Adaptive Path Following for an Underactuated Nonholonomic Mobile Manipulator

by

Orlando Jair Barrera Perez

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2019

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We investigate an adaptive path following problem for an underactuated nonholonomic mobile manipulator system and closed planar curves. As opposed to adapting to uncertain or unknown dynamics in the plant, we apply an adaptation approach with respect to an unknown geometric path. First, we present a solution to the non-adaptive path following problem using the concept of a path following output and apply it to circular and elliptical paths. To overcome a drawback associated with our first proposed solution and set the stage for our approach to the adaptive case, we apply an approximation approach based on osculating circles for strictly convex closed curves.

We transition to the adaptive path following case by first presenting an algorithm to estimate unknown path parameters in the case of a circular path. We use our estimation algorithm and our path following solution for circular paths in an indirect adaptive control scheme. Thereafter, again using the osculating circle of a curve and the approximation technique of our second non-adaptive path following solution, we extend our adaptive solution, under some mild assumptions, for unknown strictly convex closed curves in the plane.

Acknowledgements

I would like to thank my supervisors Dr. Christopher Nielsen and Dr. Baris Fidan for their genius insight, constant support and incredible work ethic. I have learned a lot working with you these past two years and I believe I am a better man because of it.

I am really thankful for the amazing community of Professors at the University of Waterloo, specifically I would like to thank Dr. Andrew Heunis and Dr. David Wang. I would also like to thank my MASc. thesis readers Dr. Daniel Miller and Dr. Soo Jeon.

I would also like to thank my colleagues and friends Rollen D'Souza and Dr. Philip McCarthy your help has been immensely valuable through this gnarly adventure called grad school.

Lastly, I would like to thank all my friends, you're awesome.

Muchas gracias!

Dedication

To my parents Alberto and Maria Estela and my brother Luis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we study the adaptive path following problem for an underactuated nonholonomic mobile manipulator. In robotic applications, path following problems involve the design of feedback controllers that drive a certain physical feature of a robotic system (output) towards a known geometric path with a predetermined desired motion about the path [2, 14]. An advantageous property that differentiates path following from trajectory tracking is that path following controllers guarantee path invariance [1]. Essentially, this means that if the output of a system is initialized on the desired path with a velocity tangent to the path, it will stay on the path for all time. Trajectory tracking controllers cannot guarantee this property, therefore if the output is initialized on the path but is not consistent with the reference, the output may leave the desired path in order to match the reference [29].

On the other hand, adaptive control is the combination of control laws and parameter estimation to control classes of systems whose parameters are unknown or time-varying [7, 17, 18, 20]. The choice of estimation method and control law as well as the manner in which these two are combined leads to different classes of adaptive control schemes [18]; in this thesis we will use an indirect adaptive control approach. With this approach the unknown parameters are first estimated online and then fed into the control law to update in real time the corresponding control input.

In this thesis we combine the ideas of path following and adaptive control. Our goal is to design a path following controller that adapts to unknown and strictly convex paths for an underactuated nonholonomic mobile manipulator. This stands in contrast to the classical adaptive motion control approach of designing controllers that adapt to uncertain parameters present in the system's model [23, 24, 28, 40], i.e., in our case we assume the

plant's dynamics are known but take the path to be unknown in a sense to be made precise later in this thesis. In general, path following applications require complete knowledge of the path to be followed, while this requirement is fine in a context where there are no path uncertainties, it is not desirable in a context where the path to be followed is uncertain or not fully known. To account for situations where a lack of complete global knowledge of the desired path exists, we take on the adaptive path following problem and design a control algorithm that only requires local information about the desired path, namely about the closest point on the path to the robot's output.

We begin by proposing a solution to the non-adaptive path following problem using the notion of a path following output [22, 38] to design an input-output feedback linearizing controller with its associated normal form [19] for arbitrary, smooth, closed paths and we apply it to circular and elliptical paths. There is a drawback associated with our proposed controller when applied to arbitrary closed paths which will be explained later in the thesis; to overcome this drawback for strictly convex paths we apply a circular approximation to the path by means of the osculating circle and under some mild assumptions propose a second solution to the path following problem of strictly convex paths by using a modified version of the circular path following controller.

Then we transition into the adaptive path following problem. We start by presenting an algorithm for circular path parameter estimation that in addition to estimating the center of a desired target circle [8, 9, 11], estimates its radius. Thereafter, we design a path following controller with an indirect adaptive control approach [18] for unknown circular paths. Lastly, we extend our results to the path following adaptation of unknown strictly convex paths by estimating the osculating circle of the path at a point on the path and apply an indirect adaptive path following control approach similar to the previous one, however, this time with respect to the estimated osculating circle.

1.1 Motivating Applications

A robotic manipulator in its classical representation has a fixed base, which constrains its access to a limited static workspace that depends on the kinematic configuration of its joints [21, 35, 37]. The workspace of a mobile manipulator is dynamic in the sense that it depends on the location of the mobile base [15, 27], i.e., the mobile manipulator has the ability to change its workspace as a function of the task's requirements. Because of this, mobile manipulators open a whole new realm of opportunities in terms of manipulability. One example is the building and construction industry; in the recent years interest in 3D printing has greatly increased due to its promises of labour-intensity reduction and

safety [39]. The problem of scalability in this area has been researched and some solutions involve the use of teams of mobile manipulators that work together to achieve goals [41]. Another example, is the use of mobile manipulators in warehouse or factory environments for the automation of repetitive tasks such as material handling [31] and Industry 4.0 [42]. There exists an obvious benefit to the research and development of motion control of mobile manipulators to automate simple repetitive tasks in industry and life in general.

In the above scenarios one could imagine a situation where the desired path to be followed by the mobile manipulator changes in real-time and our mobile manipulator needs to adapt to the new and unknown desired path automatically. For example, in a warehouse environment one could imagine a situation where the floor plan changes and the desired path changes as a function of the free available space to navigate, thus forcing our mobile manipulator to adapt to said changes. Motivated by this context we take take on the problem of designing path following controllers with the ability to adapt to unknown strictly convex closed paths in the plane.

1.2 Literature Review

1.2.1 Motion Control of Mobile Manipulators

The motion control problem of mobile manipulators has gathered large interest in the research community in the last 20 years [34]; it is usually solved in one of two ways, by either using a path following or trajectory tracking approach. The latter consisting of asymptotically driving the output of a system to a curve with an assigned time parametrization [3]. In [33] a unifying trajectory tracking control approach is proposed for a wide class of mobile manipulator systems at a kinematic level. In [27] a solution is presented at the dynamic level for a class of nonholonomic mobile manipulators.

As we already mentioned, while trajectory tracking is a viable approach to the motion control problem of mobile manipulators, it can't guarantee path invariance. In this thesis we would like to go a step further and present a path following solution to the motion control problem of an underactuated nonholonomic mobile manipulator that guarantees path invariance for the non-adaptive case.

1.2.2 Path Following for Mobile Manipulators

In [26] the path following problem for a class of nonholonomic mobile manipulators is broken down into two separate path following problems, i.e., the mounted manipulator follows a geometric path in space while the nonholonomic base follows a desired curve lying on the plane. In [22] the synchronized path following problem is solved locally for a multi-agent system consisting of a differential drive mobile base and a manipulator (i.e., mobile manipulator) both agents are given curves to follow and the task of synchronizing their motions about their given paths. In [12] a robust path following controller is presented using a set stabilization approach [30] for a class mobile manipulators; the manipulator and mobile base are assumed to be dynamically decoupled.

In this thesis, we present two solutions to the non-adaptive path following problem for an underactuated nonholonomic mobile manipulator; we take our system's dynamics to be coupled. The problem is not broken down into two path following problems for both manipulator and mobile base, but instead we are given one curve to follow with the end-effector of the mobile manipulator and our path following controllers automatically control both subsystems to achieve control objectives.

1.2.3 Adaptive Path Following

The adaptive path following problem has been thoroughly investigated in the literature and as in classical adaptive control, the goal remains to design controllers that adapt to uncertainties present in the plant model. The authors of [28] present a robust adaptive path following controller for uncertain parameters and environmental disturbances for a surface vessel. In [10] another robust adaptive path following controller is presented for an underactuated surface ship.

The adaptive path following problem has also been investigated in the context of robotics systems, in [4] the authors propose an adaptive path following controller for a unicycle-like mobile robot with a cascaded structure; specifically, a kinematic controller cascaded with an adaptive dynamic controller, the latter compensating for uncertainties present in the dynamic level. Adaptive path following control for nonholonomic mobile manipulators was proposed in [25] for both kinematic and dynamic levels given parametric uncertainties present in the system's model.

While all the literature mentioned above takes the approach of adapting to parametric uncertainties present in a system's model, in this thesis, we present an adaptive approach

with respect to an unknown path. That is, we consider our model to be set or known but take the path to be unknown.

1.3 Notation

In this thesis, $\mathbb{R}$ denotes the set of real numbers and $\mathbb{C}$ denotes the complex numbers. Let $\langle x, y \rangle$ denote the inner product of the vectors x and y in $\mathbb{R}^n$. The symbol $:=$ means equal by definition and $\|\cdot\|$ denotes the Euclidean norm of a vector. The composition of maps s and h is written $s \circ h$. The function $\arg : \mathbb{C}\backslash\{0\} \to (-\pi, \pi]$ maps complex numbers to their principal argument. The unit circle is denoted $\mathbb{S}^1$ (see Appendix A). The differential of a function f evaluated at x is written $\mathrm{d}f_x$.

1.4 Contributions and Organization

As already mentioned, the goal of this thesis and its main contribution is to propose an adaptive path following control for an underactuated mobile manipulator with the objective of adapting to unknown strictly convex closed paths in the plane. We build up to the solution of this problem throughout this thesis.

In Chapter 2, we review the Lagrangian approach for deriving the governing equations of motion for mechanical systems subject to nonholonomic constraints and apply this approach to derive the mathematical model of the underactuated nonholonomic mobile manipulator. In Chapter 3, we propose two solutions to the path following problem; our first solution seeks to make the end-effector of the robot follow closed paths with no self-intersections; we apply this solution to the circular and elliptical path case; our second solution uses a novel approximation method and our circular path following controller to follow strictly convex paths in the plane. Lastly, Chapter 4 eases the assumption that the path to be followed is known and we propose an indirect adaptive path following control approach to follow and entirely traverse unknown circular and strictly convex paths.

Chapter 2

Mathematical Modelling of a Mobile Manipulator

The objective of this chapter is to derive a mathematical model of a mobile manipulator. More specifically, we consider a mobile manipulator comprising of a two degree-of-freedom manipulator arm mounted on a differential drive mobile base. To do so, we first review the Lagrangian approach for deriving the governing equations of mechanical systems subject to nonholonomic constraints, our exposition is based on [6, 35]. We then apply this approach to the aforementioned mobile manipulator.

2.1 Underactuated Nonholonomic Mechanical Systems

The dynamic model of a Lagrangian mechanical system provides a mathematical relationship between its control inputs and the evolution of its configuration variables over time. We consider a Lagrangian mechanical system whose configuration variables belong to an n-dimensional smooth manifold $\mathcal{Q}$. We restrict ourselves to the case where $\mathcal{Q}$ is the Cartesian product of r copies of $\mathbb{R}$ and $n - r$ copies of the circle $\mathbb{S}^1$. This corresponds to a robot whose joints are either revolute or prismatic. In this case, the elements of $\mathcal{Q}$ can be represented as an n-tuple $(q_1, \ldots, q_n)$ where each q_i is either in $\mathbb{R}$ if the ith joint is prismatic, or in $\mathbb{S}^1$ if the ith joint is revolute. A mechanical system with an n-dimensional configuration manifold $\mathcal{Q}$ is said to have n **degrees-of-freedom**.

A Lagrangian mechanical system [37] with an n-dimensional configuration manifold $\mathcal{Q}$ can be described in terms of a symmetric, positive definite inertia matrix $D : \mathcal{Q} \to \mathbb{R}^{n\times n}$ and a potential function $P : \mathcal{Q} \to \mathbb{R}$ which assigns, to each $q \in \mathcal{Q}$, a value of potential energy. The **Lagrangian** $L : T\mathcal{Q} \to \mathbb{R}$ of this system is defined as[1]

$$L(q, \dot{q}) = \frac{1}{2}\dot{q}^\top D(q)\dot{q} - P(q) \tag{2.1}$$

and equals the difference between the system's kinetic energy and its potential energy. In the absence of external forces, the Euler-Lagrange equations are

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_i}(q(t), \dot{q}(t)) - \frac{\partial L}{\partial q_i}(q(t), \dot{q}(t)) = 0, \quad i = 1, \ldots, n, \tag{2.2}$$

or, in short

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0 \tag{2.3}$$

where

$$\frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} \frac{\partial L}{\partial \dot{q}_1} & \cdots & \frac{\partial L}{\partial \dot{q}_n} \end{bmatrix}, \qquad \frac{\partial L}{\partial q} = \begin{bmatrix} \frac{\partial L}{\partial q_1} & \cdots & \frac{\partial L}{\partial q_n} \end{bmatrix}. \tag{2.4}$$

Curves $(q(t), \dot{q}(t))$ which satisfy Euler-Lagrange equation (2.3) have the property of being extremizers of the action functional $\int_I L(q(t), \dot{q}(t))\mathrm{d}t$, $I \subset \mathbb{R}$ and, by "Hamilton's principle of least action," describe the evolution of the Lagrangian system.

