that the cart motions under both planners are almost the same whereas the pendulum motions differ by less than 0.1rad in maximum. For both motion planners, the comparison between the values of the cost function J for the cart-pole system is illustrated by Table 3.1. Note that the value of J for the geometric phase based approach is only 0.038% higher than that of the optimal control in maximum, which implies that both motion planners provide similar system performance. Upon this fact, the geometric phase base approach which has lower computational cost, is more suitable for the real-time applications.

ϕ_{des}	2π	4π	6π	8π	10π
Optimal control	0.00206971	0.00825442	0.0184816	0.0326325	0.0505456
Geometric phase	0.00206971	0.00825443	0.0184817	0.0326329	0.0505469
ϕ_{des}	12π	14π	16π	18π	20π
Optimal control	0.0720210	0.0968251	0.124697	0.155355	0.188503
Geometric phase	0.0720247	0.0968346	0.124718	0.155395	0.188575

Table 3.1: Comparison for the values of the cost function J , expressed by (2.30), between the minimum effort optimal control formulation and the geometric phase based approach for the cart-pole system.

Planar ballbot system

The planar ballbot system, illustrated by Fig. 3.6, is a mobile robot that moves on a rolling hoop. The generalized coordinates of the system are denoted by the hoop angle $\phi(t)$ and the body angle with respect to the ground $\theta(t)$, both of which are increasing counter-clockwise. The red dot on the hoop is its initial contact point to the ground. One revolution of the initial contact point implies a $2\pi R$ linear displacement for the center of the hoop. Note that at the starting configuration ($\phi(0) = 0$), the initial contact point touches the ground. The input torque τ is acting on the joint between the hoop and the body, also increasing

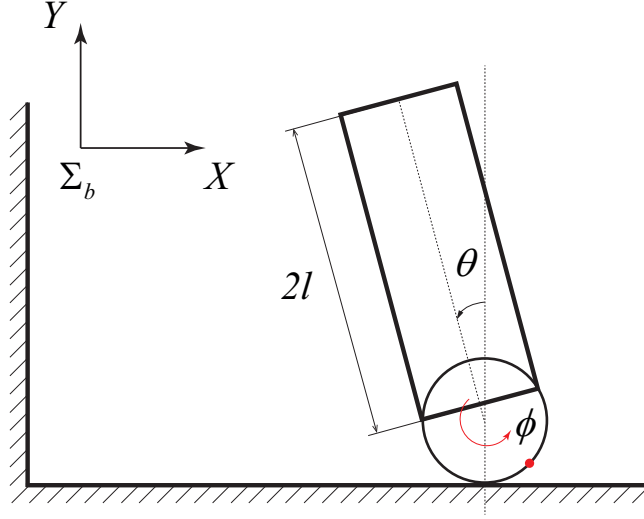


Figure 3.6: Planar ballbot system.

counter-clockwise. Then the dynamic model is obtained [32] as

$$\left(J_c - m_p R l \cos \theta\right) \ddot{\phi} + \left(J_p - m_p R l \cos \theta\right) \ddot{\theta} + m_p l \sin \theta \left(R \dot{\theta}^2 + g\right) = 0, \quad (3.31)$$

$$-m_p R l \cos \theta \ddot{\phi} + J_p \ddot{\theta} + m_p g l \sin \theta = \tau, \quad (3.32)$$

where $J_c = (\frac{4}{3}m + m_p)R^2$. The masses for the hoop and for the body are denoted by m and m_p . R stands for the radius of the hoop and g for the standard gravitational acceleration. Here one assumes that the body to be a uniform beam of length $2l$. Thus its inertia momentum with respect to the center of the hoop $J_p = \frac{1}{12}m_p(2l)^2 + m_p l^2 = \frac{4}{3}m_p l^2$.

Then, following the procedure from (3.2) to (3.4), one conducts the partial linearization for the ballbot system. By selecting the input

$$\tau = -\frac{J_p J_c - (m_p R l \cos \theta)^2}{J_c - m_p R l \cos \theta} u + m_p g l \sin \theta, \quad (3.33)$$

and through partial feedback linearization, one obtains

$$\ddot{\phi} = h(\theta, \dot{\theta}, \ddot{\theta}) = -\frac{J_p - m_p R l \cos \theta}{J_c - m_p R l \cos \theta} \ddot{\theta} - \frac{m_p l \sin \theta (R \dot{\theta}^2 + g)}{J_c - m_p R l \cos \theta}, \quad (3.34)$$

$$\ddot{\theta} = u. \quad (3.35)$$

The actuated body angle θ is the base variable and thus, the linearization is collocated.

Note that for (3.34), the condition (3.19) is satisfied if the base variable θ is chosen as (3.17). Hence, the proposed geometric phase based algorithm can be applied. The rest-to-rest motion planning problem is stated as finding a feasible trajectory for θ such that the corresponding ϕ satisfies the following boundary conditions

$$\phi(0) = 0, \dot{\phi}(0) = 0, \phi(T) = \phi_{\text{des}}, \dot{\phi}(T) = 0. \quad (3.36)$$

The candidate function for the input based variable θ is chosen in the form (3.17) with the constant parameter q_{max} to be determined. By introducing the new variable $\dot{z}(t) = \dot{\phi}(t)$, one determines q_{max} through the process shown in Section 3.3.3 with the following four boundary conditions

$$\dot{z}(0) = 0, \ddot{z}(0) = 0, \dot{z}(T) = \phi_{\text{des}}, \ddot{z}(T) = 0. \quad (3.37)$$

Solving the above boundary value problem is equivalent to solving the motion planning problem for the cart-pole system undergoing rest-to-rest motion to the desired state.

To analyze the performance of the ballbot system under proposed planner compared with that under optimal control formulation seeking for the minimum effort (the optimal criterion is $\frac{1}{2} \int u^2 dt$), we conduct simulations with respect to various desired shift ϕ_{des} of the fiber variable ϕ . The value of ϕ_{des} changes from 0 to 20π . We plot the configuration variables ϕ and θ as functions of both ϕ_{des} and time t . Motion under the geometric phase based planner is shown in Fig. 3.7 by blue surfaces whereas motion under optimal control in Fig. 3.8 by red surfaces. The numerical values used in the simulation are $m = 2\text{kg}$, $m_p = 1\text{kg}$, $R = 0.2\text{m}$, $l = 0.3\text{m}$, $g = 9.8\text{m/s}^2$ and the time interval $T = 10\text{s}$.

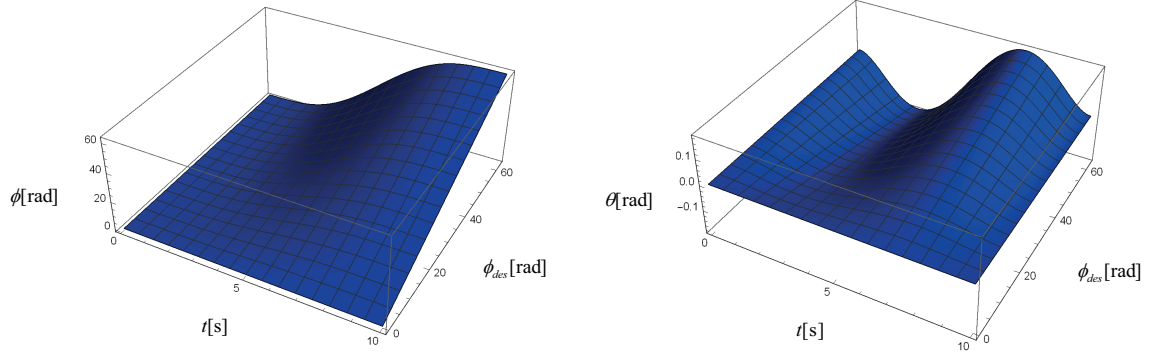


Figure 3.7: Motion of the ballbot system resulted from the geometric phase based control.

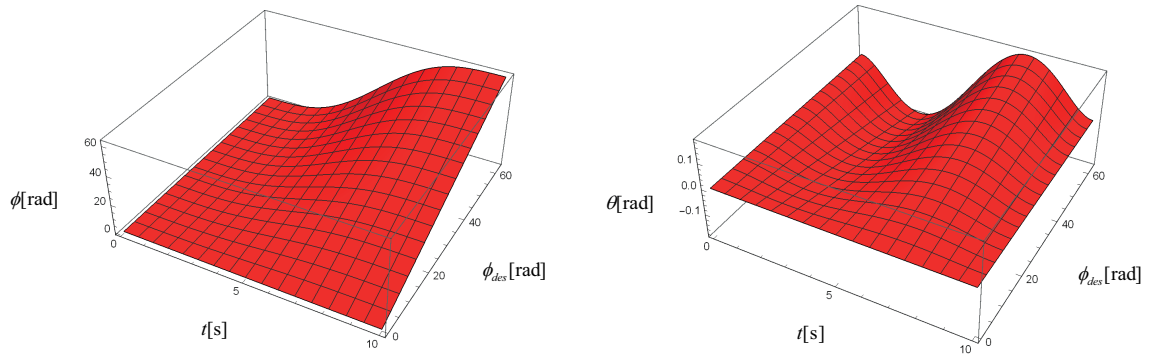


Figure 3.8: Motion of the ballbot system resulted from the minimum effort optimal control.

Similar to the cart-pole system, for the ballbot system, the hoop motions resulted from the optimal control and the geometric phase base approach, are almost the same whereas the body motions differ by less than 0.2rad in maximum. The comparison between the values of the cost function J for the ballbot system under both motion planners, is illustrated by Table 3.2. Note that the value of J for the geometric phase based approach is only 0.0013% higher than that of the optimal control in maximum, which implies that both motion planners provide similar system performance. Upon this fact, the less computationally expensive geometric phase based method is more suitable for the real-time applications.

ϕ_{des}	2π	4π	6π	8π	10π
Optimal control	0.00029573	0.0011836	0.00266544	0.00474466	0.00742581
Geometric phase	0.00029573	0.0011836	0.00266544	0.00474466	0.00742582
ϕ_{des}	12π	14π	16π	18π	20π
Optimal control	0.0107149	0.0146193	0.0191479	0.0243110	0.0301207
Geometric phase	0.0107149	0.0146194	0.0191480	0.0243113	0.0301211

Table 3.2: Comparison for the values of the cost function J , expressed by (2.30), between the minimum effort optimal control formulation and the geometric phase based approach for the ballbot system.

3.5 Summary

This chapter deals with rest-to-rest motion planning problem for planar underactuated systems. We classified the planar underactuated systems into two types. One is the base-fixed systems that one or more portions of the systems are fixed on the ground, including the serial robot with passive joints. The other is the base-free type whose position and orientation can be steered freely by internal shape change, such as the cart-pole system, the planar ballbot system and the ballbot system. In this chapter, we focused on the base-free type systems.

After the partial feedback linearization, this type of systems are commonly partially differentially flat and the base variable is the flat output. It leads to a property that the derivatives of the fiber variable can be represented purely by the base variable and its derivatives. Based on this property a motion planning algorithm has been constructed as follows.

First, the 2-nd order derivative of Beta-function weighted by a constant parameter is selected as a candidate for the base variable. Its feasibility is proved by showing that the corresponding fiber variable can satisfy the given boundary conditions. Finally the parameter of the 2-nd order derivative of Beta-function is adjusted such that a desired shift for the fiber variable can be obtained. Under the proposed motion planning algorithm, rest-to-rest trajectories for the configuration variables are generated with similar system performance but with less computational cost, compared with the optimal control formulation.