External (control) inputs $u^\top = \begin{bmatrix} u_1, \ldots, u_m \end{bmatrix}$ enter Euler-Lagrange equations via a map $B : \mathcal{Q} \to \mathbb{R}^{n\times m}$ which transforms the external input u into generalized forces. We assume that for every $q \in \mathcal{Q}$, the matrix $B(q)$ has rank m. In the presence of control inputs, the Euler-Lagrange equation (2.3) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = u^\top B^\top(q). \tag{2.5}$$

Definition 2.1.1. A Lagrangian control system is **underactuated** if $n > m$.

Using the expression (2.1) for the Lagrangian, (2.5) can be expressed in the standard vector form as

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + \nabla_q P = B(q)u. \tag{2.6}$$

[1] Here $T\mathcal{Q}$ is **tangent bundle** of $\mathcal{Q}$. This set, which is itself a manifold, can intuitively be thought of as pairs (q, v_q) with $q \in \mathcal{Q}$ and v_q a tangent vector to $\mathcal{Q}$ at q.

Here $\ddot{q}$ is viewed as an $n \times 1$ column vector, $\nabla_q P = (\mathrm{d}P_q)^\top$ is the gradient of the potential function P, viewed as a column vector, and the (i,j)th entry of the matrix $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is given by

$$c_{ij}(q,\dot{q}) = \sum_{k=1}^{n} \frac{1}{2}\left(\frac{\partial d_{ij}}{\partial q_j} + \frac{\partial d_{ik}}{\partial q_j} - \frac{\partial d_{kj}}{\partial q_i}\right), \qquad i,j \in \{1,\dots,n\}, \tag{2.7}$$

where $d_{ij}(q)$ is the (i,j)th entry of the inertia matrix $D(q)$.

The dynamic model (2.6) is an unconstrained Euler-Lagrange system. In order to model the system to be studied in this thesis, we must introduce the notion of **nonholonomic constraints**. Let us begin by considering a mechanical system subject to a linear velocity constraint that can be expressed as

$$\begin{bmatrix} a_1(q) & a_2(q) & \cdots & a_n(q) \end{bmatrix} \dot{q} = 0$$

where $\dot{q}$ is regarded as an $n \times 1$ column vector and $a_i : \mathcal{Q} \to \mathbb{R}$ are smooth functions. The constraint is said to be **holonomic** or **integrable** if (locally) there exists a real-valued function h of q such that the constraint can be expressed in the form $h(q) = 0$ or, in differential form,

$$\mathrm{d}h_q \dot{q} = 0.$$

If no such function h exists, then the constraint is said to be **nonintegrable** or **nonholonomic**. Now suppose that the mechanical system is subject to k nonintegrable velocity constraints represented by the equation

$$A(q)\dot{q} = 0, \tag{2.8}$$

where $A(q)$ is an $k \times n$ matrix with rank k for all $q \in \mathcal{Q}$. In the presence of such nonintegrable constraints, the Euler-Lagrange equation takes the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = u^\top B^\top(q) + \lambda^\top A(q), \qquad A(q)\dot{q} = 0. \tag{2.9}$$

In this model we have n second order differential equations and k constraint equations. The vector $\lambda \in \mathbb{R}^k$ is called the vector of **Lagrange multipliers**. Using the expression (2.1) for the Lagrangian, the above equation (2.9) can be expressed in the standard vector form as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + \nabla_q P = B(q)u + A^\top(q)\lambda \tag{2.10a}$$

$$A(q)\dot{q} = 0. \tag{2.10b}$$

Reduced Dynamic Model

While the equations of motion (2.10) are a valid representation of the class of systems studied in this thesis, it is convenient [5, 16] to derive a model in which the Lagrange multipliers are eliminated.

Let $G : \mathcal{Q} \to \mathbb{R}^{n\times(n-k)}$ be a smooth matrix valued function with the property that, for every $q \in \mathcal{Q}$, the image of $G(q)$ equals the kernel of $A(q)$, that is, the columns $g_i(q)$ of $G(q)$ constitute a basis for the kernel of $A(q)$ at each $q \in \mathcal{Q}$. In this case, one can replace the constraint (2.10b) with the **kinematic model**

$$\dot{q} = G(q)v = \sum_{i=1}^{n-k} g_i(q)v_i \tag{2.11}$$

where the terms v_i are usually called **pseudo-velocities** [35] to distinguish them from the generalized velocities $\dot{q}_i$. To eliminate the Lagrange multipliers, we can left multiply (2.10a) by $G^\top(q)$ to obtain the **reduced dynamic model**

$$G^\top(q)D(q)\ddot{q} + G^\top(q)C(q,\dot{q})\dot{q} + G^\top(q)\nabla_q P = G^\top(q)B(q)u, \tag{2.12}$$

which is a system of $n-k$ second order differential equations.

Differentiation of (2.11) with respect to time gives

$$\ddot{q} = G(q)\dot{v} + \dot{G}(q)v.$$

Left multiplying this expression by $G^\top(q)D(q)$ and then using the reduced dynamics (2.12) lead to

$$G^\top(q)D(q)G(q)\dot{v} + G^\top(q)D(q)\dot{G}(q)v = -G^\top(q)C(q,\dot{q})\dot{q} - G^\top(q)\nabla_q P + G^\top(q)B(q)u \tag{2.13}$$

Define

$$M(q) \coloneqq G^\top(q)D(q)G(q), \tag{2.14}$$

$$m(q,v) \coloneqq G^\top(q)D(q)\dot{G}(q)v + G^\top(q)C(q,G(q)v)G(q)v + G^\top(q)\nabla_q P, \tag{2.15}$$

and use (2.11), (2.13), (2.14) and (2.15), to express the constrained model (2.10) as

$$\dot{q} = G(q)v \tag{2.16a}$$

$$M(q)\dot{v} + m(q,v) = G^\top(q)B(q)u. \tag{2.16b}$$

Equation (2.16) is a system of $2n - k$ first order differential equations. Since the inertia matrix $D(q)$ is positive definite and since $G(q)$ is one-to-one, it follows that $M(q)$ is positive definite for every $q \in \mathcal{Q}$ and therefore we can express (2.16) in state-space form

$$\begin{aligned} \dot{q} &= G(q)v \\ \dot{v} &= -M^{-1}(q)m(q,v) + M^{-1}(q)G^{\top}(q)B(q)u. \end{aligned} \tag{2.17}$$

Assumption 2.1. The number k of nonholonomic constraints in the mechanical system (2.17) equals its degree of underactuation $n - m$. Furthermore, for all $q \in \mathcal{Q}$, the $m \times m$ matrix $G^{\top}(q)B(q)$ is non-singular. ♦

Under Assumption 2.1, the preliminary feedback control law

$$u = \left(G^{\top}(q)B(q)\right)^{-1}\left(m(q,v) + M(q)\tau\right), \tag{2.18}$$

where $\tau \in \mathbb{R}^m$ is an auxiliary control input yet to be specified, is well-defined for all $q \in \mathcal{Q}$; the partially compensated system under the feedback (2.18) is

$$\begin{aligned} \dot{q} &= G(q)v \\ \dot{v} &= \tau. \end{aligned} \tag{2.19}$$

Finally, for future use, let $x \coloneqq \begin{bmatrix} q^{\top} & v^{\top} \end{bmatrix}^{\top} \in \mathcal{Q} \times \mathbb{R}^m$ denote the state vector.

In control we are interested in making the output of a system do something useful. For the system (2.19) we will only consider outputs y that are functions of the generalized coordinates q, i.e., $y = h(q)$. This includes, as an important special case, the forward kinematics of a kinematic chain. We'll write this function as $h : \mathcal{Q} \to \mathcal{Y}$ and refer to the codomain $\mathcal{Y}$ of the output function h as the **task space**.

2.2 Modelling of a SCARA Mobile Manipulator

In this section we apply the general modelling technique of Section 2.1 to a mobile manipulator consisting of a SCARA manipulator mounted on a differential drive mobile base; Figure 2.1 shows a schematic diagram of this system. This mobile manipulator will be the object of study for the remainder of this thesis.

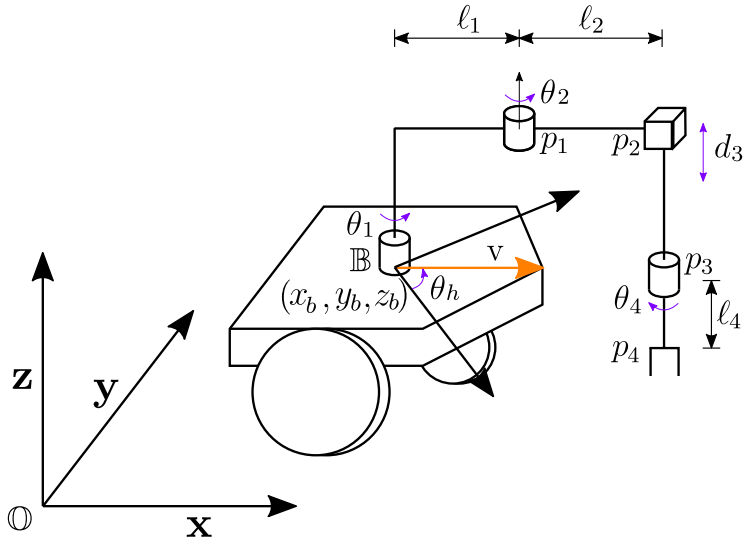

Figure 2.1: Schematic diagram of mobile manipulator.

In what follows, we ignore the vertical motion of the end-effector and instead focus on the first two links of the SCARA manipulator. In particular, we will assume that the prismatic joint variable $d_3 \in \mathbb{R}$ and the revolute joint variable $\theta_4 \in \mathbb{S}^1$ equal zero at all times. As a result, our system's schematic diagram in Figure 2.1 simplifies to the planar one shown in Figure 2.2.

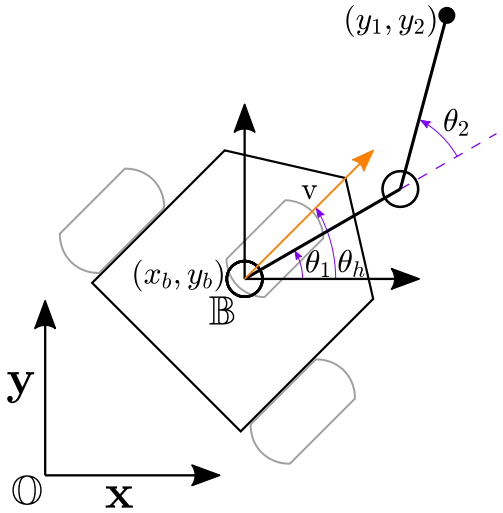

Figure 2.2: Simplified schematic diagram of mobile manipulator.

We further assume that manipulator links have their masses concentrated at their endpoints. While this assumption simplifies the mathematical expressions, the resulting model still captures the dynamic coupling between the base and the manipulator. Furthermore, the control approaches presented in this thesis are still applicable for other link mass distributions.

We begin our modelling by choosing generalized coordinates. Fix an inertial frame in $\mathbb{R}^2$ and denote it by $\mathbb{O}$ as shown in Figure 2.2. Next, let us fix a body frame $\mathbb{B}$ which moves with mobile base. Two of our generalized coordinates are $\begin{bmatrix} x_b & y_b \end{bmatrix}^\top \in \mathbb{R}^2$ which represent the location of the origin of $\mathbb{B}$ in $\mathbb{O}$ coordinates. The variable $\theta_h \in \mathbb{S}^1$ represents the angle of the rotation that aligns the coordinate axes of $\mathbb{O}$ with those of $\mathbb{B}$. Finally, $\theta_1 \in \mathbb{S}^1$ represents the angle that link 1 makes with the x_b-axis and $\theta_2 \in \mathbb{S}^1$ is the rotation which takes link 1 and makes it parallel to link 2. Thus we have $\mathcal{Q} = \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{S}^1$, $n = \dim(\mathcal{Q}) = 5$, and $q \coloneqq \begin{bmatrix} \theta_1 & \theta_2 & x_b & y_b & \theta_h \end{bmatrix}^\top$.