There are several points to be improved for the proposed algorithm in the future work. Firstly, the partial feedback linearization with respect to the base variable q_b sometimes

induces a singularity (for example, $\theta = \pi/2$ in the cart-pole system). To deal with the singularity, several methods can be adopted. For example, by adjusting the time interval T , q_b is limited in certain range that does not contain the singular points; by dividing the desired shift of the fiber variable q_{des} into several portions, q_b can be formed by two or more connecting 2-nd derivatives of Beta-function; or, by selecting a rest-to-rest function for q_b with more than one oscillation, q_{des} can be reached with acceptable amplitudes of q_b away from the singularities.

Secondly, to apply the proposed motion planning algorithm for the base-free systems, one extra requirement exists: in Section 3.3, it is pointed out that only when (3.19) holds true, the geometric phase based algorithm works. This drawback limits the class of under-actuated systems where the proposed algorithm can be applied. One can easily prove that this requirement is fulfilled for common base-free systems such as cart-pole, ballbot and planar ballbot system. However, a generic proof for all base-free systems is expected to be developed in our future work.

Thirdly, in this chapter, we considered only rest-to-rest motion planning problem for a class of 2DOF underactuated systems. In the future work, we plan to generalize the proposed algorithm to non-rest-to-rest motion planning for n DOF base-free systems. To achieve this goal, we intend to build connections between the solved (2DOF, rest-to-rest) and currently unsolved (n DOF, non-rest-to-rest) motion planning problems. For example, the trajectories under a non-rest-to-rest motion could be viewed as a portion of the trajectories under a rest-to-rest motion.

Chapter 4

Function approximation technique based control for non-square systems

4.1 Background

The need for control of mechanical systems arises in many practical applications that include the use of ships, underwater vehicles, helicopters, aircraft, satellites, mobile robots, space platforms, flexible joint robots, hyper-redundant and snake-like manipulator, walking robots, capsule robots, and hybrid machines [2–5], and control methods have been proposed [2–5, 36–47]. The proposed methods have certain requirements on some special features of mechanical systems and often are not feasible for those systems that fail to satisfy the requirements. Instead of classifying the systems by their features and developing corresponding techniques to tackle the control problem, we intend to propose a generic method that works on a wide class of mechanical systems. This constitutes the main goal of this research.

Then the construction of the function approximation based controller is formulated by two parts. The first part is to control the auxiliary system, which is trivial. The second part is to eliminate the influence of the time varying uncertainty to the control process. We parameterize the uncertainty term with a set of chosen basis functions weighted by unknown parameters. These parameters are required to be constants for the sake of applying the

Lyapunov design. Then define update laws such that the parameters of the weighted basis functions can be automatically determined and the variation between the auxiliary square system and the original non-square system can be eliminated. By combining these two parts, we obtain a feedback controller that makes the non-square system asymptotically stable.

The rest of this chapter is organized as follows. First, in Section 4.2 we state the control problem for the non-square system and propose the control method based on the function approximation technique. Also, in this section, we establish the asymptotic stability for the closed loop system combined by the non-square system and the constructed feedback controller. The feasibility of the proposed control method is verified under simulations in Section 4.3. Finally, conclusions are drawn in Section 4.4.

4.2 Function approximation technique based control

A non-square control system can be written in the state-space form as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}, \quad (4.1)$$

where $\mathbf{x} \in \mathbf{R}^m$, $\mathbf{u} \in \mathbf{R}^n$ and $m > n$. By selecting the state $\mathbf{x}$ to be the system output, the control problem addressed in this chapter can be stated as constructing an input $\mathbf{u}$ such that $\lim_{t \rightarrow \infty} \mathbf{x} = \mathbf{0}$.

For a square system where $\mathbf{G}$ is an invertible matrix, a feedback law can be easily formulated as

$$\mathbf{u} = -\mathbf{G}^{-1}(\mathbf{f} + \mathbf{K}\mathbf{x}), \quad (4.2)$$

where $\mathbf{K}$ is a positive definite matrix. However for a non-square system where $\mathbf{G}$ is an $m \times n$ matrix, $\mathbf{G}^{-1}$ does not exist. Hence the control law (4.2) is not applicable.

4.2.1 Design of controller

To square the control system, one introduces the auxiliary input $\mathbf{u}^* \in \mathbf{R}^m$ where

$$\mathbf{u} = \mathbf{G}^* \mathbf{u}^*, \quad (4.3)$$

and the auxiliary matrix $\mathbf{G}^*$ is chosen to be a full rank $n \times m$ matrix, the weighted pseudoinverse of matrix $\mathbf{G}$, that is

$$\mathbf{G}^* = (\mathbf{G}^\top \mathbf{W} \mathbf{G})^{-1} \mathbf{G}^\top \mathbf{W}, \quad (4.4)$$

where $\mathbf{W}$ is a constant positive definite symmetric matrix. Then the non-square system (4.1) is rewritten as

$$\dot{\mathbf{x}} = \mathbf{f} + \mathbf{G} \mathbf{G}^* \mathbf{u}^*, \quad (4.5)$$

which is a square system. However as matrix $\mathbf{G} \mathbf{G}^*$ is singular and not invertible, a feedback control law for the auxiliary input $\mathbf{u}^*$ in form of (4.2) is not easily defined. For the control purpose, we restructure the system (4.5) as

$$\dot{\mathbf{x}} = \mathbf{f} + \mathbf{G} \mathbf{G}^* \mathbf{u}^* + \mathbf{u}^* - \mathbf{u}^* = \mathbf{u}^* + \mathbf{d}, \quad (4.6)$$

where

$$\mathbf{d}(\mathbf{x}, \mathbf{u}^*, t) = \mathbf{f} + (\mathbf{G} \mathbf{G}^* - \mathbf{I}) \mathbf{u}^*. \quad (4.7)$$

Note that through the above rearrangement, the original non-square system is restructured as the combination of two parts, a square system

$$\dot{\mathbf{x}} = \mathbf{u}^*, \quad (4.8)$$

referring to the auxiliary system, and $\mathbf{d}$, which can be viewed as an uncertainty term. Thus the control problem for the non-square system is reformulated to the adaptive control prob-

lem for a square system with time-varying uncertainties, which is stated as constructing $\mathbf{u}^*$ such that $\lim_{t \rightarrow \infty} \mathbf{x} = \mathbf{0}$, with $\mathbf{d}$ unknown.

Note that compared with the adaptive control problem for a system with parametric uncertainty, it is challenging for a system with time-varying uncertainty where traditional techniques such as the model reference adaptive control (MRAC) are not feasible. To tackle this problem, we adopt the function approximation technique (FAT) based adaptive control [53–56]. Compared with the traditional MRAC, the advantage of the FAT-based control is in the representation of the time-varying uncertainties by a set of given basis functions weighted by a set of unknown constant parameters. Thus the problem of eliminating the influence of the uncertainty terms is transformed to the estimation of parametric errors. Then Lyapunov designs are applied to derive proper update laws adjusting the estimates of the unknown parameters. In what follows, the construction of the controller based on the FAT approach is shown in details.

To control (4.6), the effect of the uncertainty term $\mathbf{d}$ to the system needs to be eliminated. Note that the uncertainty term varies with respect to time t and the state $\mathbf{x}$ (the input $\mathbf{u}$ can also be expressed by t and $\mathbf{x}$). However, at any moment $\mathbf{d}$ is a constant. We use weighted basis functions to approximate $\mathbf{d}$ at each moment as

$$\mathbf{d}(\mathbf{x}, t, \mathbf{u}^*) = \sum_{i=0}^N \mathbf{d}_i \psi_i(\mathbf{x}, t) + \boldsymbol{\epsilon}, \quad (4.9)$$

where $\mathbf{d}_i$, referring to the plant parameter, is constant, ψ_i , referring to the basis function, consists of $\mathbf{x}$ and t , and $\boldsymbol{\epsilon}$, referring to the approximation error, describes the deviation between the uncertainty $\mathbf{d}$ and the weighted basis functions, N is the order of the basis function. For instance, if we select the basis function as a polynomial with $N = 2$,

$$\mathbf{d} = \mathbf{d}_0 + \mathbf{d}_1 t + \mathbf{d}_2 t^2 + \boldsymbol{\epsilon}. \quad (4.10)$$

Hence by substituting (4.9) into the the state equation (4.6) yields

$$\dot{\mathbf{x}} = \mathbf{u}^* + \sum_{i=0}^N \mathbf{d}_i \psi_i(\mathbf{x}, t) + \boldsymbol{\epsilon} \quad (4.11)$$

at each moment.

The control of (4.11) requires the unknown plant parameters $\mathbf{d}_i$ to be identified. To reach this goal, these plant parameters $\mathbf{d}_i$ at each time t are estimated by $\hat{\mathbf{d}}_i(t)$, referring to the control parameter. The change of $\hat{\mathbf{d}}_i(t)$, with respect to time, is governed by an update law that we are to define in the construction of a feedback controller $\mathbf{u}^*$ from the following process.

To construct $\mathbf{u}^*$ that steers $\mathbf{x}$ to zero and an update law that defines $\hat{\mathbf{d}}_i(t)$, a feasible Lyapunov candidate function would be

$$V = \frac{1}{2} \mathbf{x}^\top \mathbf{x} + \frac{1}{2} \sum_{i=0}^N \left(\hat{\mathbf{d}}_i(t) - \mathbf{d}_i \right)^\top \left(\hat{\mathbf{d}}_i(t) - \mathbf{d}_i \right), \quad (4.12)$$

which combines both the state $\mathbf{x}$ and the estimation error $\hat{\mathbf{d}}_i(t) - \mathbf{d}_i$ between the plant parameters $\mathbf{d}_i$ and their estimates $\hat{\mathbf{d}}_i(t)$. Note that in (4.12), the parameters $\mathbf{d}_i$ are constants, which implies $\dot{\mathbf{d}}_i = \mathbf{0}$. Therefore the derivative of the Lyapunov function candidate is calculated as

$$\dot{V} = \mathbf{x}^\top \dot{\mathbf{x}} + \sum_{i=0}^N \left(\hat{\mathbf{d}}_i(t) - \mathbf{d}_i \right)^\top \dot{\hat{\mathbf{d}}}_i(t). \quad (4.13)$$

To further simplifying the above equation, substituting the vector $\dot{\mathbf{x}}$ expressed by (4.11) into (4.13) gives

$$\begin{aligned} \dot{V} &= \mathbf{x}^\top \left(\mathbf{u}^* + \sum_{i=0}^N \mathbf{d}_i \psi_i(\mathbf{x}, t) + \boldsymbol{\epsilon} \right) + \sum_{i=0}^N \left(\hat{\mathbf{d}}_i(t)^\top \dot{\hat{\mathbf{d}}}_i(t) - \mathbf{d}_i^\top \dot{\hat{\mathbf{d}}}_i(t) \right) \\ &= \mathbf{x}^\top \left(\mathbf{u}^* + \boldsymbol{\epsilon} \right) + \sum_{i=0}^N \hat{\mathbf{d}}_i(t)^\top \dot{\hat{\mathbf{d}}}_i(t) + \sum_{i=0}^N \mathbf{d}_i^\top \left(\mathbf{x} \psi_i(\mathbf{x}, t) - \dot{\hat{\mathbf{d}}}_i(t) \right). \end{aligned}$$

As the control law to be constructed cannot contain unmeasurable elements, the unknown parameters $\mathbf{d}_i$ in the derivative of the Lyapunov function need to be excluded. To cancel the

terms with $\hat{\mathbf{d}}_i$, define the update law

$$\dot{\hat{\mathbf{d}}}_i(t) = \mathbf{x}\psi_i(\mathbf{x}, t), \quad (4.14)$$

which leads to

$$\dot{V} = -\mathbf{x}^\top \left(\mathbf{u}^* + \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t) + \boldsymbol{\epsilon} \right).$$

The approximation error $\boldsymbol{\epsilon}$ needs to be considered in the construction of the auxiliary input $\mathbf{u}^*$. Select $\mathbf{u}^*$ as

$$\mathbf{u}^* = \mathbf{u}_x^* + \mathbf{u}_\epsilon^*, \quad (4.15)$$

where $\mathbf{u}_\epsilon^*$ is to cover the effect of $\boldsymbol{\epsilon}$. Thus the derivative of the Lyapunov candidate function $\dot{V}$ becomes

$$\dot{V} = -\mathbf{x}^\top \left(\mathbf{u}_x^* + \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t) \right) - \mathbf{x}^\top \left(\mathbf{u}_\epsilon^* + \boldsymbol{\epsilon} \right). \quad (4.16)$$

Constructing

$$\mathbf{u}_x^* = -\mathbf{K}\mathbf{x} - \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t), \quad (4.17)$$

where $\mathbf{K}$ is a positive matrix, yields

$$\dot{V} = -\mathbf{x}^\top \mathbf{K}\mathbf{x} - \mathbf{x}^\top \left(\mathbf{u}_\epsilon^* + \boldsymbol{\epsilon} \right). \quad (4.18)$$

Then one designs a robust control law for $\mathbf{u}_\epsilon^*$ to cover the effect of $\boldsymbol{\epsilon}$. Denote the components of $\boldsymbol{\epsilon} \in \mathbf{R}^m$ to be ϵ_j , where $j = 1, 2, \dots, m$. Suppose ϵ_j is bounded and its variation bound is available, i.e., there exists $\delta_j > 0$ such that $\|\epsilon_j\| \leq \delta_j$. Then selecting $\mathbf{u}_x^*$ the same as (4.24) and $u_{\epsilon_j}^*$ as

$$u_{\epsilon_j}^* = -\delta_j \operatorname{sgn}(x_j) \quad (4.19)$$

yields

$$\dot{V} = -\mathbf{x}^\top \mathbf{K} \mathbf{x} + \sum_{j=0}^m \left(x_j \epsilon_j - \delta_j \|x_j\| \right). \quad (4.20)$$

Note that

$$\sum_{j=0}^m \left(x_j \epsilon_j - \delta_j \|x_j\| \right) \leq 0 \quad (4.21)$$

holds true when ϵ_j is bounded by δ_j . Therefore, $\dot{V} \leq 0$ is guaranteed under the control law

$$\mathbf{u} = \mathbf{G}^*(\mathbf{u}_x^* + \mathbf{u}_\epsilon^*). \quad (4.22)$$

The convergence of the state $\mathbf{x}$ is achieved while the effect of the approximation error ϵ is covered. Thus, (4.22) together with the update law (4.14) formulates the FAT based controller for non-square systems.

Note that if the number of the basis functions ψ_i is chosen to be sufficiently large such that $\epsilon \approx 0$, the designed controller could be simplified while guaranteeing the convergence of the state $\mathbf{x}$. Specifically, by selecting $\mathbf{u}_\epsilon^* = \mathbf{0}$, the derivative of the Lyapunov function candidate (4.16) becomes

$$\dot{V} = -\mathbf{x}^\top \left(\mathbf{u}_x^* + \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t) \right). \quad (4.23)$$

Constructing the auxiliary input

$$\mathbf{u}^* = \mathbf{u}_x^* = -\mathbf{K} \mathbf{x} - \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t), \quad (4.24)$$

where $\mathbf{K}$ is a positive matrix, yields

$$\dot{V} = -\mathbf{x}^\top \mathbf{K} \mathbf{x} \leq 0, \quad (4.25)$$

which implies the convergence of $\mathbf{x}$. Hence the system input $\mathbf{u}$ can be simplified from (4.22)

as

$$\mathbf{u} = -\mathbf{G}^* \left(\mathbf{K} \mathbf{x} + \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t) \right) \quad (4.26)$$

with the update law defined by (4.14).

4.2.2 Proof of stability

To prove the asymptotic stability of the closed-loop system formulated by (4.1), (4.14), and (4.26) around the equilibrium point, we follow the Lyapunov-like analysis [97] based on the following

Lemma 4.2.1. *If a scalar function V satisfies the following conditions*

1. V is lower bounded
2. $\dot{V}$ is negative-semidefinite
3. $\dot{V}$ is uniformly continuous in time

then $\dot{V} \rightarrow 0$ as $t \rightarrow \infty$.

The Lyapunov function V expressed by (2.78) satisfies the first and second conditions that V is lower bounded and $\dot{V}$ is negative semi-definite. To follow the Lyapunov-like analysis, we need to show the uniform continuity of $\dot{V}$. This can be done by proving that $\ddot{V}$ is bounded, where

$$\ddot{V} = -2\mathbf{x}^\top \mathbf{K} \dot{\mathbf{x}}. \quad (4.27)$$

As $\dot{V} \leq 0$, V is bounded and thus so is its component $\mathbf{x}$. According to (4.25), we see that $\dot{V}$ is only composed of variables $\mathbf{x}$, $\dot{\mathbf{x}}$ is bounded, and, thus, so is its component $\dot{\mathbf{x}}$. Therefore $\ddot{V}$ is bounded since it only contains the elements $\mathbf{x}$ and $\dot{\mathbf{x}}$ which are shown above to be bounded. Hence, $\dot{V}$ is uniformly continuous. Application of Barbalat's lemma then indicates that $\mathbf{x}$ converges to $\mathbf{0}$ when $t \rightarrow \infty$.

It should be noted that $\hat{\mathbf{d}}_i(t) - \mathbf{d}_i$, the error between the parameters and their estimations, is not necessarily asymptotically stable. However, according to (4.25), we see that the derivative of the Lyapunov function of the closed loop system is negative semi-definite and $\hat{\mathbf{d}}_i(t) - \mathbf{d}_i$ is bounded, which indicates that the update laws $\hat{\mathbf{d}}_i(t)$ finally converge to certain constant values.

4.2.3 Selection of weight matrices

Note that the constructed control law (4.26) varies from a classical feedback that it synthesizes an adaptation mechanism

$$\mathbf{u}_a = -\mathbf{G}^* \sum_{i=0}^N \hat{\mathbf{d}}_i(t) \psi_i(\mathbf{x}, t), \quad (4.28)$$

which deals with the variation $\mathbf{d}$ between the non-square system (4.1) and the auxiliary square system (4.8). However the influence of the adaption mechanism $\mathbf{u}_a$ to the control process is vanishing when the coefficients in the gain matrix $\mathbf{K}$ are chosen large and the controller is approaching a pure state feedback

$$\mathbf{u} = -\mathbf{G}^* \mathbf{K} \mathbf{x}. \quad (4.29)$$

This feature is significant when applying this function approximation based algorithm on the nonholonomic systems since according to well-known Brockett's condition [57], there exists no state feedback for nonholonomic systems that achieves asymptotic stability. Therefore for the control of this specific type of systems, the following points need to be aware of.

First the gain coefficients $k_1, k_2, \dots, k_m$ corresponding to the state variables $x_1, x_2, \dots, x_m$ are required to be small to weaken the influence of non-adaptive portion (4.29) in the controller. However, the small value of feedback coefficients leads to the low converging speed for the system output. This problem can be solved by assigning positive functions to the feedback coefficients $k_1, k_2, \dots, k_m$, which start from small values but keep growing with respect to time, instead of using positive constants.

Second, the initial value of the control parameters $\hat{\mathbf{d}}_i(t)$ cannot be chosen zero. If so,

the controller (4.26) yields to a state feedback in from of (4.29) at the instant $t = 0$ and the nonholonomic system is not controllable via (4.29). The closer the initial values of the state variables to the equilibrium are given, the larger values of $\hat{\mathbf{d}}_i(0)$ are required to assign to weight the adaptation mechanism $\mathbf{u}_a$. More details about the control for nonholonomic systems will be discussed in Section 4.3 via an example of the unicycle system.

4.3 Case study

Here we test the control law for three cases, a linear system (double integrator system), a nonlinear underactuated system (cart-pole system), and a nonholonomic system (unicycle system). For the linear and the nonlinear systems, we choose a set of polynomial basis functions for estimating the variation $\mathbf{d}$ expressed by (4.7), that is,

$$\psi_i(\mathbf{x}, t) = t^i. \quad (4.30)$$

Thus the variation $\mathbf{d}$ is estimated as

$$\mathbf{d} \approx \sum_{i=0}^N \mathbf{d}_i t^i. \quad (4.31)$$

Note that the estimation of $\mathbf{d}$, formulated by (4.31), may diverge because the polynomials are proportional to time t . For finite t however, this does not represent a problem — the output signal will still converge as shown in Section 4.2.2, even if the estimation (4.31) does not approach to $\mathbf{d}$. For the nonholonomic system, we choose the basis function to be the step function.

The feedback controller and the update law corresponding to (4.31) are defined as

$$\mathbf{u} = -\mathbf{G}^* \left(\mathbf{K} \mathbf{x} + \sum_{i=0}^N \hat{\mathbf{d}}_i(t) t^i \right), \quad \dot{\hat{\mathbf{d}}}_i(t) = \mathbf{x} t^i. \quad (4.32)$$

In the linear and nonlinear cases, we choose the second order polynomial functions to estimate the system variation $\mathbf{d}$, namely, $N = 2$. In the nonholonomic case, $N = 4$.

Linear system

Let us start from a linear example, the double integrator system. The control problem is stated as given a double integrator system

$$\ddot{x} = u, \quad (4.33)$$

construct u such that x converges to zero. Select the state vector to be

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad (4.34)$$

where $x_1 = x$, $x_2 = \dot{x}$, and rewrite the system equations (4.33) in the state space form as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \quad (4.35)$$

The control problem is then redescribed as constructing an input u such that $\lim_{t \rightarrow \infty} \mathbf{x} = \mathbf{0}$.

Note that a linear system is a special case of nonlinear system where

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}, \quad \mathbf{G}(\mathbf{x}) = \mathbf{B}. \quad (4.36)$$

Then to apply the constructed control law (4.32), we need to select a suitable matrix $\mathbf{B}^*$. As $\mathbf{B}^*$ cannot contain zero columns, it is chosen to be a weighted pseudoinverse of matrix $\mathbf{B}$, calculated as

$$\mathbf{B}^* = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

To know whether the selected $\mathbf{B}^*$ is proper, let us check the controllability for the reformu-

lated system

$$\dot{x} = Ax + \bar{B}u^*, \quad \bar{B} = BB^* = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \quad (4.37)$$

after introducing the auxiliary input $u = B^*u^*$. For the new system (4.37) with auxiliary input u^* , the controllability is easily proved. As system (4.37) is controllable, the controller (4.32) is applicable and simulation is conducted. In the simulation, the initial values of the control parameters $\hat{d}_i(t)$ are chosen to be zero. The gain matrix is selected positive definite, as

$$K = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix},$$

and the initial conditions of the system output x_1 and x_2 are specified as 2 m and 1 m/s.

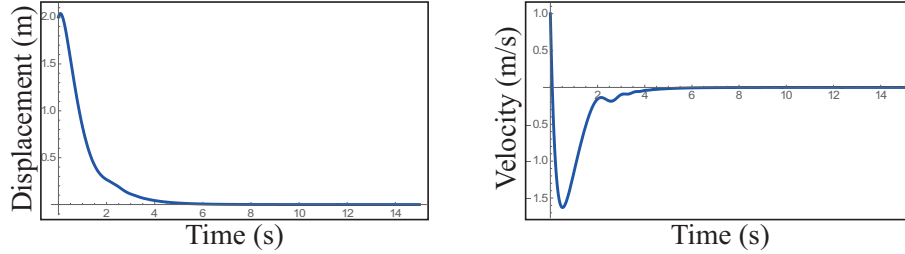


Figure 4.1: System response of double integrator under constructed controller.