To compute this system's Langrangian (2.1) we denote the centre of mass locations for the base, link 1 and link 2 in the inertial frame $\mathbb{O}$ in terms of our generalized coordinates by, respectively,

$$p_0(q) = \begin{bmatrix} x_b \\ y_b \end{bmatrix}, \quad p_1(q) = p_0(q) + \ell_1 \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix}, \quad p_2(q) = p_1(q) + \ell_2 \begin{bmatrix} \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) \end{bmatrix}. \quad (2.20)$$

Let m_b, m_1 and m_2 denote the masses of, respectively, the base, link 1 and link 2, and let I_z be the moment of inertia of the mobile base about the axis of rotation at its centre of mass. Then the kinetic energy of the system is given by

$$K(q, \dot{q}) = \underbrace{\sum_{i=1}^{2} \frac{m_i}{2} \|\dot{p}_i(q)\|_2^2}_{\text{Manipulator}} + \underbrace{\frac{1}{2} m_b (\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2} I_z \dot{\theta}_h^2}_{\text{Mobile Base}}.$$

After some straightforward computations the kinetic energy can be written in the form $K(q, \dot{q}) = \frac{1}{2} \dot{q}^\top D(q) \dot{q}$ discussed in Section 2.1 with $D : \mathcal{Q} \to \mathbb{R}^{5 \times 5}$ given by

$$D(q) = \begin{bmatrix} d_{11}(q) & d_{12}(q) & d_{13}(q) & d_{14}(q) & 0 \\ d_{21}(q) & \ell_2^2 m_2 & d_{23}(q) & d_{24}(q) & 0 \\ d_{31}(q) & d_{32}(q) & m_b + m_1 + m_2 & 0 & 0 \\ d_{41}(q) & d_{42}(q) & 0 & m_b + m_1 + m_2 & 0 \\ 0 & 0 & 0 & 0 & I_z \end{bmatrix} \quad (2.21)$$

where

$$
\begin{aligned}
d_{11}(q) &= \ell_1^2 m_1 + \ell_1^2 m_2 + 2\ell_1 \ell_2 m_2 \cos(\theta_2), \\
d_{21}(q) &= d_{12}(q) = \ell_2^2 m_2 + \ell_1 \ell_2 m_2 \cos(\theta_2), \\
d_{31}(q) &= d_{13}(q) = -\ell_1 m_1 \sin(\theta_1) - \ell_1 m_2 \sin(\theta_1) - \ell_2 m_2 \sin(\theta_1 + \theta_2), \\
d_{41}(q) &= d_{14}(q) = \ell_1 m_1 \cos(\theta_1) + \ell_1 m_2 \cos(\theta_1) + \ell_2 m_2 \cos(\theta_1 + \theta_2), \\
d_{32}(q) &= d_{23}(q) = -\ell_2 m_2 \sin(\theta_1 + \theta_2), \\
d_{42}(q) &= d_{24}(q) = \ell_2 m_2 \cos(\theta_1 + \theta_2).
\end{aligned}
$$

Since we are ignoring the vertical motion of the SCARA manipulator, the robot is planar and therefore this system has constant, and without loss of generality zero, potential energy $P(q) = 0$ and thus the Lagrangian of the mobile manipulator is simply $L(q, \dot{q}) = \frac{1}{2}\dot{q}^\top D(q)\dot{q}$.

We assume that the mobile base rolls without slipping; this constraint can be written in the form (2.8) with

$$
A(q) = \begin{bmatrix} 0 & 0 & \sin(\theta_h) & -\cos(\theta_h) & 0 \end{bmatrix}. \tag{2.22}
$$

This constraint is known to be nonintegrable and so this system has $k = 1$ nonholonimic constraints. The control inputs to our system are taken to be torques τ_1 and τ_2 applied at joints 1 and 2 respectively; the translational acceleration a of the base and the angular acceleration α of the base's heading angle θ_h. This results in $u \coloneqq \begin{bmatrix} \tau_1 & \tau_2 & a & \alpha \end{bmatrix}^\top \in \mathbb{R}^4$, $m = 4$, and $B : \mathcal{Q} \to \mathbb{R}^{5\times 4}$ defined as

$$
B(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta_h) & 0 \\ 0 & 0 & \sin(\theta_h) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.23}
$$

Equations (2.21), (2.22), (2.23) with $L(q, \dot{q}) = \frac{1}{2}\dot{q}^\top D(q)\dot{q}$ provide all the expressions needed to establish the constrained Euler-Lagrange equation (2.9) for the mobile manipulator system. In terms of the vector form of these equations (2.10), we have that $\nabla_q P = 0$ and

$$
C(q, \dot{q}) = \begin{bmatrix} -2\ell_1 \ell_2 m_2 \sin(\theta_2)\, \dot{\theta}_2 & -\ell_1 \ell_2 m_2 \sin(\theta_2)\, \dot{\theta}_2 & 0 & 0 & 0 \\ \ell_1 \ell_2 m_2 \sin(\theta_2)\, \dot{\theta}_1 & 0 & 0 & 0 & 0 \\ c_{31}(q, \dot{q}) & -\ell_2 m_2 \cos(\theta_1 + \theta_2)\, \dot{\theta}_2 & 0 & 0 & 0 \\ c_{41}(q, \dot{q}) & -\ell_2 m_2 \sin(\theta_1 + \theta_2)\, \dot{\theta}_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.24}
$$

where

$$
\begin{aligned}
c_{31}(q, \dot{q}) &= -(\ell_1 m_1 \cos(\theta_1) + \ell_1 m_2 \cos(\theta_1) + \ell_2 m_2 \cos(\theta_1 + \theta_2))\dot{\theta}_1 - 2\ell_2 m_2 \cos(\theta_1 + \theta_2)\, \dot{\theta}_2, \\
c_{41}(q, \dot{q}) &= -(\ell_1 m_1 \sin(\theta_1) + \ell_1 m_2 \sin(\theta_1) + \ell_2 m_2 \sin(\theta_1 + \theta_2))\dot{\theta}_1 - 2\ell_2 m_2 \sin(\theta_1 + \theta_2)\, \dot{\theta}_2.
\end{aligned}
$$

Reduced Dynamic Model

To obtain the reduced dynamic model we start by finding the matrix valued function $G : \mathcal{Q} \to \mathbb{R}^{5\times 4}$ whose columns for a basis for the constraint $A(q)$ in (2.22). In the case of the mobile manipulator it turns out that $G(q) = B(q)$ and that the vector of pseudo-velocities is given by $v = \begin{bmatrix}\dot\theta_1 & \dot\theta_2 & \mathrm{v} & \dot\theta_h\end{bmatrix}^\top \in \mathbb{R}^4$ where v is the translational velocity of the base; this gives the kinematic model (2.11) as

$$\dot q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta_h) & 0 \\ 0 & 0 & \sin(\theta_h) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} v.$$

Once again simple calculations give that the expressions (2.14) and (2.15) evaluate to

$$M(q) = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) & 0 \\ M_{21}(q) & \ell_2^2 m_2 & M_{23}(q) & 0 \\ M_{31}(q) & M_{32}(q) & m_b + m_1 + m_2 & 0 \\ 0 & 0 & 0 & I_z \end{bmatrix} \tag{2.25}$$

where

$$\begin{aligned}
M_{11}(q) &= \ell_1^2 m_1 + \ell_1^2 m_2 + \ell_2^2 m_2 + 2\ell_1\ell_2 m_2 \cos(\theta_2), \\
M_{21}(q) &= M_{12}(q) = \ell_2^2 m_2 + \ell_1\ell_2 m_2 \cos(\theta_2), \\
M_{31}(q) &= M_{13}(q) = -\ell_1 m_1 \sin(\theta_1 - \theta_h) - \ell_1 m_2 \sin(\theta_1 - \theta_h) - \ell_2 m_2 \sin(\theta_1 + \theta_2 - \theta_h), \\
M_{32}(q) &= M_{23}(q) = -\ell_2 m_2 \sin(\theta_1 + \theta_2 - \theta_h).
\end{aligned}$$

and

$$m(q, v) = \begin{bmatrix} m_1(q, v) \\ m_2(q, v) \\ m_3(q, v) \\ 0 \end{bmatrix} \tag{2.26}$$

where

$$\begin{aligned}
m_1(q, v) &= -\ell_1\ell_2 m_2 \sin(\theta_2)(\dot\theta_2^2 + \dot\theta_1\dot\theta_2) + (\ell_1 m_1 \cos(\theta_1 - \theta_h) + \ell_1 m_2 \cos(\theta_1 - \theta_h) + \\
&\qquad \ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h))\mathrm{v}\dot\theta_h, \\
m_2(q, v) &= \ell_1\ell_2 m_2 \sin(\theta_2)\dot\theta_1^2 + \ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h)\mathrm{v}\dot\theta_h, \\
m_3(q, v) &= -(\ell_1 m_1 \cos(\theta_1 - \theta_h) + \ell_1 m_2 \cos(\theta_1 - \theta_h))\dot\theta_1^2 - (\ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h))(\dot\theta_1^2 + \dot\theta_2^2) \\
&\qquad -2\ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h)\dot\theta_1\dot\theta_2.
\end{aligned}$$

As we have outlined, for the mobile manipulator system we have $n = 5$, $m = 4$ and $k = 1$ and therefore the first part of Assumption 2.1 holds. Since $G^\top(q)B(q) = I_4$, the second part of Assumption 2.1 also holds. Therefore, we can apply the preliminary feedback control law (2.18), which in this case simplifies to

$$u = m(q, v) + M(q)\tau, \tag{2.27}$$

where $m(q, v)$ and $M(q)$ are given by, respectively, (2.25) and (2.26) and $\tau \in \mathbb{R}^4$ an auxiliary input that we will design in the next chapter. Under this feedback control law the partially compensated mobile manipulator system dynamics has the form (2.19), which we re-write here for convenience:

$$\begin{aligned} \dot{q} &= G(q)v \\ \dot{v} &= \tau. \end{aligned} \tag{2.28}$$

The output is taken to be

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = h(q) \coloneqq \begin{bmatrix} x_b + \ell_1 \cos(\theta_1) + \ell_2 \cos(\theta_1 + \theta_2) \\ y_b + \ell_1 \sin(\theta_1) + \ell_2 \sin(\theta_1 + \theta_2) \\ \theta_2 \\ \arg\left(\exp(j(\theta_1 - \theta_h))\right) \end{bmatrix}, \tag{2.29}$$

where $h : \mathcal{Q} \to \mathcal{Y}$, $\mathcal{Y} \coloneqq \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^1$. In (2.29), $\begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top$ equals the position of the end of link 2, i.e., the end-effector position in the inertial frame $\mathbb{O}$ while the third and fourth components are user defined functions, or virtual holonomic constraints, that serve to restrict the motion our system in task space. In particular, when $y_3 \equiv 0$ link 2 is constrained to be aligned with link 1 and when $y_4 \equiv 0$, link 1 is restricted to be aligned with the heading vector of the mobile base. Note that, for y_4 we use $\arg(\exp(j(\theta_1 - \theta_h)))$ instead of $\theta_1 - \theta_h$ to avoid "unwinding" issues.

Chapter 3

Path Following for Closed Paths

In this chapter we use the model of the mobile manipulator derived in Chapter 2 to present a control solution to the path following problem; we seek to make the end-effector of the robot follow closed paths with no self-intersections. We then specialize our controller to the computationally simple case of circular paths as well as elliptical paths. Finally, we demonstrate that this simple circular controller can be used to follow strictly convex closed paths by using a novel approximation method.

3.1 Problem Formulation

The dynamic model of the simplified SCARA mobile manipulator considered in Chapter 2, after application of the preliminary feedback defined by (2.25), (2.26) and (2.27) is given by

$$\begin{aligned} \dot{q} &= G(q)v \\ \dot{v} &= \tau, \end{aligned} \tag{3.1}$$

where the configuration variables are $q \coloneqq \begin{bmatrix} \theta_1 & \theta_2 & x_b & y_b & \theta_h \end{bmatrix}^\top \in \mathcal{Q}$, the pseudo-velocities are $v = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \mathrm{v} & \dot{\theta}_h \end{bmatrix}^\top \in \mathbb{R}^4$, $\tau \in \mathbb{R}^4$ is an auxiliary input yet to be specified and

$$G(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta_h) & 0 \\ 0 & 0 & \sin(\theta_h) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.2}$$

and the system output is taken to be (2.29).

3.1.1 Admissible Paths

In this chapter we assume that we are given a *known*[1], smooth, closed and regular path, with no self intersections in the inertial frame $\mathbb{O}$ that is represented parametrically as

$$\sigma : \mathbb{S}^1 \to \mathbb{R}^2, \quad \sigma'(\lambda) \neq 0 \text{ for all } \lambda \in \mathbb{S}^1. \tag{3.3}$$

Let $\mathcal{C} \coloneqq \sigma(\mathbb{S}^1)$ denote the closed curve which is the image of σ.