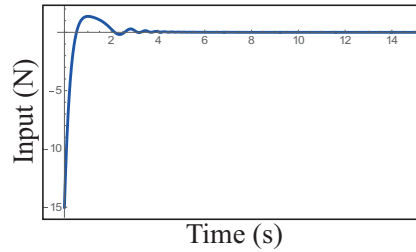


Figure 4.2: Input signal.

Under the constructed control law, the system responses x_1 and x_2 converge to desired

values, as illustrated in Fig. 4.1, where the left and right graphs stand for the trajectories for x_1 and x_2 respectively. The input signal is shown by Fig. 4.2.

Nonlinear system

Then let us test the controller for a nonlinear example, the well-known cart-pole system. The dynamic equations (3.25) govern the relation between the input torque and corresponding motion of the cart-pole system. The constructed control law (4.32) is tested under the above dynamic model. To apply the control algorithm, we choose the state vector and the system input to be

$$\mathbf{x} = \begin{bmatrix} s \\ \dot{s} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad u = f, \quad (4.38)$$

and rewrite the dynamic equations of the cart-pole system (3.25) in the state space form as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})u \\ &= \begin{bmatrix} x_2 \\ \frac{H_{cp}h_p - H_{pp}h_c}{H_{cc}H_{pp} - H_{cp}H_{pc}} \\ x_4 \\ \frac{H_{pc}h_c - H_{cc}h_p}{H_{cc}H_{pp} - H_{cp}H_{pc}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{H_{pp}}{H_{cc}H_{pp} - H_{cp}H_{pc}} \\ 0 \\ \frac{H_{pc}}{H_{cc}H_{pp} - H_{cp}H_{pc}} \end{bmatrix} u. \end{aligned}$$

The control problem is then defined as constructing an input u such that $\lim_{t \rightarrow \infty} \mathbf{x} = \mathbf{0}$.

To apply the constructed control law (4.32), we select the weight matrix $\mathbf{W}$ to be a

symmetric positive definite matrix as

$$\mathbf{W} = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 2 & 5 & 0 & 0 \\ 0 & 0 & 15 & 5 \\ 0 & 0 & 5 & 2 \end{bmatrix}.$$

Note that the off-diagonal elements of $\mathbf{W}$ affect the converging speed and the pattern of convergence (less oscillatory) of the state $\mathbf{x}$, while the diagonal elements were adjusted to keep the weight matrix $\mathbf{W}$ positive definite. Thus, $\mathbf{G}^*$ is calculated as

$$\mathbf{G}^* = \begin{bmatrix} \frac{2H_{pp}(H_{cc}H_{pp}-H_{cp}H_{pc})}{2H_{cc}^2+5H_{cp}^2} \\ \frac{5H_{pp}(H_{cc}H_{pp}-H_{cp}H_{pc})}{2H_{cc}^2+5H_{cp}^2} \\ \frac{5H_{pc}(H_{cc}H_{pp}-H_{cp}H_{pc})}{2H_{cc}^2+5H_{cp}^2} \\ \frac{2H_{pc}(H_{cc}H_{pp}-H_{cp}H_{pc})}{2H_{cc}^2+5H_{cp}^2} \end{bmatrix}^\top.$$

To know whether the selected $\mathbf{G}^*$ is proper, the controllability for the reformulated system

$$\dot{\mathbf{x}} = \mathbf{f} + \mathbf{G}\mathbf{G}^*\mathbf{u}^* = \mathbf{f} + \mathbf{g}_1u_1 + \mathbf{g}_2u_2 + \mathbf{g}_3u_3 + \mathbf{g}_4u_4 \quad (4.39)$$

after introducing the auxiliary input $u = \mathbf{G}^*\mathbf{u}^*$ is checked in Appendix A.

As (4.39) is locally controllable, the control law (4.32) is applicable and a simulation is conducted. In the simulation, the gain matrix is selected as

$$\mathbf{K} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix},$$

and the initial conditions of the system output x_1 , x_2 , x_3 , and x_4 are specified as 2, 0, 0, and 0. The numerical values used in the simulation are $m = 1$ kg, $m_p = 1$ kg, $l = 0.1$ m, $g = 9.8$ m/s², and $T = 15$ s. The initial state is (2, 0, 0, 0) and the initial values of the control parameters $\hat{d}_i(t)$ are chosen to be zero.

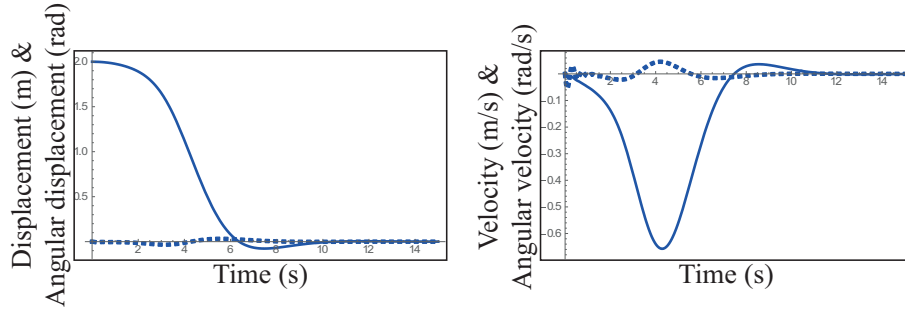


Figure 4.3: Nonlinear system response under constructed controller.

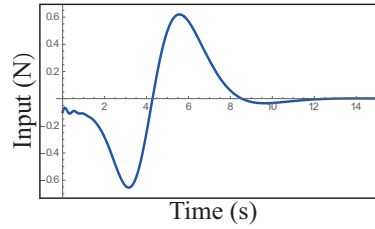


Figure 4.4: Input signal.

Under the constructed controller, the system outputs x_1 , x_2 , x_3 , and x_4 converge to zero, as illustrated in Fig. 4.3. The solid curves stand for the trajectories of the displacement x_1 and velocity x_2 of the cart, whereas the dashed curves for the angular displacement x_3 and angular velocity x_4 of the pendulum. The input signal is shown by Fig. 4.4.

Nonholonomic system

Let us check the validity of the constructed controller (4.32) applied on a nonholonomic system, the unicycle system. A unicycle is a vehicle with a single orientable wheel. Its

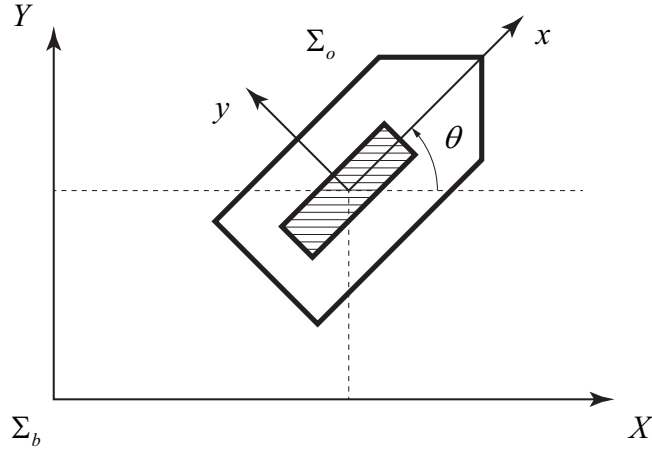


Figure 4.5: Unicycle system.

configuration is described by

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad (4.40)$$

where x and y are the Cartesian coordinates of the contact point of the wheel with the ground, and the angle θ is the orientation of the wheel with respect to the x -axis (see Fig. 4.5). The pure rolling constraint for the wheel, which is nonholonomic, is expressed as

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0, \quad (4.41)$$

implying that the velocity of the contact point is zero in the direction orthogonal to the local y -axis of the vehicle. From the nonholonomic constraint, the kinematic model of the

unicycle is then derived as

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (4.42)$$

where the inputs v and ω have a clear physical interpretation. In particular, v is the linear velocity of the vehicle whereas the steering velocity ω is the wheel angular speed around the vertical axis. To apply the control algorithm, we choose the state vector and the system input to be

$$\mathbf{x} = \mathbf{q}, \quad \mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix},$$

and rewrite the system equations in state space form as

$$\dot{\mathbf{x}} = \mathbf{G}(\mathbf{x})\mathbf{u} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (4.43)$$

The control problem is then described as constructing an input $\mathbf{u}$ such that $\lim_{t \rightarrow \infty} \mathbf{x} = \mathbf{0}$. For the construction of input $\mathbf{u}$, let us introduce the auxiliary input $\mathbf{u}^*$, where

$$\mathbf{u} = \mathbf{G}^* \mathbf{u}^*, \quad (4.44)$$

By selecting

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

one calculates matrix $\mathbf{G}^*$ as

$$\mathbf{G}^* = \mathbf{G}^\top = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and the state equation is rewritten as

$$\dot{\mathbf{x}} = \mathbf{G}\mathbf{G}^\top \mathbf{u}^* = \begin{bmatrix} \cos^2 \theta & \cos \theta \sin \theta & 0 \\ \cos \theta \sin \theta & \sin^2 \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u}^*. \quad (4.45)$$

To test the validity of the chosen $\mathbf{G}^*$ matrix, the local controllability around the equilibrium point $\mathbf{x} = (0, 0, 0)$ for the following system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{g}_1 u_1 + \mathbf{g}_2 u_2 + \mathbf{g}_3 u_3 \\ &= \begin{bmatrix} \cos^2 \theta \\ \cos \theta \sin \theta \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} \cos \theta \sin \theta \\ \sin^2 \theta \\ 0 \end{bmatrix} u_2 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_3 \end{aligned}$$

is checked in Appendix B.

Knowing that the unicycle system is locally controllable, we apply the constructed control law (4.32), where the gain matrix is given as

$$\mathbf{K} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & k_\theta \end{bmatrix}.$$

The system responses x_1 , x_2 , and x_3 converge to desired values, as illustrated in Fig. 4.6, where the solid and dashed curves in the left picture stand for the trajectories displacement of the vehicle x_1 and x_2 respectively, while the solid curve in the right picture for the orientation

x_3 of the system. The initial conditions of the system output x_1 , x_2 , and x_3 are specified as 5 m, 5 m, and π rad. The initial values of the control parameters $\hat{\mathbf{d}}_i(t)$ are chosen to be 1. The input signals are shown by Fig. 4.7, where the linear and angular velocity inputs are respectively represented by solid and dashed curves.

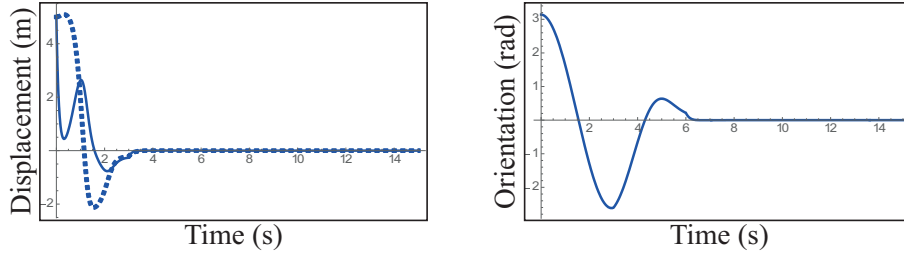


Figure 4.6: Nonlinear system response under constructed controller.

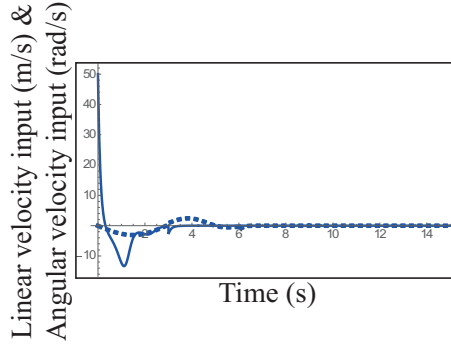


Figure 4.7: Input signal.

As introduced in Section 4.2, there are two essential points to be noted for the control of the nonholonomic unicycle system. First the gain coefficient k_θ corresponding to the orientational angle θ of the unicycle, which is the only state variable appears in the control matrix $\mathbf{G}$, is required to be small. However, the small value of k_θ leads to the low converging speed for the system output θ . To address this problem, in the simulation k_θ is selected to be $\frac{e^{(t+0.5)}}{50}$, which keeps increasing with respect to time.

Second, the initial value of the control parameters $\hat{\mathbf{d}}_i(t)$ cannot be chosen small. The reason is formulated as follows. Although the restructured system (4.45) with auxiliary

input $\mathbf{u}^*$ is proved to be controllable, its linearized system near the equilibria

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u}^* \quad (4.46)$$

is not controllable. At the equilibrium, the selection of the initial value for the control parameters $\hat{\mathbf{d}}_i(0) = 0$ leads to a condition that the auxiliary input

$$\mathbf{u}^* = -\mathbf{K}\mathbf{x} \quad (4.47)$$

becomes a state feedback law. According to well-known Brockett's condition [57], for those systems, of which the linearized system is not controllable, there exists no pure state feedback. Therefore, for the nonholonomic systems, the initial value of the control parameters cannot be chosen as zero. Moreover, to enhance the influence of the adaptation term $\mathbf{u}_a$ in form of (4.28) to the control process, the smaller the initial values of the state variables are given, the larger values of $\hat{\mathbf{d}}_i(0)$ are required to assign in order to weight the adaptation mechanism $\mathbf{u}_a$.

4.4 Summary

A generic control method has been proposed for the non-square systems where the number of system inputs is not equal to that of the states. By introducing an auxiliary input the dimension of which equals that of the system state, the non-square system was restructured in the form of the combination of a square system and the variation from the original non-square system, which is treated as system uncertainties.

In the control process, the uncertainty term was replaced by its approximation as a chosen basis function weighted by constant parameters to be determined. These unknown plant parameters are estimated at each instant, denoted by the adjustable control parameters using a defined update law. Thus the influence to the control process caused by the variation

between the auxiliary square system and the original non-square system can be eliminated.

The asymptotic stability was established for the closed loop system formulated by the non-square system and the constructed controller. The feasibility of the proposed control method was verified under simulations for the point stabilization problem of a linear system, a nonlinear underactuated system, and a nonholonomic system, respectively.

The following issues need to be clarified in the future research. Firstly, the relation between the actual input $\mathbf{u}$ and the auxiliary input $\mathbf{u}^*$ is expressed by (4.3). Note that in (4.3) there exists a matrix $\mathbf{G}^*$ which was assigned to be a weighted pseudoinverse of $\mathbf{G}$ in this chapter. However, the features of a good candidate for matrix $\mathbf{G}^*$ are not specifically analyzed and a more systematic way of selecting $\mathbf{G}^*$ needs to be developed.

Secondly, in this chapter, we selected various types of the basis functions for estimating the variation term $\mathbf{d}$ of the auxiliary square system from the original non-square system. However, the advantages and disadvantages of these selections are not analyzed and more candidate functions need to be investigated in the future study.

Thirdly, the proposed controller is not only composed of an adaptive portion, but also a robust portion, which features discontinuities. The robust portion is designed for the estimation error ϵ of the basis functions. When ϵ is small, the robust portion has little effect to the control system. Moreover, the discontinuities in the control input, after integration, would not appear in the output signal. In case the effect of the discontinuities is remarkable, the signum function in the controller can be replaced by the saturation function.

Chapter 5

Concluding remarks

The motion planning and control problems for underactuated systems have been addressed in this thesis. To demonstrate the difficulties in solving these problems, this thesis started with a specific example, the ball-pendulum system, a class of spherical rolling robots actuated by a pendulum with two degrees of mobility. One of the most critical research problems for control of the mobile robot is to steer the robot from an initial configuration to the final configuration. For the ball-pendulum system, this reconfiguration problem was first divided into two stages, motion planning, and trajectory tracking, and then solved separately. Next, I extended the motion planning and control methods developed for the ball-pendulum system to more generic cases. Specifically, a geometric phase based approach and a function approximation technique based control method are proposed respectively for the motion planning and control problems for a class of underactuated systems.

Ball-pendulum system

Motion planning The motion planning problem for the ball-pendulum system features three factors: nonholonomic rolling constraints, underactuated dynamics, and the gravitational drift. The combination of these factors makes this problem unique and very challenging. Conventional decoupling to the levels of kinematics and dynamics is not always possible because kinematically admissible trajectories can be dynamically unrealizable. The novelty of my research is that I established the dynamic realizability conditions for the ball-

pendulum system. Then, assuming that the contact path is specified and the sphere moves in a pure rolling mode which is realizable, the full dynamic model was reduced by imposing corresponding virtual constraints. A timing control law was constructed, based on the geometric phase approach, for tracing the contact curves in rest-to-rest motion.

Trajectory tracking The trajectory tracking problem is challenging for the ball-pendulum because the system is non-square where the number of system inputs is not equal to that of the states. In my research, the non-square system (the ball-pendulum system) was first restructured in the form of the combination of a square system and the variation from the original non-square system. Then, a function approximation technique (FAT) based adaptive controller was proposed, under which the system output converged to the desired trajectory, with the effect of the variation term eliminated. The proposed controller is adaptive to time-varying uncertainties including un-modeled dynamics and external disturbance and this feature was tested under simulations.

Geometric phase based motion planning

Based on my study of motion planning for the ball-pendulum system, I extended the problem to dealing with rest-to-rest motion planning problem for a class of planar underactuated systems. We classified the planar underactuated systems into two types. One is the base-fixed systems that one or more portions of the systems are fixed on the ground, including the serial robot with passive joints. The other is the base-free type whose position and orientation can be steered freely by internal shape change, such as the cart-pole system, the planar ballbot system and the hoop-pendulum system. In this paper, we focused on the base-free type systems.

After the partial feedback linearization, this type of systems are commonly partially differentially flat and the base variable is the flat output. It leads to a property that the derivatives of the fiber variable can be represented purely by the base variable and its derivatives. Based on this property a motion planning algorithm has been constructed as follows.

First, the 2-nd order derivative of Beta-function weighted by a constant parameter is selected as a candidate for the base variable due to that it is shown to be the optimal solution

for the linearized system. Its feasibility is proved by showing that the corresponding fiber variable can satisfy the given boundary conditions. Finally the parameter of the 2-nd order derivative of Beta-function is adjusted such that a desired shift for the fiber variable can be obtained. Under the proposed motion planning algorithm, rest-to-rest trajectories for the configuration variables are generated with similar system performance but with less computational cost, compared with the optimal control formulation.

There are several points to be improved for the proposed algorithm in the future work. First, the partial feedback linearization with respect to the base variable q_b sometimes induces a singularity (for example, $\theta = \pi/2$ in the cart-pole system). To deal with the singularity, several methods can be adopted. For example, by adjusting the time interval T , q_b is limited in certain range that does not contain the singular points; by dividing the desired shift of the fiber variable q_{des} into several portions, q_b can be formed by two or more connecting 2-nd derivatives of Beta-function; or, by selecting a rest-to-rest function for q_b with more than one oscillation, q_{des} can be reached with acceptable amplitudes of q_b away from the singularities.

Second, to apply the proposed motion planning algorithm for the base-free systems, one extra requirement exists: only when (3.19) holds true, the geometric phase based algorithm works. This drawback limits the class of underactuated systems where the proposed algorithm can be applied. One can easily prove that this requirement is fulfilled for common base-free systems such as cart-pole, hoop-pendulum and planar ballbot system. However, a generic proof for all base-free systems is expected to be developed in our future work.

Third, we considered only rest-to-rest motion planning problem for a class of 2DOF underactuated systems. In the future work, we plan to generalize the proposed algorithm to non-rest-to-rest motion planning for n DOF base-free systems. To achieve this goal, we intend to build connections between the solved (2DOF, rest-to-rest) and currently unsolved (n DOF, non-rest-to-rest) motion planning problems. For example, the trajectories under a non-rest-to-rest motion could be viewed as a portion of the trajectories under a rest-to-rest motion.

Also, it is worth to mention that one interesting phenomenon was discovered from simulation, that the derivative of Beta-function is also a feasible solution to the motion planning

problem for the acrobot, a base-fixed system. However, this feasibility is yet unable to be proved systematically. It could be a breaking point of our future work on the construction of generic motion planning techniques for a wider range of underactuated systems.

Function approximation technique based control

A generic control method has been proposed for the non-square systems where the number of system inputs is not equal to that of the states. By introducing an auxiliary input the dimension of which equals that of the system state, the non-square system was restructured in the form of the combination of a square system and the variation from the original non-square system, which is treated as system uncertainties.

In the control process, the uncertainty term was replaced by its approximation as a chosen basis function weighted by constant parameters to be determined. These unknown plant parameters are estimated at each instant, denoted by the adjustable control parameters using a defined update law. Thus the influence to the control process caused by the variation between the auxiliary square system and the original non-square system can be eliminated.

The asymptotic stability was established for the closed loop system formulated by the non-square system and the constructed controller. The feasibility of the proposed control method was verified under simulations for the point stabilization problem of a linear system, a nonlinear underactuated system, and a nonholonomic system, respectively.

The following issues need to be clarified in the future research. First, the relation between the actual input u and the auxiliary input u^* is expressed by (4.3). Note that in (4.3) there exists a matrix G^* which was assigned to be a weighted pseudoinverse of G in this paper. However, the features of a good candidate for matrix G^* are not specifically analyzed and a more systematic way of selecting G^* needs to be developed.

Second, in this paper, we chose polynomials to be the basis functions for estimating the variation term d of the auxiliary square system from the original non-square system. However, the advantages and disadvantages of polynomials as basis function are not analyzed and more candidate functions need to be investigated in the future study.

Future work

Neural network aided function approximation technique based control The function approximation technique based controller constructed in my previous research is rather generic (works on a wide range of systems) and adaptive (requires very little knowledge of system dynamics). Based on these features of the FAT approach, I plan to tackle a challenging problem: designing robot controllers by an AI system. This goal would be pursued with the following three steps. Firstly, I shall train the AI to distinguish the links and joints of which most robots are composed. This step consists of research on computer vision and machine learning. Secondly, by counting the number of joints and links, the AI would know the dimensions of inputs and outputs for the robotic system (typically, joints for inputs and links for outputs) which are the only knowledge that the FAT approach requires to design an adaptive controller. Thirdly, the coefficients in the FAT based controller, which affect the converging speed and the pattern of convergence of the output signal, would be assigned by a neural network based optimization algorithm, instead of by tuning as I did in my previous research.