Assumption 3.1. In addition to the parametric representation (3.3), there exists a known smooth function $s : U \subseteq \mathbb{R}^2 \to \mathbb{R}$, whose domain is an open and connected set, such that

$$\mathcal{C} = \{(y_1, y_2) \in \mathbb{R}^2 : s(y_1, y_2) = 0\} \tag{3.4}$$

and, for all $(y_1, y_2) \in \mathcal{C}$, $\mathrm{d}s_{(y_1,y_2)} \neq 0$. ♦

Assumption 3.1 asks that the path $\mathcal{C}$ also be represented as the zero level set of a smooth function and that the Jacobian of said function have full rank at each point on the curve. The "path" assigned to the remaining outputs y_3 and y_4 of our system (2.29), corresponding to the user defined virtual holonomic constraints, is simply a point, i.e., we seek to regulate y_3 and y_4 to zero. As discussed at the end of Chapter 2, when $y_3 \equiv 0$ the second link is constrained to be aligned with the first and when $y_4 \equiv 0$, the first link is restricted to be aligned with the base's heading.

With these assumptions, the path in the task space $\mathcal{Y} = \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^1$ of the SCARA mobile manipulator equals the set

$$\gamma \coloneqq \{y \in \mathcal{Y} : s(y_1, y_2) = y_3 = y_4 = 0\} = \left\{ y = \begin{bmatrix} \sigma(\lambda) \\ 0 \\ 0 \end{bmatrix} : \lambda \in \mathbb{S}^1 \right\} \subset \mathcal{Y}, \tag{3.5}$$

the set γ is an embedded submanifold of $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^1$.

3.1.2 Desired Motion Along the Assigned Path

In the path following problem we seek to drive the output y of the mobile manipulator towards the target set γ; this is sometimes called the **geometric task** [36]. Additionally,

[1]This stands in contrast to Chapter 4 where we consider adaptive control and ease this assumption.

many applications also require that the output move along γ in a useful way, e.g., we may want that the end-effector traverse the entire curve $\mathcal{C}$; such specifications are sometimes called the **dynamic task** [36]. To model this desired motion along $\mathcal{C}$, we invoke the notion of a timing law generated by an exosystem which we assume has linear dynamics.

Assumption 3.2. The desired motion along the assigned path (3.3) is described by a timing law $\lambda^{\text{ref}} : \mathbb{R} \to \mathbb{S}^1$ so that at each moment in time, $\sigma(\lambda^{\text{ref}}(t))$ represents the desired location along the path. The timing law is produced by a known exogenous system

$$\dot{w}(t) = Sw(t), \qquad w(0) \in \mathbb{R}^r, \tag{3.6a}$$

$$\lambda^{\text{ref}}(t) = \arg\left(\exp(jQw(t)\right) \tag{3.6b}$$

with $S \in \mathbb{R}^{r\times r}$, $Q \in \mathbb{R}^{1\times r}$ and where $\arg : \mathbb{C}\backslash\{0\} \to (-\pi, \pi]$ is the principle argument. ♦

We present an example to illustrate the utility of the timing law.

Example 3.1.1. Consider the case of a unit circle centred at the origin of the inertial frame $\mathbb{O}$; in this case the parameterized representation (3.3) is given by

$$\sigma(\lambda) = \begin{bmatrix} \cos(\lambda) \\ \sin(\lambda) \end{bmatrix}$$

and the zero level set representation in Assumption 3.1 is given by, say, $s(y_1, y_2) = \sqrt{y_1^2 + y_2^2} - 1$. If our application asks that the robot's end-effector move periodically between the "north pole" and the "south pole" of the circle, then one choice of timing law might be

$$\dot{w}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} w(t), \qquad w(0) = (1, 0),$$
$$\lambda^{\text{ref}}(t) = \arg\left(\exp\left(j \begin{bmatrix} \frac{\pi}{2} & 0 \end{bmatrix} w(t)\right)\right).$$

If instead we want the robot's end-effector to traverse the entire circle in a counter-clockwise direction, then the timing law might be

$$\dot{w}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} w(t), \qquad w(0) = (0, 1),$$
$$\lambda^{\text{ref}}(t) = \arg\left(\exp\left(j \begin{bmatrix} 1 & 0 \end{bmatrix} w(t)\right)\right).$$

▲

3.1.3 Problem Statement

We are now ready to state the two main problems considered in this thesis.

Problem 1 (Path following). *Consider the mobile manipulator model (3.1) with output (2.29). Suppose we are given a closed parametric path (3.3) that satisfies Assumption 3.1 and a desired motion along the path that satisfies Assumption 3.2. Find, if possible, a state-feedback controller $\tau : \mathcal{Q} \times \mathbb{R}^4 \to \mathbb{R}^4$ such that the closed-loop system enjoys the following properties:*

(i) The output is driven towards the path (3.5). In particular, there exists an open set in $\mathcal{Q} \times \mathbb{R}^4$ such that any initial condition in this set results in

$$y(t) \to \gamma \quad as \quad t \to \infty \ (attractivity)$$

with $(q(t), v(t))$ bounded.

(ii) The path (3.5) is output invariant. In particular, for every $(q(0), v(0)) \in \mathcal{Q} \times \mathbb{R}^4$, and for every $\lambda_0 \in \mathbb{S}^1$, $\alpha \in \mathbb{R}$, such

- *$y(0) = h(q(0)) = \begin{bmatrix} \sigma(\lambda_0) \\ 0 \\ 0 \end{bmatrix}$, (output is initialized on path)*
- *$\frac{\mathrm{d}y}{\mathrm{d}t}\big|_{t=0} \in \mathsf{T}_{y(0)}\gamma$, (output is initially moving tangent to the path)*

the output $y(t)$ belongs to γ for all $t \geq 0$.

(iii) The output asymptotically converges to the desired motion along the path

$$\lim_{t \to \infty} \left\| y(t) - \begin{bmatrix} \sigma(\lambda^{\mathrm{ref}}(t)) \\ 0 \\ 0 \end{bmatrix} \right\| = 0.$$

In Chapter 4 we will consider an adaptive version of this problem in the special case where the path is given by a circle.

Problem 2 (Adaptive path following). *Consider the mobile manipulator model (3.1) with output (2.29). Suppose we are given a curve $\mathcal{C}$, which is a circle of unknown radius and unknown location in the plane. Further suppose that the controller has access to the entire state $(q, v) \in \mathcal{Q} \times \mathbb{R}^4$ and access to, at all times, the signed distance from the end-effector to the circular path as well as the rate of change of this distance. Find, if possible, a dynamic control law τ such that the closed-loop system enjoys the following properties:*

(i) The output is driven towards the path (3.5). In particular, there exists an open set in $\mathcal{Q} \times \mathbb{R}^4$ such that, for any initial condition in the set results in

$$y(t) \to \gamma \quad as \quad t \to \infty \ (attractivity)$$

with $(q(t), v(t))$ bounded.

(ii) The output traverses the entirety of the circular path in a user defined direction, i.e., either the clockwise or counterclockwise direction.

3.2 Path Following Output and Control Design

To solve the aforementioned problems, it is convenient to define the so-called path following output [22, 38] for the mobile manipulator. Let $U \subseteq \mathbb{R}^2$ be an open set containing the curve $\mathcal{C}$ with the property that if $(y_1, y_2) \in U$, then there exists a (unique) closest point on $\mathcal{C}$. Without loss of generality, we assume that U equals the previously defined open set U discussed in Assumption 3.1. Next, define a function $\varpi : U \subset \mathbb{R}^2 \to \mathbb{S}^1$

$$\varpi(y_1, y_2) = \arg\min_{\lambda \in \mathbb{S}^1} \left\| \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \sigma(\lambda) \right\|. \tag{3.7}$$

Intuitively, ϖ returns the parameter $\lambda^* \in \mathbb{S}^1$ with the property that $\sigma(\lambda^*)$ is the closest point on the curve $\mathcal{C}$ to (y_1, y_2). Combining the function (3.7) with the function (3.4) from Assumption 3.1, we define the **path following output** y_{PF} using the map $h_{\text{PF}} : U \times \mathbb{S}^1 \times \mathbb{S}^1 \subseteq \mathcal{Y} \to \mathbb{R} \times \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$

$$y_{\text{PF}} = h_{\text{PF}}(y) = \begin{bmatrix} s(y_1, y_2) \\ \varpi(y_1, y_2) \\ y_3 \\ y_4 \end{bmatrix}. \tag{3.8}$$

We write $y_{\text{PF}} = h_{\text{PF}}(y) = h_{\text{PF}} \circ h(q)$ to distinguish the value of the path following output (3.8) from the output y given by (2.29). The function h_{PF} is a diffeomorphism onto its image and so it can be viewed as a coordinate change on an open set of the task-space $\mathcal{Y}$ containing the set (3.5).

To understand why the path following output (3.8) is useful for solving Problem 1, observe that driving the robot's physical output (2.29) to the path (3.5) is equivalent, under mild technical conditions, to driving the 1st, 3rd and 4th components of the path

following output (3.8) to zero. Furthermore, converging to the desired motion along the path is equivalent to driving the second component of the path following output (3.8) to $\lambda^{\text{ref}}(t)$.

Consider the mobile manipulator (3.1) with the path following output

$$\begin{aligned} \dot{q} &= G(q)v \\ \dot{v} &= \tau \\ y_{\text{PF}} &= h_{\text{PF}} \circ h(q). \end{aligned} \tag{3.9}$$

First we show that this system can be input-output feedback linearized in an open set containing the path. Observe that

$$\ddot{y}_{\text{PF}} = \frac{\partial}{\partial q}\left(\mathrm{d}h_{\text{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)v\right)G(q)v + \mathrm{d}h_{\text{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)\tau.$$

Turning to the expression multiplying the input $\tau \in \mathbb{R}^4$, usually called the **decoupling matrix**, we have

$$\mathrm{d}h_{\text{PF}}|_{h(q)} = \begin{bmatrix} \mathrm{d}s_{(y_1,y_2)} & 0 & 0 \\ \mathrm{d}\varpi_{(y_1,y_2)} & 0 & 0 \\ 0 \quad 0 & 1 & 0 \\ 0 \quad 0 & 0 & 1 \end{bmatrix}_{h(q)} \in \mathbb{R}^{4\times 4}.$$

By [14, Lemma 3.1], the differentials $\mathrm{d}s_{(y_1,y_2)}$ and $\mathrm{d}\varpi_{(y_1,y_2)}$ are non-zero and linearly independent for all $(y_1, y_2) \in \mathcal{C}$ and therefore this 4×4 matrix is non-singular, without loss of generality by shrinking U if necessary, on the set $U \times \mathbb{S}^1 \times \mathbb{S}^1 \subset \mathcal{Y}$. Next using (3.2) and (2.29) we directly compute

$$\det\left(\mathrm{d}h_q\ G(q)\right) = -\ell_1 \cos(\theta_1 - \theta_h) - \ell_2 \cos(\theta_1 + \theta_2 - \theta_h).$$

On the path (3.5), $\theta_1 - \theta_h = 0$ and $\theta_2 = 0$ and so $\det\left(\mathrm{d}h_q\ G(q)\right)|_{\gamma} = -\ell_1 - \ell_2 \neq 0$. These arguments show that, for all q such that $h(q) \in \gamma$, $\mathrm{d}h_{\text{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q) \in \mathbb{R}^{4\times 4}$ is non-singular. Thus, we have shown that the system (3.9) has vector relative degree $\{2, 2, 2, 2\}$ at each $q \in h^{-1}(\gamma)$. This means that there is an open set in the configuration space $\mathcal{Q}$ containing $h^{-1}(\gamma)$, which we take without loss of generality to be $h^{-1}(U \times \mathbb{S}^1 \times \mathbb{S}^1)$, in which the input-output feedback linearizing controller

$$\tau := \left(\mathrm{d}h_{\text{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)\right)^{-1}\left(\frac{\partial}{\partial q}\left(\mathrm{d}h_{\text{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)v\right)G(q)v + v_{\text{aux}}\right), \tag{3.10}$$

where $v_{\text{aux}} \in \mathbb{R}^4$ is yet another auxiliary input to be designed, is well-defined.

A natural question to ask is whether or not the system (3.9) is minimum phase. The next result shows that (3.9) is minimum phase but not strictly minimum phase.

Lemma 3.2.1. *The zero dynamics manifold of*

$$\begin{aligned} \dot{q} &= G(q)v, \\ \dot{v} &= \tau, \\ y_{PF} &= h_{PF} \circ h(q), \end{aligned} \tag{3.11}$$

is diffeomorphic to $\mathbb{S}^1$ *and consists solely of equilibria.*

Proof. Let

$$\begin{aligned} \mathcal{Z} &\coloneqq \left\{(q,v) \in \mathcal{Q} \times \mathbb{R}^4 : y_{PF} = 0\right\}, \\ &= \{q \in \mathcal{Q} : h_{PF} \circ h(q) = 0\} \times \mathbb{R}^4, \end{aligned} \tag{3.12}$$

be the *output zeroing manifold*, we proceed to find the *zero dynamics manifold*[2] $\mathcal{Z}^\star$ by computing the largest controlled invariant set contained in $\mathcal{Z}$.