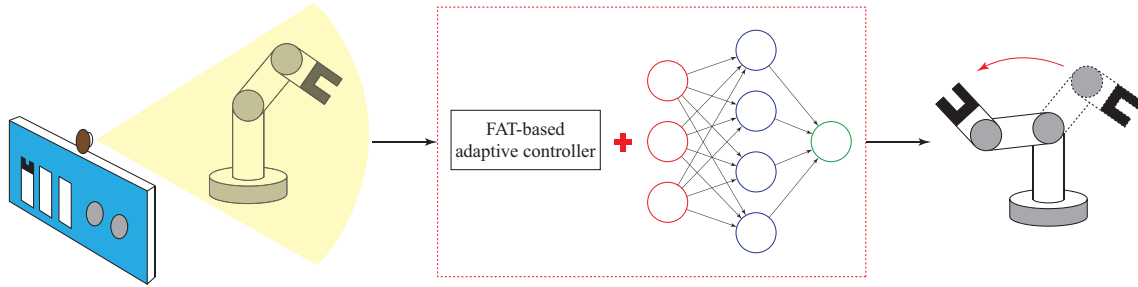


Figure 5.1: A process of the neural network aided FAT approach to controller design for robots.

Multi-agent SLAM based indoor mapping Indoor mapping is a crucial problem for the study of autonomous robots especially in its application of searching and rescuing survivals from, e.g., an earthquake-damaged building. Instead of single-robot SLAM which is widely used and well developed, I plan to address this problem with multi-agent SLAM algorithm that has the following advantages. First, cooperation between robots will be more effective

in the exploration of the working environment. Second, multi-agent SLAM is more robust: one broken robot would not affect the functioning of the whole system.

To construct a multi-agent SLAM algorithm, I would first study the control problem for a swarm of robots. In literature, classical control methods proposed for this problem commonly rely on many assumptions that, in swarm robotics, are often violated because of noise, asynchronicity, and stochasticity. Instead of classical methods, I plan to follow a data-driven reinforcement learning approach to solving cooperative multi-agent tasks. Then, an extension from single to multi-agent SLAM could be made based on the developed swarm control method. Note that spherical robot could be used in SLAM algorithm as an agent due to its omnidirectional mobility (optimal path can be planned to avoid gathering redundant data) and robustness (all the devices, e.g., camera, driving, and computing units, are covered by an outer solid spherical shell).

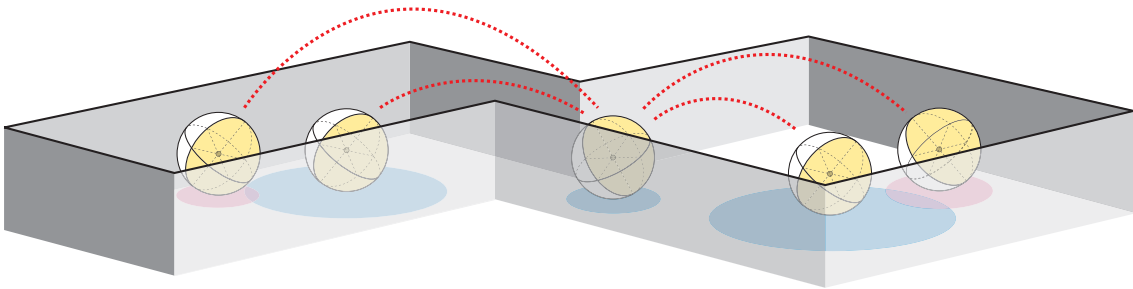


Figure 5.2: Swarm control of spherical robots for mapping of an indoor environment.

References

- [1] S. Kelly and R. Murray, “Geometric phases and robotic locomotion,” *Journal of Robotic Systems*, vol. 12, no. 6, pp. 417–431, 1995.
- [2] R. Lozano and I. Fantoni, *Non-Linear Control for Underactuated Mechanical Systems*. Springer, 2001.
- [3] A. Bloch, D. Chang, N. Leonard, and J. Marsden, “Controlled lagrangians and the stabilization of mechanical systems II: Potential shaping,” *IEEE Transactions on Robotics and Automation*, vol. 46, no. 10, pp. 1556–1571, 2001.
- [4] N. P. I. Aneke, *Control of underactuated mechanical systems*. TU Eindhoven, Mechanical Engineering: Springer Verlag, 2003.
- [5] D. Chang, “On the method of interconnection and damping assignment and passivity-based control for the stabilization of mechanical systems,” *Regular and Chaotic Dynamics*, vol. 19, no. 5, pp. 556–575, 2014.
- [6] A. Halme, T. Schonberg, and Y. Wang, “Motion control of a spherical mobile robot,” in *Proc. of 4th Int. Workshop on Advanced Motion Control*, Tsu-City, Japan, March 1996, pp. 259–264.
- [7] Y. Ming, D. Zongquan, Y. Xinyi, and Y. Weizhen, “Introducing hit spherical robot: Dynamic modeling and analysis based on decoupled subsystem,” in *2006 IEEE International Conference on Robotics and Biomimetics*, Kuming, China, Dec 2006, pp. 181–186.

- [8] Q. Zhan, Y. Cai, and C. Yan, "Design, analysis and experiments of an omni-directional spherical robot." in *Proc. of IEEE Int. Conference on Robotics and Automation*, Shanghai, May 2011, pp. 4921–4926.
- [9] D. Liu, H. Sun, T. A., Q. Jia, and L. Wang, "Motion control of a spherical mobile robot by feedback linearization," in *Proc. of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China, June 25–27 2008, pp. 965–970.
- [10] M. Ishikawa, Y. Kobayashi, R. Kitayoshi, and T. Sugie, "The surface walker: a hemispherical mobile robot with rolling contact constraints," in *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS'2009*, St. Louis, USA, October 10–15 2009, pp. 2446–2451.
- [11] B. Zhao, M. Li, H. Yu, H. Hu, and L. Sun, "Dynamics and motion control of a two pendulums driven spherical robot," in *Proc. IEEE Int. Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010, pp. 147–153.
- [12] T. Urakubo, S. Maekawa, H. Tamaki, Y. Tada, T. Otani, and H. Kobayashi, "Dynamics and control of a spherical rolling robot equipped with a gyro," in *Proc. Int. Symposium on Flexible Automation*, Osaka, Japan, July 10–12 2006, pp. 1013–1016.
- [13] V. Joshi, R. Banavar, and R. Hippalgaonkar, "Design and analysis of a spherical mobile robot," *Mechanism and Machine Theory*, vol. 45, no. 2, pp. 130–136, February 2010.
- [14] T. Fukui, A. Morinaga, M. Svinin, and M. Yamamoto, "Motion experiments with a spherical rolling robot actuated by internal rotors," in *Proc. Annual RSJ/JSME Conference ROBOMECH'13*, no. 2A2–P09, Tsukuba, Japan, May 23–24 2013, (In Japanese).
- [15] K. Wait, P. Jackson, and L. Smoot, "Self locomotion of a spherical rolling robot using a novel deformable pneumatic method," in *Proc. IEEE Int. Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 3–7 2010, pp. 3757–3762.
- [16] Y. Kim, S. Ahn, and Y. Lee, "Kisbot: New spherical robot with arms," in *Proc. of 10th WSEAS Int. Conference on Robotics, Control and Manufacturing Technology*, Hangzhou, China, April 2010, pp. 63–67.

- [17] M. Artusi, M. Potz, J. Aristizabal, C. Menon, S. Cocuzza, and S. Debei, “Electroactive elastomeric actuators for the implementation of a deformable spherical rover,” *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 1, pp. 50–57, February 2011.
- [18] C. Romero-Melendez and F. Monroy-Perez, “The motion planning problem: differential flatness and nilpotent approximation,” *Cybernetics and Physics*, vol. 2, no. 3, pp. 133–142, 2013.
- [19] F. Bullo, N. Leonard, and A. Lewis, “Controllability and motion algorithms for underactuated lagrangian systems on lie groups,” *IEEE Transactions on Automatic Control*, vol. 45, no. 8, pp. 1437–1454, August 2000.
- [20] S. Martinez, J. Cortes, and F. Bullo, “Analysis and design of oscillatory control systems,” *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1164–1177, July 2003.
- [21] F. Bullo and K. Lynch, “Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 402–412, August 2001.
- [22] Z. Neusser and M. Valasek, “Control of the underactuated mechanical systems using natural motion,” *Kybernetika*, vol. 48, no. 2, pp. 223–241, 2012.
- [23] R. Seifried, “Two approaches for feedforward control and optimal design of underactuated multibody systems,” *Multibody System Dynamics*, vol. 27, no. 1, pp. 75–93, 2012.
- [24] M. Fliess, J. Levine, and P. Rouchon, “Flatness and defect of nonlinear systems: Introductory theory and examples,” *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.
- [25] M. Rathinam and R. M. Murray, “Configuration flatness of lagrangian systems underactuated by one control,” *SIAM Journal of Control and Optimization*, vol. 36, no. 1, pp. 164–179, 1998.

- [26] J. Imura, K. Kobayashi, and T. Yoshikawa, “Nonholonomic control of a three-link planar manipulator with a free joint,” in *Proc. IEEE Int. Conference on Decision and Control*, Kobe, Japan, December 13–13 1996, pp. 1435–1436.
- [27] Y. Nakamura, T. Suzuki, and M. Koinuma, “Nonholonomic control of a three-dof planar underactuated manipulator,” *IEEE Transactions Robotics and Automation*, vol. 13, no. 6, pp. 853–862, 1997.
- [28] H. Arai, K. Tanie, and N. Shiroma, “Nonholonomic control of a three-dof planar underactuated manipulator,” *IEEE Transactions Robotics and Automation*, vol. 14, no. 5, pp. 681–695, 1998.
- [29] T. Yoshikawa, K. Kobayashi, and T. Watanabe, “Design of a desirable trajectory and convergent control for 3-dof manipulator with a nonholonomic constraint,” in *Proc. IEEE Int. Conference on Robotics and Automation*, San Francisco, CA, USA, April 24–28 2000, pp. 1805–1810.
- [30] A. De Luca, J. Levine, and G. Oriolo, “Trajectory planning and control for planar robots with passive last joint,” *International Journal of Robotics Research*, vol. 21, no. 5-6, 2002.
- [31] T. Lauwers, G. Kantor, and R. Hollis, “A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive,” in *Proc. IEEE Int. Conference on Robotics and Automation*, Orlando, FL, USA, May 15–19 2006, pp. 2884–2889.
- [32] Y. Bai, M. Svinin, and M. Yamamoto, “Motion planning for a hoop-pendulum type of underactuated systems,” in *Proc. IEEE Int. Conference on Robotics and Automation*, Stockholm, Sweden, May 16–21 2016, pp. 2739–2744.
- [33] U. Nagarajan and R. Hollis, “Shape space planner for shape accelerated balancing mobile robots,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1323–1341, 2013.

- [34] S. Ramasamy, G. Wu, and K. Sreenath, “Dynamically feasible motion planning through partial differential flatness,” in *Proc. of Robotics: Science and Systems*, Berkeley, USA, 2014, pp. 125–130.
- [35] E. Shammass, “Generalized motion planning for underactuated mechanical systems,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania, 3 2006.
- [36] A. Shiriaev, H. Ludvigsen, O. Egeland, and A. Pogromsky, “On global properties of passivity based control of the inverted pendulum,” in *Proc. 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, USA, December 7–10 1999, pp. 2513–2518.
- [37] X. Xin and M. Kaneda, “Analysis of the energy-based control for swinging up two pendulums,” *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 679–684, May 2005.
- [38] B. Gao, X. Zhang, H. Chen, and J. Zhao, “Energy-based bontrol design of an underactuated 2-dimensional tora system,” in *Proc. IEEE Int. Conference on Intelligent Robots and Systems*, St. Louis, USA, October 10–15 2009, pp. 1296–1301.
- [39] M. W. Spong, “Partial feedback linearization of underactuated mechanical systems,” in *Proc. IEEE Int. Conference on Intelligent Robots and Systems*, Munich, Germany, September 12–16 1994, pp. 314–321.
- [40] K. Pettersen and H. Nijmeijer, “Tracking control of an underactuated surface vessel,” in *Proc. 37th IEEE Conference on Decision and Control*, Tampa, Florida, USA, December 16–18 1998, pp. 4561–4566.
- [41] R. Olfati-Saber, “Trajectory tracking for a flexible one-link robot using a nonlinear noncollocated output,” in *Proc. 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 12–15 2000, pp. 4024–4029.
- [42] D. Qian, J. Yi, and D. Zhao, “Robust control using sliding mode for a class of underactuated systems with mismatched uncertainties,” in *Proc. of IEEE Int. Conference on Robotics and Automation*, Rome, Italy, April 10–14 2007, pp. 1449–1454.