First, we have that $\mathcal{Z}^\star \subseteq \{(q,v) \in \mathcal{Z} : \dot{y}_{PF} = 0\}$ and by chain rule

$$\dot{y}_{PF} = \mathrm{d}h_{\mathrm{PF}}|_{h(q)} \; \mathrm{d}h_q \; G(q)v,$$

where $\mathrm{d}h_{PF}$ is non-singular on γ, so it is non-singular on $\mathcal{Z}$. Also

$$\det(\mathrm{d}h_q \; G(q)) = -\ell_1 \cos(\theta_1 - \theta_h) - \ell_2 \cos(\theta_1 + \theta_2 - \theta_h),$$

therefore when $(q,v) \in \mathcal{Z}$, $\theta_1 - \theta = 0$ and $\theta_2 = 0$ and so

$$\det(\mathrm{d}h_q \; G(q))\Big|_{\mathcal{Z}} = -\ell_1 - \ell_2 \neq 0.$$

Therefore, since $\mathrm{d}h_{\mathrm{PF}}|_{h(q)} \; \mathrm{d}h_q \; G(q)$ is non-singular on $\mathcal{Z}$, this implies $\dot{y}_{PF}|_{\mathcal{Z}} = 0$ if, and only if, $v = 0$, thus

$$\mathcal{Z}^\star \subseteq \{(q,v) \in \mathcal{Q} \times \mathbb{R}^4 : y_{PF} = 0, v = 0\}. \tag{3.13}$$

Finally the set to the right of (3.13) can be made controlled invariant since $\dot{v} = \tau$ and we can pick $\tau^\star(q,v) = 0$, so that

$$\mathcal{Z}^\star = \{(q,v) \in \mathcal{Q} \times \mathbb{R}^4 : y_{PF} = 0, v = 0\},$$

from the above we immediately see that

$$\begin{aligned} \dot{q}\big|_{\mathcal{Z}^\star} &= 0 \\ \dot{v}\big|_{\mathcal{Z}^\star} &= 0 \end{aligned}$$

[2]See Appendix B for the definition.

so that $\mathcal{Z}^\star$ is a set consisting of equilibria as claimed in Lemma 3.2.1.

Note that $dim(\mathcal{Z}^\star) = 1$ and the remaining degree of freedom is the particular values of θ_1 and θ_h that satisfy y_4. Therefore the value of, say θ_h, is not unique on $\mathcal{Z}^\star$. It follows that $\mathcal{Z}^\star$ is diffeomorphic to $\mathbb{S}^1$. □

It is well-known that if a nonlinear control system has a well-defined relative degree at a point, then there exists a local coordinate transformation which puts the system into the Byrnes-Isidori normal form in a neighbourhood of that point [19]. We now show that, in the case of the mobile manipulator with path following output (3.9), the system can be put into the Byrnes-Isidori normal form on an open set of its state-space $\mathcal{Q} \times \mathbb{R}^4$ whose image under h includes the path (3.5), i.e., the normal form is valid in a neighbourhood of the *entire path*, not just a neighbourhood of a point on the path.

We begin by defining the candidate coordinate transformation

$$T(q, v) \coloneqq \begin{bmatrix} \theta_h \\ h_{\mathrm{PF}} \circ h(q) \\ \mathrm{d}h_{\mathrm{PF}}|_{h(q)} \;\; \mathrm{d}h_q \, G(q)v \end{bmatrix}. \tag{3.14}$$

The last 8 components of the candidate transformation (3.14) are simply y_{PF} and $\dot{y}_{\mathrm{PF}}$ while the first component is the heading angle of the mobile base. We need two technical results before proving the main result of this section.

Lemma 3.2.2. *The function*

$$F(y_1, y_2) = \begin{bmatrix} s(y_1, y_2) \\ \varpi(y_1, y_2) \end{bmatrix}$$

maps an open subset of $\mathbb{R}^2$ *containing the curve* $\mathcal{C}$ *diffeomorphically onto its image.*

Proof. By [14, Lemma 3.1], $\mathrm{d}F_{(y_1,y_2)}$ is non-singular at each $(y_1, y_2) \in \mathcal{C}$. The result now follows from the Generalized Inverse Function Theorem [13, Exercise 3.10]. □

Lemma 3.2.3. *If* $q \in \{q \in \mathcal{Q} : |\theta_1 - \theta_h| < \pi/2,\ |\theta_2 + \theta_1 - \theta_h| < \pi/2\}$, *then* $\mathrm{d}h_q \cdot G(q) \in \mathbb{R}^{4\times 4}$ *is non-singular.*

Proof. This results follows the previously discussed fact that

$$\det\left(\mathrm{d}h_q \, G(q)\right) = -\ell_1 \cos(\theta_1 - \theta_h) - \ell_2 \cos(\theta_1 + \theta_2 - \theta_h).$$

□

Proposition 3.2.1. *Let*

$$W \coloneqq h^{-1}(U \times \mathbb{S}^1 \times \mathbb{S}^1) \cap \{q \in \mathcal{Q} : |\theta_1 - \theta_h| < \pi/2,\ |\theta_2 + \theta_1 - \theta_h| < \pi/2\}\,.$$

The function (3.14) maps the set $W \times \mathbb{R}^4 \subset \mathcal{Q} \times \mathbb{R}^4$ diffeomorphically onto its image.

Proof. We prove this result by explicitly constructing a smooth inverse to (3.14) on $W \times \mathbb{R}^4$. Let $\xi \coloneqq \begin{bmatrix}\xi_1 & \xi_2\end{bmatrix}^\top$, $\eta = \begin{bmatrix}\eta_1 & \eta_2\end{bmatrix}^\top$, $\zeta = \begin{bmatrix}\zeta_1 & \zeta_2 & \zeta_3 & \zeta_4\end{bmatrix}^\top$ and write

$$\begin{bmatrix} z \\ \xi_1 \\ \eta_1 \\ \zeta_1 \\ \zeta_2 \\ \xi_2 \\ \eta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} = T(q,v) = \begin{bmatrix} \theta_h \\ h_{\mathrm{PF}} \circ h(q) \\ \mathrm{d}h_{\mathrm{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)v \end{bmatrix}.$$

If $(z, \xi, \eta, \zeta) \in T(W, \mathbb{R}^4)$, then

$$\theta_2 = \zeta_1, \qquad \theta_h = z, \qquad \theta_1 = \arg\left(\exp(j(\zeta_2 + z))\right). \tag{3.15}$$

By construction $\xi_1 = s(h_1(q), h_2(q))$, $\eta_1 = \varpi(h_1(q), h_2(q))$ with $(h_1(q), h_2(q)) \in U$. Thus, using the notation from Lemma 3.2.2,

$$\begin{bmatrix} h_1(q) \\ h_2(q) \end{bmatrix} = F^{-1}(\xi_1, \eta_1)$$

from which we deduce, using the definition (2.29) of h, that

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = F^{-1}(\xi_1, \eta_1) - \begin{bmatrix} \ell_1 \cos(\zeta_2 + z) + \ell_2 \cos(\zeta_1 + \zeta_2 + z) \\ \ell_1 \sin(\zeta_2 + z) + \ell_2 \sin(\zeta_1 + \zeta_2 + z) \end{bmatrix}. \tag{3.16}$$

Together (3.15) and (3.16) show that we can compute $q \in W$ as a smooth function $\tilde{T}$ of $\xi_1, \eta_2, z, \zeta_1, \zeta_2$.

By Lemma 3.2.3, if $q \in W$, the decoupling matrix $\mathrm{d}h_{\mathrm{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)$ is non-singular and therefore

$$v = \Big(\mathrm{d}h_{\mathrm{PF}}|_{h(q)}\ \mathrm{d}h_q\ G(q)\Big)^{-1}_{q=\tilde{T}} \begin{bmatrix} \xi_2 \\ \eta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \tag{3.17}$$

is a smooth function on $T(W, \mathbb{R}^4)$. In conclusion, the functions (3.15), (3.16), and (3.17) define $T^{-1}(\xi, \eta, \zeta, z)$ on the set $T(W, \mathbb{R}^4)$. □

Proposition 3.2.1 together with the feedback (3.10) show that on the set $W \times \mathbb{R}^4$, the mobile manipulator with path following output (3.9) is feedback equivalent to the input-output feedback linearized system

$$\dot{z} = v_4\Big|_{T^{-1}(\xi,\eta,\zeta,z)} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \left(\mathrm{d}h_{\mathrm{PF}}|_{h(q)} \ \mathrm{d}h_q \, G(q) \right)^{-1} \begin{bmatrix} \xi_2 \\ \eta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \tag{3.18a}$$

$$\dot{\xi} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{\pitchfork} \tag{3.18b}$$

$$\dot{\eta} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \eta + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{\parallel} \tag{3.18c}$$

$$\dot{\zeta} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \zeta + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} v_{\zeta} \tag{3.18d}$$

where we have expressed the auxiliary input as

$$v_{\mathrm{aux}} \coloneqq \begin{bmatrix} v_{\pitchfork} \\ v_{\parallel} \\ v_{\zeta} \end{bmatrix}.$$

The path following output is given, in (ξ, η, ζ, z)-coordinates, by

$$y_{\mathrm{PF}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ \zeta \\ z \end{bmatrix}.$$

Returning to the problem statement from Section 3.1.3, we have that property (i) is equivalent to stabilizing $\xi = 0$, $\zeta = 0$. Property (ii) is satisfied if $\xi = 0$ is an equilibrium point of the ξ-subsystem (3.18b) and $\zeta = 0$ is an equilibrium point of the ζ-subsystem (3.18d).

The η-subsystem (3.18c) is sometimes called the **tangential subsystem** with respect to the path because it governs the portion of the manipulator's dynamics that produce

observable motion along the path. The tangential dynamics can be used to frame the desired motion along the path, property (iii), in (ξ, η, ζ, z)-coordinates. Define the tracking error

$$e_1 \coloneqq \arg\left(\exp\left(j(\eta_1 - \lambda^{\text{ref}})\right)\right). \tag{3.19}$$

Simple calculations using (3.18c) and the exosystem model (3.6) give that $\dot{e}_1 = \eta_2 - QSw$. Thus letting $e = \begin{bmatrix} e_1 & e_2 \end{bmatrix}^\top \coloneqq \begin{bmatrix} e_1 & \dot{e}_1 \end{bmatrix}^\top \in \mathbb{S}^1 \times \mathbb{R}$, the full system model becomes

$$\begin{aligned}
\dot{z} &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \left(\mathrm{d}h_{\text{PF}}|_{h(q)} \ \mathrm{d}h_q \, G(q) \right)^{-1} \begin{bmatrix} \xi_2 \\ e_2 + QSw \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \\
\dot{\xi} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v^{\pitchfork} \\
\dot{e} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v^{\parallel} - \begin{bmatrix} 0 \\ QS^2 \end{bmatrix} w \\
\dot{\zeta} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \zeta + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} v_\zeta \\
\dot{w} &= Sw \qquad \text{(exosystem)}.
\end{aligned} \tag{3.20}$$

Driving $e(t)$ to zero is equivalent to tracking the desired motion along the path and therefore meeting objective (iii).

Remark 3.2.1. *When on the path, i.e., $\xi = 0$, $\zeta = 0$ and when tracking the timing law, i.e., $e = 0$, the system's dynamics reduce to*

$$\dot{z} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \left(\mathrm{d}h_{\text{PF}}|_{h(q)} \ \mathrm{d}h_q \, G(q) \right)^{-1} \begin{bmatrix} \xi_2 \\ QSw \\ \zeta_3 \\ \zeta_4 \end{bmatrix}.$$

Since $z(t) \in \mathbb{S}^1$ and since $\mathbb{S}^1$ is compact, the internal state $z(t)$ doesn't exhibit finite escape time and the control signal produced by (3.10) is bounded whenever v_{aux} is bounded.

Remark 3.2.2. *The transformed manipulator system (3.18) also allows us to consider other types of desired motions along the path than just those described in Section 3.1.2. For example, if we want the manipulator to move along its assigned path with a particular*

speed, but there is no particular location on the path where the robot should be, then we can allow exosystems of the form

$$\dot{w}(t) = Sw(t), \qquad w(0) \in \mathbb{R}^r, \tag{3.21a}$$

$$\eta_2^{\text{ref}}(t) = Qw(t). \tag{3.21b}$$

In this case, when on the path, i.e., $\xi = 0$, $\zeta = 0$ and when tracking the desired speed, i.e., $\eta_2 = \eta_2^{\text{ref}}$, the system's dynamics reduce to

$$\dot{z} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \left(\mathrm{d}h_{\text{PF}}|_{h(q)} \; \mathrm{d}h_q \, G(q) \right)^{-1} \begin{bmatrix} \xi_2 \\ Qw \\ \zeta_3 \\ \zeta_4 \end{bmatrix}$$

$$\dot{\eta}_1 = Qw(t).$$

In this case, since $(z, \eta_1) \in \mathbb{S}^1 \times \mathbb{S}^1$, the comments from Remark 3.2.1 regarding finite escape time and boundedness also apply.

The transformed mobile manipulator system (3.18) dynamics suggest the linear control laws

$$v_{\pitchfork} = F_{\pitchfork}\xi, \tag{3.22a}$$

$$v_{\parallel} = F_{\parallel}e + QS^2w, \tag{3.22b}$$

$$v_{\zeta} = F_{\zeta}\zeta, \tag{3.22c}$$

where $F_{\pitchfork} : \mathbb{R}^2 \to \mathbb{R}$, $F_{\parallel} : \mathbb{R}^2 \to \mathbb{R}$ and $F_{\zeta} : \mathbb{R}^4 \to \mathbb{R}^2$ are chosen so that, respectively, the three linear subsystems in (3.18) are exponentially stable.

Remark 3.2.3. *When considering velocity tracking using an exosystem of the form (3.21), the tangential controller is modified to equal*

$$v_{\parallel} = F_{\parallel}(\eta_2 - \eta_2^{\text{ref}}) + QSw, \quad F_{\parallel} < 0. \tag{3.23}$$

In summary, our proposed path following controller consists of Equations (2.25), (2.26) and (2.18) which define the preliminary feedback u, Equation (3.6) which produces the desired motion along the path, Equation (3.10) which input-output feedback linearizes system (3.9), and the coordinate transformation (3.14) which is needed to implement the linear feedbacks in Equation (3.22). The block diagram of the proposed controller is shown in Figure 3.1.

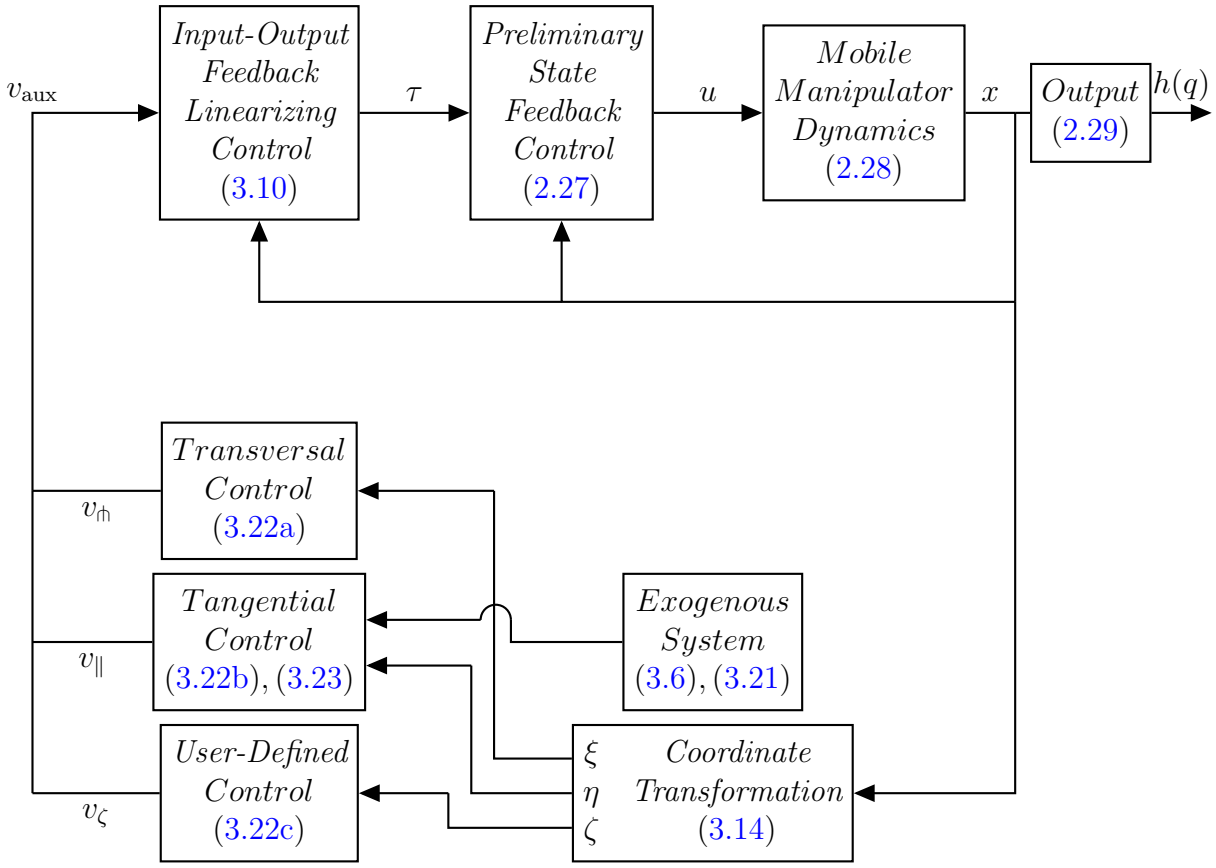

Figure 3.1: Block diagram of non-adaptive path following control.

3.3 Examples

3.3.1 Circular Path

To illustrate the general design procedure of Section 3.2, we now consider the problem of making the end-effector of the mobile manipulator system follow a circle in the counter clockwise direction. The circle has parametric representation

$$\sigma(\lambda) = r \begin{bmatrix} \cos(\lambda) \\ \sin(\lambda) \end{bmatrix} + c, \tag{3.24}$$

where $r > 0$ and $c = \begin{bmatrix} c_1 & c_2 \end{bmatrix}^\top \in \mathbb{R}^2$ are, respectively, the radius and center of the circle. In addition to (3.24), the implicit representation of the path satisfying Assumption 3.1 is

$$s(y_1, y_2) = \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - c \right\| - r, \tag{3.25}$$

with $U = \mathbb{R}^2 \backslash \{c\}$. In order to have the end-effector move in the counter clockwise direction about the circle, we take the exogenous system (3.6) to be

$$\begin{aligned} \dot{w}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} w(t), \qquad w(0) = (0, 1), \\ \lambda^{\text{ref}}(t) &= \arg\left(\exp\left(j \begin{bmatrix} 1 & 0 \end{bmatrix} w(t) \right) \right). \end{aligned}$$

To implement our proposed controller, we must define the path following output (3.8) in the case of a circular path. The implicit representation (3.4) is given by (3.25) and, in the case of a circle, the projection (3.7) can be written in closed-form as

$$\varpi(y_1, y_2) = \text{arctan2}(y_2 - c_2, y_1 - c_1). \tag{3.26}$$

Equations (3.25) and (3.26) complete the definition of the path following output (3.8) for a circular path. With the path following output in hand, it is straightforward to compute the controller given by (2.27), (3.10). The linear feedback laws in Equation (3.22) can be computed, for example, using pole-placement or LQR optimal control.

Simulation Results

To illustrate the effectiveness of our control scheme from Section 3.3.1, we present a simulation for a circle of radius $r = 1.9$ and center $c = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, where the initial conditions of

the mobile manipulator are

$$q(0) = \begin{bmatrix} 0 \\ \pi/4 \\ 1 \\ -3 \\ 0 \end{bmatrix}, \quad v(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and the gains of our linear controls (3.22) are

$$F_{\pitchfork} = \begin{bmatrix} -25 & -10 \end{bmatrix}, \quad F_{\parallel} = \begin{bmatrix} -12 & -7 \end{bmatrix}, \quad F_{\zeta} = \begin{bmatrix} -9 & -6 & 0 & 0 \\ 0 & 0 & -9 & -6 \end{bmatrix},$$

see Figure 3.2 and 3.3.

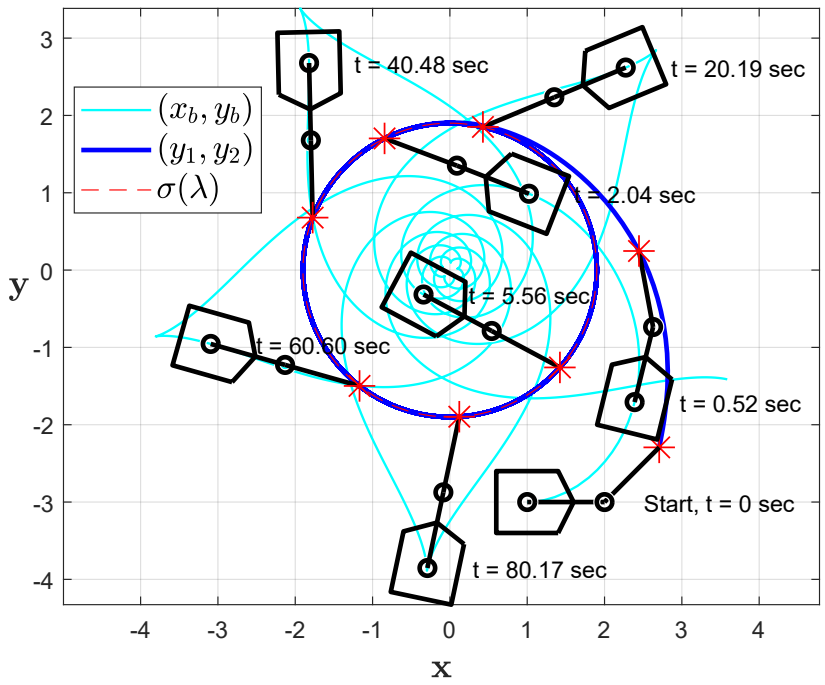

Figure 3.2: End-effector and mobile base trajectories for known circle.

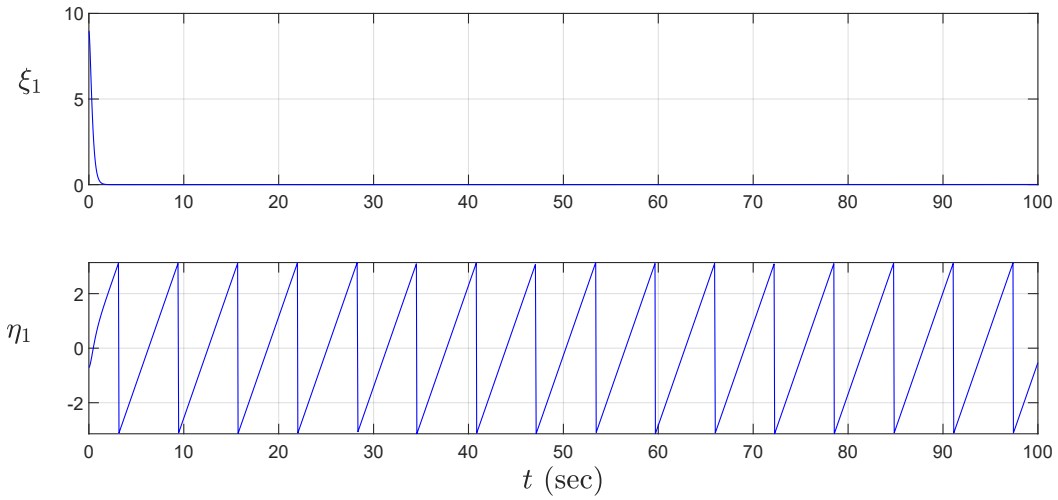

Figure 3.3: $\xi_1 = \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top \right\| - 1.9$ and $\eta_1 = \text{arctan2}(y_2, y_1)$ for known circle.