- [43] R. Xu and U. Özgüner, “Sliding mode control of a class of underactuated systems,” *Automatica*, vol. 44, no. 1, pp. 233–241, January 2008.
- [44] V. Sankaranarayanan and A. Mahindrakar, “Control of a class of underactuated mechanical systems using sliding modes,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 459–467, February 2009.
- [45] J. Huang, Z. Guan, T. Matsuno, T. Fukuda, and S. K., “Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 750–758, August 2010.
- [46] S. Nersesov, H. Ashrafiuon, and P. Ghorbanian, “On the stability of sliding mode control for a class of underactuated nonlinear systems,” in *Proc. the American Control Conference*, Baltimore, MD, USA, June 30–July 2 2010, pp. 3446–3451.
- [47] L. McNinch and H. Ashrafiuon, “Predictive and sliding mode cascade control for unmanned surface vessels,” in *Proc. the American Control Conference*, San Francisco, CA, USA, June 29–July 01 2011, pp. 184–189.
- [48] Y. Liu and H. Yu, “A survey of underactuated mechanical systems,” *IET Control Theory and Appl.*, vol. 7, no. 7, pp. 921–935, 2013.
- [49] S. Li and J. Zhang, “Some properties of generalized inverse of non-square systems,” in *Proc. 52nd IEEE Conference on Decision and Control 2013*, Florence, Italy, December 10–13 2013, pp. 6446–6451.
- [50] S. Aloui, O. Pages, A. Hajjaji, A. Chaari, and Y. Koubaa, “Robust fuzzy tracking control for a class of perturbed non square nonlinear systems,” in *Proc. the American Control Conference*, Baltimore, MD, USA, June 30–July 2 2010, pp. 4788–4793.
- [51] Y. Son and J. Seo, “Passivity-based robust control of non-square linear systems using a parallel feedforward compensator,” in *Proc. the American Control Conference*, Anchorage, AK, USA, May 8–10 2002, pp. 3902–3907.

- [52] M. Demetriou, “Adaptive observers for non-square positive real infinite dimensional systems,” in *Proc. the American Control Conference*, Boston, MA, USA, July 6–8 2016, pp. 3441–3448.
- [53] F. Tyan and S. Lee, “An adaptive control for rotating stall and surge of jet engines - a function approximation approach,” in *Proc. 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, December 12–15 2005, pp. 5498–5503.
- [54] M. Chien and A. Huang, “Adaptive control for flexible-joint electrically driven robot with time-varying uncertainties,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 1032–1038, April 2007.
- [55] Y. Tsai and A. Huang, “Fat-based adaptive control for pneumatic servo systems with mismatched uncertainties,” *Mechanical Systems & Signal Processing*, vol. 22, no. 6, pp. 1263–1273, August 2008.
- [56] Y. Liang, S. Cong, and W. Shang, “Function approximation-based sliding mode adaptive control,” *Nonlinear Dynamics*, vol. 54, no. 3, pp. 223–230, November 2008.
- [57] R. W. Brockett, “Asymptotic stability and feedback stabilization,” in *Differential Geometric Control Theory*. Birkhauser, 1983, pp. 181–191.
- [58] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC Press, 1994.
- [59] J. Wittenburg, *Dynamics of Systems of Rigid Bodies*, 2nd ed. Berlin: Springer, 2008.
- [60] A. Lurie, *Analytical Mechanics*, ser. Foundations of Engineering Mechanics. Berlin: Springer-Ferlag, 2002.
- [61] J. Neimark and N. Fufaev, *Dynamics of Nonholonomic Systems*, ser. Translation of Mathematical Monographs. American Mathematical Society, 1972, vol. 33.
- [62] A. Bicchi, A. Balluchi, D. Prattichizzo, and A. Gorelli, “Introducing the ”sphericle”: an experimental testbed for research and teaching in nonholonomy,” in *In Proceedings*

of *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997, pp. 2620–2625.

- [63] S. Bhattacharya and S. Agrawal, “Spherical rolling robot: A design and motion planning studies,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 835–839, December 2000.
- [64] T. Das, R. Mukherjee, and H. Yuksel, “Design considerations in the development of a spherical mobile robot,” in *Proc. 15th SPIE Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, vol. 4364, Orlando, Florida, April 2001, pp. 61–71.
- [65] A. Javadi and P. Mojabi, “Introducing glory: A novel strategy for an omnidirectional spherical rolling robot,” *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 3, pp. 678–683, September 2004.
- [66] J. Shen, D. Schneider, and A. Bloch, “Controllability and motion planning of a multi-body chaplygin’s sphere and chaplygin’s top,” *Int. Journal of Robust and Nonlinear Control*, vol. 18, no. 9, pp. 905–945, 2008.
- [67] R. Balasubramanian, A. Rizzi, and M. Mason, “Legless locomotion: A novel locomotion technique for legged robots,” *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 575–594, May 2008.
- [68] V. Joshi and R. Banavar, “Motion analysis of a spherical mobile robot,” *Robotica*, vol. 27, no. 3, pp. 343–353, May 2009.
- [69] K. W. Wait, P. J. Jackson, and L. S. Smoot, “Self locomotion of a spherical rolling robot using a novel deformable pneumatic method,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 3757–3762.
- [70] R. Chase and A. Pandya, “A review of active mechanical driving principles of spherical robots,” *Robotics*, vol. 1, no. 1, pp. 3–23, 2012.

- [71] Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 62–72, February 1990.
- [72] V. Jurdjevic, "The geometry of the plate-ball problem," *Archive for Rational Mechanics and Analysis*, vol. 124, no. 4, pp. 305–328, December 1993.
- [73] A. Marigo and A. Bicchi, "Rolling bodies with regular surface: Controllability theory and applications," *IEEE Trans. on Automatic Control*, vol. 45, no. 9, pp. 1586–1599, September 2000.
- [74] R. Mukherjee, M. Minor, and J. Pukrushpan, "Motion planning for a spherical mobile robot: Revisiting the classical ball-plate problem," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 124, no. 4, pp. 502–511, 2002.
- [75] M. Svinin and S. Hosoe, "Motion planning algorithms for a rolling sphere with limited contact area," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 612–625, June 2008.
- [76] K. Harada, T. Kawashima, and M. Kaneko, "Rolling based manipulation under neighborhood equilibrium," *The International Journal of Robotics Research*, vol. 21, no. 5-6, pp. 463–474, 2002.
- [77] K. Suzuki, M. Svinin, and S. Hosoe, "Motion planning for rolling based locomotion," *Journal of Robotics and Mechatronics*, vol. 17, no. 5, pp. 537–545, November 2005.
- [78] A. Morinaga, M. Svinin, and M. Yamamoto, "A motion planning strategy for a spherical rolling robot driven by two internal rotors," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 993–1002, August 2014.
- [79] A. H. J. A and P. Mojab, "Introducing august: a novel strategy for an omnidirectional spherical rolling robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, 2002, pp. 3527–3533 vol.4.
- [80] Q. Zhan, T. Zhou, M. Chen, and S. Cai, "Dynamic trajectory planning of a spherical mobile robot," in *2006 IEEE Conference on Robotics, Automation and Mechatronics*, June 2006, pp. 1–6.

- [81] E. Kayacan, Z. Bayraktaroglu, and W. Saeys, “Modeling and control of a spherical rolling robot: A decoupled dynamics approach,” *Robotica*, vol. 30, no. 4, pp. 671–680, July 2012.
- [82] T. B. Ivanova and E. N. Pivovarova, “Dynamics and control of a spherical robot with an axisymmetric pendulum actuator,” *Nonlinear Dynamics & Mobile Robotics*, vol. 1, no. 1, pp. 71–85, 2013.
- [83] T. Ylikorpi, P. Forsman, A. Halme, and J. Saarinen, “Unified representation of decoupled dynamic models for pendulum-driven ball-shaped robots,” in *Proc. 28th European Conference on Modelling and Simulation*, Brescia, Italy, May 27–30 2014, pp. 411–420.
- [84] T. Ylikorpi, “Mobility and motion modelling of pendulum-driven ball-robots: Decoupled models for steering and obstacle crossing,” Ph.D. dissertation, Aalto University, Finland, 2017.
- [85] D. Luenberger, *Optimization by Vector Space Methods*. New York: John Wiley & Sons, 1969.
- [86] M. Sofronniou and R. Knapp, *Advanced Numerical Differential Equation Solving in MATHEMATICA*. Wolfram Research, Inc., 2008.
- [87] M. Svinin and S. Hosoe, “Planning of smooth motions for a ball-plate system with limited contact area,” in *Proc. IEEE Int. Conference on Robotics and Automation*, vol. 1, Pasadena, CA, May 19-23 2008, pp. 1193–1200.
- [88] T. Urakubo, M. Monno, S. Maekawa, and H. Tamaki, “Dynamic modeling and controller design for a spherical rolling robot equipped with a gyro,” *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–11, 2015.
- [89] A. Meghdari, M. Rajaei, A. Tadayon, and Y. Radparvar, “Geometric and control of a spherical mobile robot,” in *Proc. of the 2nd WSEAS Int. Conference on Applied and Theoretical Mechanics*, Venice, Italy, November 20–22 2006, pp. 91–96.

- [90] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, “Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm,” *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 170–179, February 2013.
- [91] D. V. Balandin, M. A. Komarov, and G. V. Osipov, “A motion control for a spherical robot with pendulum drive,” *Journal of Computer and Systems Sciences International*, vol. 52, no. 4, pp. 650–663, 2013.
- [92] Y. Bai, M. Svinin, and M. Yamamoto, “Backstepping trajectory tracking control for a spherical rolling robot,” in *Proc. IEEE Int. Conference on Intelligent Robots and Systems*, Daejeon, Korean, October 9–14 2016, pp. 298–303.
- [93] A. Huang, Y. Chen, and C. Kai, *Adaptive Control of Underactuated Mechanical Systems*. Singapore: World Scientific, 2015.
- [94] N. Azlan and H. Yamaura, “Fat based adaptive impedance control for unknown environment position,” *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 6, no. 8, pp. 1538–1543, August 2012.
- [95] H. Al-shuka, B. Corves, and W. Zhu, “Function approximation technique-based adaptive virtual decomposition control for a serial-chain manipulator,” *Robotica*, vol. 32, no. 3, pp. 375–399, May 2014.
- [96] H. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage, “Segway robotic mobility platform,” vol. 5609.
- [97] J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, New Jersey: Prentice Hall International Inc., 1991.
- [98] H. Sussmann, “Lie brackets and local controllability: A sufficient condition for scalar-input systems,” *SIAM J. Control and Optimization*, vol. 21, no. 5, pp. 686–713, 1983.

- [99] T. Mita, *Introduction to Nonlinear control theory: Skill Control of Underactuated Robots*. Shokodo Co. Ltd., 2000, (In Japanese).