3.3.2 Elliptical Path

Next we consider the problem of making the end-effector of our robot follow an ellipse in the counter clockwise direction. We are given an ellipse $\mathcal{C}$, with parametric representation

$$\sigma(\lambda) = \begin{bmatrix} a\cos(\lambda) \\ b\sin(\lambda) \end{bmatrix} + c. \tag{3.27}$$

where $a, b > 0$ and $c = \begin{bmatrix} c_1 & c_2 \end{bmatrix}^\top \in \mathbb{R}^2$ is the center of the ellipse. In addition to (3.27), the implicit representation of the path satisfying Assumption 3.1 is

$$s(y_1, y_2) = \frac{(y_1 - c_1)^2}{a^2} + \frac{(y_2 - c_2)^2}{b^2} - 1, \tag{3.28}$$

with $U = \mathbb{R}^2 \backslash \{c\}$. In order to implement our path following controller from Section 3.2, the input-output feedback linearizing input (3.10), the coordinate transformation (3.14) and linear feedbacks (3.22) need to be computed. However, in general and for the case of the ellipse, the projection (3.7) that partially defines our path following output (3.8) does not have a closed-form solution [2]. Because of this we compute the value of $\lambda^* \in \mathbb{S}^1$ numerically using a line search algorithm over the compact set $\mathbb{S}^1$ [2] and set the first

tangential state $\eta_1 \coloneqq \lambda^*$. Therefore since, $\eta_1 = \varpi(y_1, y_2)$, to compute η_2 we have

$$\begin{aligned} \eta_2 &= \frac{\partial(\varpi \circ h)}{\partial q}\, \dot{q} = \frac{\partial(\varpi \circ h)}{\partial q}\, G(q)\, v, \\ &= \left(\frac{\partial \varpi}{\partial(y_1, y_2)}\right)\Bigg|_{\substack{y_1 = h_1(q) \\ y_2 = h_2(q)}} \begin{bmatrix} I_2 & 0_2 \end{bmatrix} \mathrm{d}h_q\, G(q)\, v. \end{aligned} \tag{3.29}$$

Simple geometric arguments give that (3.29) equals

$$\begin{aligned} \eta_2 &= (\sigma'(\lambda^*))^\top \begin{bmatrix} I_2 & 0_2 \end{bmatrix} \mathrm{d}h_q\, G(q)\, v, \\ &= \left\langle \sigma'(\lambda^*), \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} \right\rangle. \end{aligned} \tag{3.30}$$

Therefore differentiating (3.30), we get

$$\dot{\eta}_2 = \left\langle \sigma''(\lambda^*)\eta_2, \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} \right\rangle + \left\langle \sigma'(\lambda^*), \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} \right\rangle, \tag{3.31}$$

where

$$\begin{aligned} \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}\right), \\ &= \frac{\partial}{\partial q}\left(\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}\right)\dot{q} + \frac{\partial}{\partial v}\left(\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}\right)\dot{v}, \\ &= \frac{\partial}{\partial q}\left(\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}\right) G(q)v + \frac{\partial}{\partial v}\left(\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}\right)\dot{v}, \end{aligned} \tag{3.32}$$

and $\dot{v}$ is given by (2.17). The expressions (3.30), (3.31) and (3.32) provide us closed-from expressions that we can use in the computation of input-output feedback linearizing controller (3.10).

In order to have the end-effector move in the counter clockwise direction about the ellipse, we take the desired motion along the path to be generated by an exosystem of the form (3.21)

$$\begin{aligned} \dot{w}(t) &= 0, \qquad w(0) = w_0 \in \mathbb{R}, \\ \eta_2^{\mathrm{ref}}(t) &= w(t). \end{aligned} \tag{3.33}$$

If $w_0 > 0$, the desired motion is in the same direction as increasing λ in the parametrized ellipse (3.27). We apply the linear controllers from (3.22), however, cf. Remark 3.2.3, we take the tangential controller to be $v_\parallel = F_\parallel(\eta_2 - w(t))$.

Simulations Results

To illustrate the effectiveness of our control scheme from Subsection 3.3.2, we present a simulation for the ellipse

$$\sigma(\lambda) = \begin{bmatrix} 2\cos(\lambda) + 1 \\ \sin(\lambda) + 1 \end{bmatrix}.$$

with implicit representation

$$s(y_1, y_2) = \frac{(y_1 - 1)^2}{4} + (y_2 - 1)^2 - 1.$$

The robot is initialized

$$q(0) = \begin{bmatrix} 0 \\ \pi/4 \\ 2 \\ -2 \\ 0 \end{bmatrix}, \quad v(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

We take the reference signal to be $\eta_2^{\text{ref}}(t) = 1$. We use the velocity tracking controller for the tangential controller described in Remark 3.2.3 and the linear feedback matrices (3.22) are

$$F_{\pitchfork} = \begin{bmatrix} -25 & -10 \end{bmatrix}, \quad F_{\parallel} = -28, \quad F_{\zeta} = \begin{bmatrix} -9 & -6 & 0 & 0 \\ 0 & 0 & -9 & -6 \end{bmatrix},$$

the resulting motion of the end-effector and mobile base are shown in Figure 3.4 with ξ_1 and η_1 in Figure 3.5.

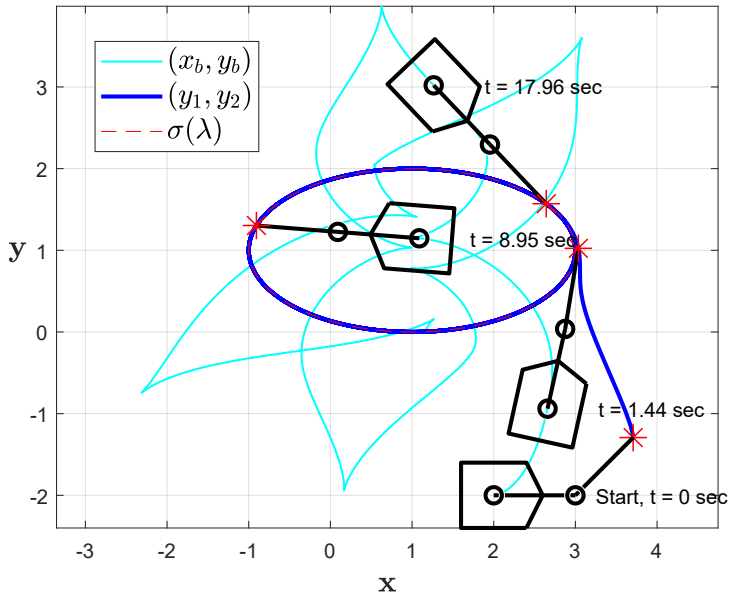

Figure 3.4: End-effector and mobile base trajectories for known ellipse.

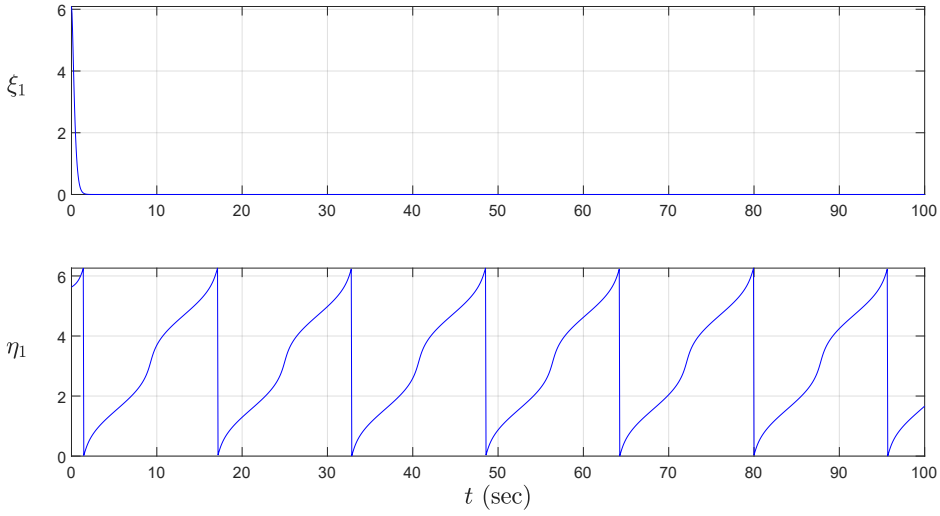

Figure 3.5: $\xi_1 = \frac{(y_1-1)^2}{4} + (y_2-1)^2 - 1$ and $\eta_1 = \lambda^*$ for known ellipse.

3.4 Circular Approximation of Closed Paths

A drawback of the proposed path following controller is that the functions $s(y_1, y_2)$ and $\varpi(y_1, y_2)$ that appear in the path following output (3.8) do not, in general, have closed-form expressions. Furthermore, even when closed-form expressions are available, the expressions in the coordinate transformation (3.14), which are needed in order to implement the linear feedback laws (3.22), can be complicated. In this subsection we propose a path following controller for arbitrary closed curves that applies the circular path following controller from Subsection 3.3.1 to the osculating circle [32] associated to the closest point of the closed curve.

Given a strictly convex curve $\mathcal{C}$ with parametric representation (3.3), its signed curvature at $\lambda \in \mathbb{S}^1$ is

$$\kappa(\lambda) = \frac{\sigma_1''(\lambda)\sigma_2'(\lambda) - \sigma_2''(\lambda)\sigma_1'(\lambda)}{\|\sigma'(\lambda)\|^3}. \tag{3.34}$$

The centre of curvature $\epsilon(\lambda)$ of σ at the point $\sigma(\lambda)$ is defined to be

$$\epsilon(\lambda) = \sigma(\lambda) + \frac{1}{\kappa(\lambda)} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \frac{\sigma'(\lambda)}{\|\sigma'(\lambda)\|}. \tag{3.35}$$

Since $\mathcal{C}$ is strictly convex $(\forall \lambda \in \mathbb{S}^1)\ \kappa(\lambda^*) > 0$. The circle with centre $\epsilon(\lambda)$ and radius $1/|\kappa(\lambda)|$ is called the osculating circle to σ at the point $\sigma(\lambda)$. It is the unique circle which is tangent to σ at $\sigma(\lambda)$ and has the same (unsigned) curvature as σ at that point.

Therefore, to each $\lambda \in \mathbb{S}^1$, we can uniquely associate a circle in the plane given parametrically by

$$\sigma_\lambda(s) \coloneqq \frac{1}{|\kappa(\lambda)|} \begin{bmatrix} \cos(s) \\ \sin(s) \end{bmatrix} + \epsilon(\lambda) \tag{3.36}$$

and implicitly by

$$s_\lambda(y_1, y_2) \coloneqq \| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \epsilon(\lambda)\| - \frac{1}{|\kappa(\lambda)|}. \tag{3.37}$$

The idea proposed in this section is to first compute the parameter $\lambda^\star$ returned by the function (3.7) for the *given* curve $\mathcal{C}$. As mentioned earlier, in general the function $\varpi(y_1, y_2)$ doesn't have a closed-form expression; therefore the calculation of $\lambda^* \in \mathbb{S}^1$ will normally be done numerically and it can be done efficiently using a line search algorithm over the compact set $\mathbb{S}^1$. The parameter $\lambda^\star(t) = \varpi(y_1(t), y_2(t))$ defines a circle with parametric representation (3.36) and implicit representation (3.37). To this circle we associate a path

following output

$$y_{\mathrm{PF}} = h_{\mathrm{PF}}(y) = \begin{bmatrix} \| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \epsilon(\lambda^\star)\| - \frac{1}{|\kappa(\lambda^\star)|} \\ \arctan2(y_2 - \epsilon_2(\lambda^\star), y_1 - \epsilon_1(\lambda^\star)) \\ y_3 \\ y_4 \end{bmatrix}. \tag{3.38}$$

In (3.38), $\lambda^\star$ is a function of (y_1, y_2) but in applying the circular path following controller from Subsection 3.3.1 we neglect this fact. More specifically, when we compute the differential of h_{PF} we treat κ and ϵ as being constant. This results in a simplified expression for the input-output feedback linearizing controller (3.10) and the coordinate transformation (3.14).

We also modify the definition of the desired motion along the path because the second component in (3.38) doesn't return a value for the parameter of the actual path σ. In particular, at each moment in time, $y_{\mathrm{PF},2}(t)/|\kappa(\lambda^\star(t))|$ equals the arc-length along the osculating circle from $\sigma_{\lambda^\star}(0)$ to $\sigma_{\lambda^\star}(y_{\mathrm{PF},2}(t))$. Thus, in order to get close to unit-speed traversal along the path, we take the reference for $\eta_2 \approx \dot{y}_{\mathrm{PF},2}$ to be $\eta_2^{\mathrm{ref}}(t) = \kappa(\lambda^\star(t))$. Therefore, the tangential linear control is set to

$$v_{\|} = F_{\|}(\eta_2 - \kappa(\lambda^\star(t))), \tag{3.39}$$

the block diagram of the proposed controller is shown in Figure 3.6.