Appendix A

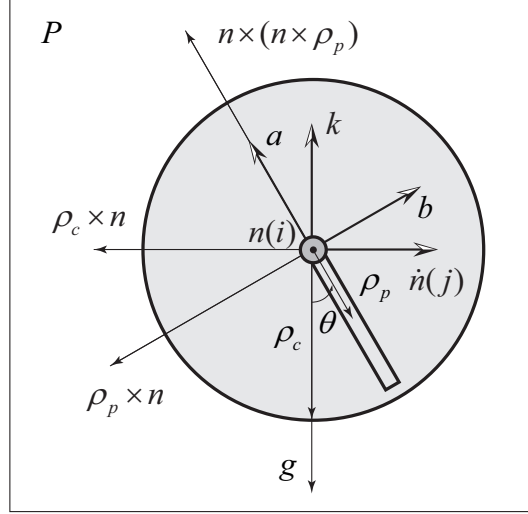


Figure 5.3: Definition of basic vectors with respect to the plane P .

The calculation of terms in the virtually constrained motion equations (2.18,2.19) is as follows. Define the unit vectors $\mathbf{a}$, $\mathbf{k}$, $\mathbf{b}$, $\mathbf{j}$ and $\mathbf{i}$ as shown in Fig. 5.3. Note that the direction of $\mathbf{i}$ is pointed out of the paper. Hence, the following basic vectors appeared in (2.18,2.19) can be expressed by the unit vectors as

$$\mathbf{n} = \mathbf{i}, \quad \dot{\mathbf{n}} = \mathbf{j}, \quad \mathbf{g} = -g\mathbf{k}, \quad \rho_c = -R\mathbf{k}, \quad \rho_p = -l\mathbf{a}, \quad (5.1)$$

through which, the following items are calculated as

$$\begin{aligned} \mathbf{J}_o \mathbf{n} &= J_o \mathbf{i}, \quad \mathbf{n}^\top \mathbf{J}_o \mathbf{n} = J_o, \quad \mathbf{n}^\top \mathbf{J}_o \dot{\mathbf{n}} = 0, \\ \mathbf{J}_p \mathbf{n} &= J_p \mathbf{i}, \quad \mathbf{n}^\top \mathbf{J}_p \mathbf{n} = J_p, \quad \mathbf{n}^\top \mathbf{J}_p \dot{\mathbf{n}} = 0, \quad \mathbf{n}^\top \mathbf{n} \times \mathbf{J}_p \mathbf{n} = 0, \\ \mathbf{n}^\top \rho_c \times (\rho_c \times \mathbf{n}) &= -R^2, \quad \mathbf{n}^\top \rho_c \times (\rho_c \times \dot{\mathbf{n}}) = 0, \\ \mathbf{n}^\top \rho_c \times (\rho_p \times \mathbf{n}) &= -Rl \cos \theta, \quad \mathbf{n}^\top \rho_c \times (\rho_p \times \dot{\mathbf{n}}) = 0, \\ \mathbf{n}^\top \rho_p \times (\rho_c \times \mathbf{n}) &= -Rl \cos \theta, \quad \mathbf{n}^\top \rho_p \times (\rho_c \times \dot{\mathbf{n}}) = 0, \\ \mathbf{n}^\top \rho_c \times (\mathbf{n} \times (\mathbf{n} \times \rho_p)) &= -Rl \sin \theta, \\ \mathbf{n}^\top \rho_p \times \mathbf{g} &= -lg \sin \theta, \end{aligned} \quad (5.2)$$

where $\theta = \phi + \theta_{\text{rel}}$ and $\dot{\theta} = \dot{\phi} + \dot{\theta}_{\text{rel}}$. Then by substituting (5.2) into (2.18,2.19), one obtains the expression for each block of (2.18,2.19), as

$$\begin{aligned}
\mathbf{n}^\top \overline{\mathbf{H}}_{oo} \mathbf{n} &= J_o + J_p + (m + m_p)R^2 - 2m_p Rl \cos \theta, \\
\mathbf{n}^\top \overline{\mathbf{H}}_{op} \mathbf{n} &= \mathbf{n}^\top \overline{\mathbf{H}}_{po} \mathbf{n} = J_p - m_p Rl \cos \theta, \\
\mathbf{n}^\top \overline{\mathbf{H}}_{pp} \mathbf{n} &= J_p, \\
\mathbf{n}^\top \left(\overline{\mathbf{H}}_{oo} \dot{\mathbf{n}}\dot{\phi} + \overline{\mathbf{H}}_{op} \dot{\mathbf{n}}\dot{\theta}_{\text{rel}} + \overline{\mathbf{h}}_o \right) &= m_p l \sin \theta (R\dot{\theta}^2 + g), \\
\mathbf{n}^\top \left(\overline{\mathbf{H}}_{po} \dot{\mathbf{n}}\dot{\phi} + \overline{\mathbf{H}}_{pp} \dot{\mathbf{n}}\dot{\theta}_{\text{rel}} + \overline{\mathbf{h}}_p \right) &= m_p g l \sin \theta.
\end{aligned}$$

Appendix B

To know whether the selected $\mathbf{G}^*$ is proper, let us check the controllability, following to [98, 99], for the reformulated cart-pole system (4.39) after introducing the auxiliary input $u = \mathbf{G}^* \mathbf{u}^*$. Noting that the drift term $\mathbf{f}$ of the reformulated system (4.39) becomes zero at the equilibrium point $\mathbf{x} = \mathbf{x}_o = (0, 0, 0, 0)$, the system is considered to be locally controllable at $\mathbf{x} = \mathbf{x}_o$ if

- a) It is locally accessible at $\mathbf{x} = \mathbf{x}_o$.
- b) All the so-called *bad* Lie brackets calculated for (4.39), can be represented by the linear combinations of the *good* ones having lesser value of p ,

where the definitions of the *good*, *bad* Lie brackets and p , can be summarized as follows.

- a) Define δ_0 , the number of times when $\mathbf{f}$ appears in $\mathbf{L}$.
- b) Define δ_1 , the number of times when $\mathbf{g}$ appears in $\mathbf{L}$.
- c) Define $p = \delta_0 + \delta_1$, the total number of times when $\mathbf{f}$ and $\mathbf{g}$ appear in $\mathbf{L}$.
- d) $\mathbf{L}$ is *bad* if δ_0 is an odd number and δ_1 is an even number.
- e) $\mathbf{L}$ is *good* if it is not *bad*.

Let us first check whether the system is locally accessible at the equilibrium point $\mathbf{x} = \mathbf{x}_o$. The Lie brackets calculated for the control system (4.39) are

$$\begin{aligned}
L_1 &= \mathbf{f}, L_2 = \mathbf{g}_1, L_3 = \mathbf{g}_2, L_4 = \mathbf{g}_3, L_5 = \mathbf{g}_4, \\
L_6 &= [\mathbf{f}, \mathbf{g}_1], L_7 = [\mathbf{f}, \mathbf{g}_2], L_8 = [\mathbf{f}, \mathbf{g}_3], L_9 = [\mathbf{f}, \mathbf{g}_4], \\
L_{10} &= [\mathbf{g}_1, \mathbf{g}_2], L_{11} = [\mathbf{g}_1, \mathbf{g}_3], L_{12} = [\mathbf{g}_1, \mathbf{g}_4], \\
L_{13} &= [\mathbf{g}_2, \mathbf{g}_3], L_{14} = [\mathbf{g}_2, \mathbf{g}_4], L_{15} = [\mathbf{g}_3, \mathbf{g}_4], \\
L_{16} &= [\mathbf{f}, [\mathbf{f}, \mathbf{g}_1]], L_{17} = [\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_1]], L_{18} = [\mathbf{g}_2, [\mathbf{f}, \mathbf{g}_1]], \\
L_{19} &= [\mathbf{g}_3, [\mathbf{f}, \mathbf{g}_1]], L_{20} = [\mathbf{g}_4, [\mathbf{f}, \mathbf{g}_1]], \\
L_{21} &= [\mathbf{f}, [\mathbf{f}, \mathbf{g}_2]], L_{22} = [\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_2]], \dots \\
L_{66} &= [\mathbf{f}, [\mathbf{f}, [\mathbf{f}, \mathbf{g}_1]]], L_{67} = [\mathbf{g}_1, [\mathbf{f}, [\mathbf{f}, \mathbf{g}_1]]], \\
L_{68} &= [\mathbf{g}_2, [\mathbf{f}, [\mathbf{f}, \mathbf{g}_1]]], L_{69} = [\mathbf{g}_3, [\mathbf{f}, [\mathbf{f}, \mathbf{g}_1]]], \\
L_{70} &= [\mathbf{g}_4, [\mathbf{f}, [\mathbf{f}, \mathbf{g}_1]]], \\
L_{71} &= [\mathbf{f}, [\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_1]]], L_{72} = [\mathbf{g}_1, [\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_1]]], \dots
\end{aligned}$$

and pick all the non-zero Lie brackets to construct the accessibility matrix

$$\mathbf{A} = \left[\begin{array}{c|c|c} L_a & L_b & \dots \end{array} \right].$$

As $\text{Rank}(\mathbf{A}) = 4$ at the equilibrium point, the system under consideration is locally accessible at the equilibrium point. The first condition for the local controllability is satisfied. Furthermore, in our case, all the *Bad* Lie brackets L_1 , L_{17} , L_{23} , L_{29} and L_{35} , are zero vectors where

$$\begin{aligned}
L_1 &= \mathbf{f}, L_{17} = [\mathbf{g}_1, [\mathbf{f}, \mathbf{g}_1]], L_{23} = [\mathbf{g}_2, [\mathbf{f}, \mathbf{g}_2]], \\
L_{29} &= [\mathbf{g}_3, [\mathbf{f}, \mathbf{g}_3]], L_{35} = [\mathbf{g}_4, [\mathbf{f}, \mathbf{g}_4]].
\end{aligned}$$

Therefore, all the *bad* brackets can be represented by the linear combinations of the *good* ones having lesser value of p . Thus the system is locally controllable at the equilibrium.

Appendix C

To test the validity of the chosen $\mathbf{G}^*$ matrix, let us then check the local controllability around the equilibrium point $\mathbf{x}_o = (0, 0, 0)$ for the reconstructed unicycle system (4.45). The Lie brackets calculated for (4.45) around the equilibrium point $\mathbf{x} = \mathbf{x}_o$ are

$$\begin{aligned} \mathbf{L}_1 &= \mathbf{g}_1, \mathbf{L}_2 = \mathbf{g}_2, \mathbf{L}_3 = \mathbf{g}_3, \\ \mathbf{L}_4 &= [\mathbf{g}_1, \mathbf{g}_2], \mathbf{L}_5 = [\mathbf{g}_1, \mathbf{g}_3], \mathbf{L}_6 = [\mathbf{g}_2, \mathbf{g}_3], \\ \mathbf{L}_7 &= [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]], \mathbf{L}_8 = [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]], \\ \mathbf{L}_9 &= [\mathbf{g}_3, [\mathbf{g}_1, \mathbf{g}_2]], \\ \mathbf{L}_{10} &= [\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_3]], \mathbf{L}_{11} = [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_3]], \\ \mathbf{L}_{12} &= [\mathbf{g}_3, [\mathbf{g}_1, \mathbf{g}_3]], \mathbf{L}_{13} = [\mathbf{g}_1, [\mathbf{g}_2, \mathbf{g}_3]], \\ \mathbf{L}_{14} &= [\mathbf{g}_2, [\mathbf{g}_2, \mathbf{g}_3]], \mathbf{L}_{15} = [\mathbf{g}_3, [\mathbf{g}_2, \mathbf{g}_3]], \end{aligned}$$

and pick all the non-zero Lie brackets at the equilibrium point to construct the accessibility matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

As $\text{Rank}(\mathbf{A}) = 3$, the system under consideration is locally accessible at the equilibrium point. For driftless system, local accessibility is equivalent to local controllability.