3.4.1 Simulation Results

To illustrate the effectiveness of our control scheme from Section 3.4, we present a simulation using the circle approximation approach for the ellipse

$$\sigma(\lambda) = \begin{bmatrix} 2\cos(\lambda) + 1 \\ \sin(\lambda) + 1 \end{bmatrix}. \tag{3.40}$$

with implicit representation

$$s(y_1, y_2) = \frac{(y_1 - 1)^2}{4} + (y_2 - 1)^2 - 1.$$

The robot is initialized as

$$q(0) = \begin{bmatrix} 0 \\ \pi/4 \\ 2 \\ -2 \\ 0 \end{bmatrix}, \quad v(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

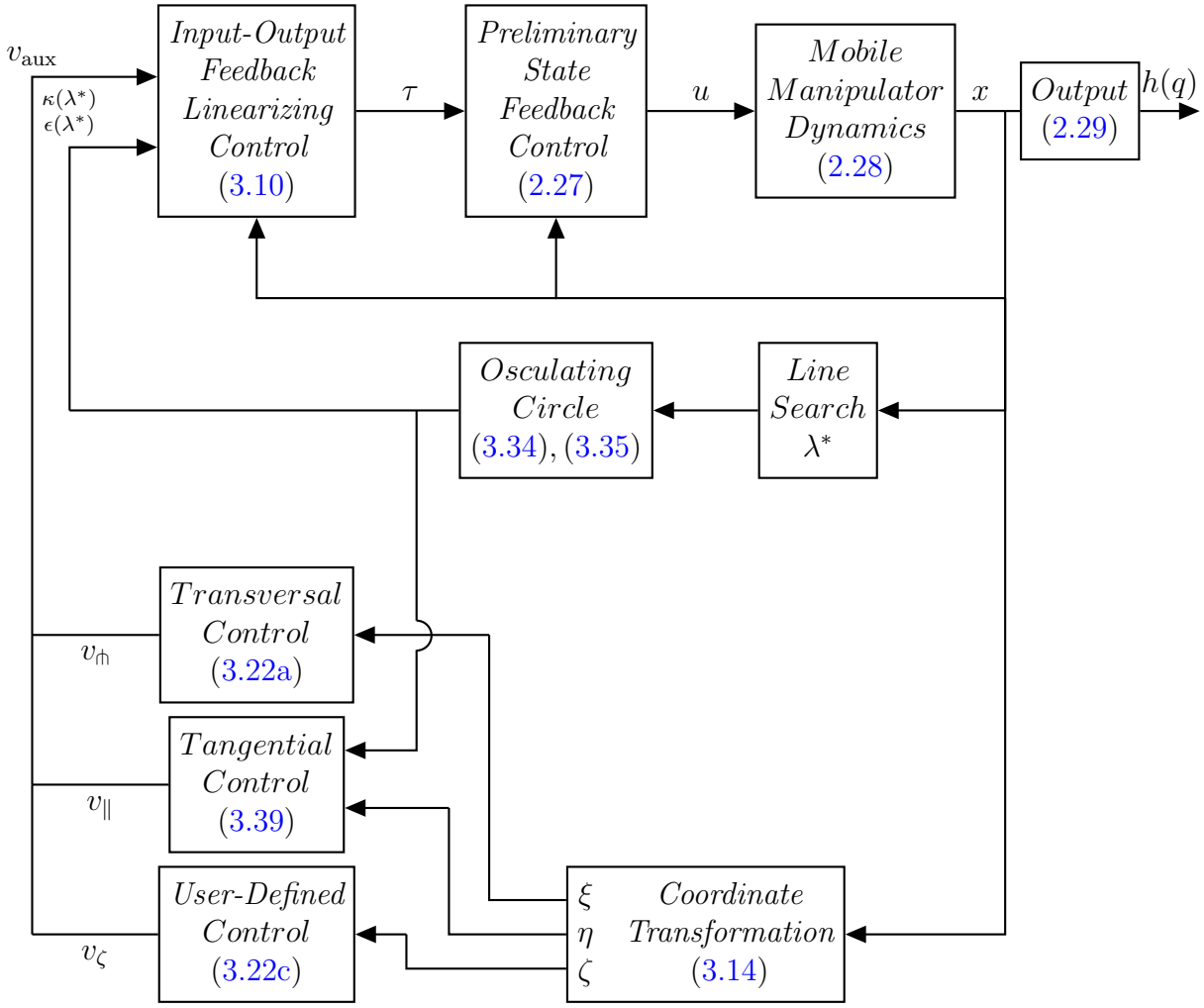

Figure 3.6: Block diagram of non-adaptive path following control using circular approximation.

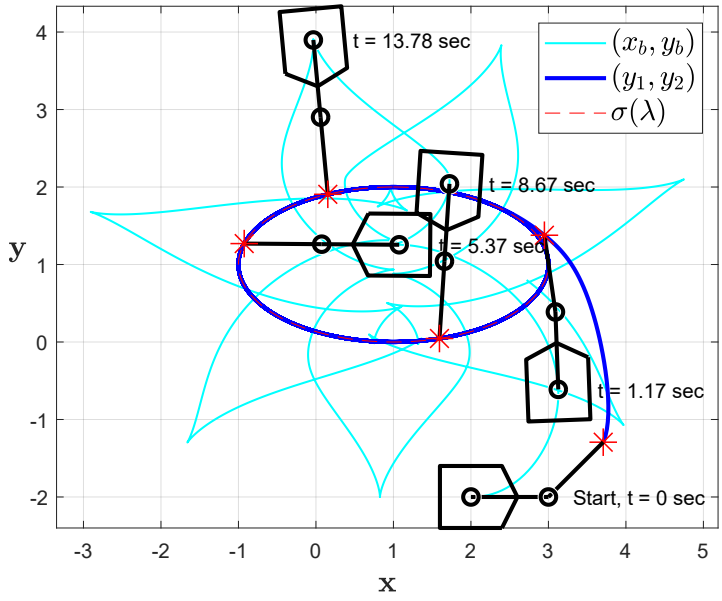

Figure 3.7: End-effector and mobile base trajectories for known osculating circle.

We take the reference signal to be $\eta_2^{\text{ref}}(t) = \kappa(\lambda^\star(t))$. We use the velocity tracking controller for the tangential controller described in Remark 3.2.3 and the linear feedback matrices (3.22) are

$$F_{\text{⫛}} = \begin{bmatrix} -25 & -10 \end{bmatrix}, \quad F_{\parallel} = -28, \quad F_{\zeta} = \begin{bmatrix} -9 & -6 & 0 & 0 \\ 0 & 0 & -9 & -6 \end{bmatrix}, \tag{3.41}$$

the resulting motion of the end-effector and mobile base are shown in Figure 3.7 with ξ_1 and the arc-length in Figure 3.8.

3.4.2 Comparison

Given the path following controllers from Subsection 3.3.2 and Section 3.4, we simulated for the ellipse from (3.40), with linear feedback gains (3.41) and initial conditions as listed in Table 3.1 and computed the integral of the error of distance to path, defined as follows

$$e_{\text{PF}} = \int_0^T \left\| \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix}^\top - \sigma(\lambda^\star(t)) \right\| \mathrm{d}t, \tag{3.42}$$

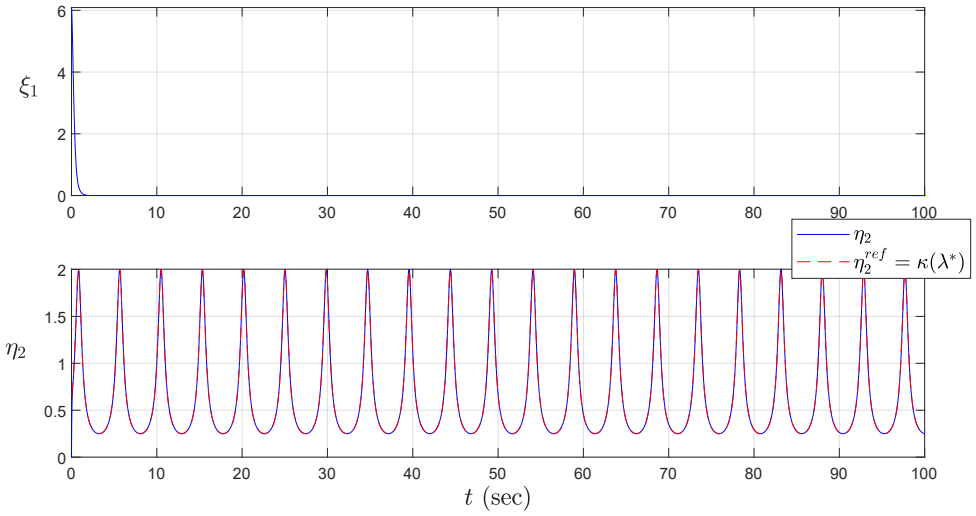

Figure 3.8: $\xi_1 = \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \epsilon(\lambda^\star) \right\| - \frac{1}{|\kappa(\lambda^\star)|}$ and η_2 tracking $\kappa(\lambda^\star)$.

to compare the difference in accuracy between our two proposed solutions, given that one solution uses an local approximation of the path.

Table 3.1: Comparison of the integral of error of distance to path (3.42) for path following simulation of ellipse (3.40).

Initial conditions									PF1	PF2
$\theta_1(0)$	$\theta_2(0)$	$x_b(0)$	$y_b(0)$	$\theta_h(0)$	$\dot{\theta}_1(0)$	$\dot{\theta}_2(0)$	v(0)	$\dot{\theta}_h(0)$	(3.42)	(3.42)
0	0	-1	-0.1	0	0	0	0	0	0.6825	0.6745
0	0	-1	2.1	0	0	0	0	0	0.6822	0.6747
$\pi/2$	0	3.1	-1	$\pi/2$	0	0	0	0	0.6221	0.6945
$\pi/2$	0	-1.1	-1	$\pi/2$	0	0	0	0	0.6228	0.6946
π	0	3	3	π	0	0	0	0	7.5958	6.9519

In Table 3.1, entry PF1 references our proposed solution from Subsection 3.3.1, while PF2 references controller from Section 3.4. From Table 3.1 we can observe that the difference in error of distance to path between the two control schemes is quite negligible and that depending on the initial conditions that the robot is initialized in, one controller may perform better than the other.

Chapter 4

Adaptive Path Following for Circular and Strictly Convex Paths

In this chapter we ease the assumption that we are given a *known* path from Chapter 3 and present two solutions for the design of path following controllers that adapt to *unknown* paths. Specifically, we design an indirect adaptive path following controller that achieves control objectives from Problem 2. Thereafter, we extend these results to *unknown*, strictly convex[1], closed paths by again designing an indirect adaptive path following controller that also achieves control objectives from Problem 2.

4.1 Circular Path Parameter Estimation

Consider an unknown circular path $\mathcal{C}$ with radius $r \in \mathbb{R}$ and origin $c \in \mathbb{R}^2$. We start by describing a method to estimate r and c given an agent whose location $p(t) \in \mathbb{R}^2$ and velocity $\dot{p}(t)$ are known for all t.

Assumption 4.1. The agent's trajectory $p(t) \in \mathbb{R}^2$ is a smooth function of t and the quantities

$$\xi_1(t) \coloneqq ||p(t) - c|| - r, \tag{4.1}$$

$$\xi_2(t) \coloneqq \dot{\xi}_1(t). \tag{4.2}$$

are measurable. ♦

[1] A curve σ is strictly convex, if and only if, $(\forall \lambda \in \mathbb{S}^1)$ it's signed curvature is always positive, i.e., $\kappa(\lambda) > 0$.

Here $\xi_1(t)$ is the distance from $p(t)$ to the closest point on the circle $\mathcal{C}$, and $\xi_2(t)$ is the rate of change of this distance at time t.

4.1.1 Parametric Model

Under Assumption 4.1, we now derive a parametric model for our circular path parameter estimation algorithm. Differentiating the identity

$$(\xi_1(t) + r)^2 = (p(t) - c)^\top (p(t) - c) \tag{4.3}$$

with respect to time

$$\frac{\mathrm{d}}{\mathrm{d}t} (\xi_1(t) + r)^2 = 2 (\xi_1(t) + r) \xi_2(t) = 2 (p(t) - c)^\top \dot{p}(t) \tag{4.4}$$

and re-arranging (4.4) we get

$$p^\top(t)\dot{p}(t) - \xi_1(t)\xi_2(t) = r\xi_2(t) + c^\top \dot{p}(t). \tag{4.5}$$

Since the only unknowns in (4.5) are r and c, it is in **linear parametric model form** [18]. Define

$$\begin{aligned} \mu(t) &\coloneqq p^\top(t)\dot{p}(t) - \xi_1(t)\xi_2(t) \in \mathbb{R}, \\ \Omega^\star &\coloneqq \begin{bmatrix} r \\ c \end{bmatrix} \in \mathbb{R}^3, \\ \phi(t) &\coloneqq \begin{bmatrix} \xi_2(t) \\ \dot{p}(t) \end{bmatrix} \in \mathbb{R}^3, \end{aligned} \tag{4.6}$$

so that (4.5) can be expressed as

$$\mu(t) = \Omega^{\star\top} \phi(t). \tag{4.7}$$

4.1.2 Estimation Model and Estimation Error

The estimation model has the same form as the linear parametric model (4.7), with the only exception that the unknown parameters $\Omega^\star$ are replaced with their estimates at time t. These estimates will be denoted by $\Omega(t)$ and the estimation model will be defined as follows

$$\hat{\mu}(t) \coloneqq \Omega^\top(t)\phi(t). \tag{4.8}$$

Given (4.7) and (4.8), observe that if $\Omega(t) \to \Omega^\star$ as $t \to \infty$, then $\hat{\mu}(t) \to \mu(t)$; the converse is false. With this in mind, define the estimation error to be

$$\varepsilon(t) \coloneqq \frac{\mu(t) - \hat{\mu}(t)}{1 + \beta\|\phi(t)\|^2}, \qquad \beta > 0. \tag{4.9}$$

Equation (4.9) essentially defines a signal that indirectly reflects the difference between $\Omega(t)$ and $\Omega^\star$.

4.1.3 Adaptive Law

Using (4.8) and (4.9) we can now derive an adaptive law to update the estimate $\Omega(t)$, first we define an instantaneous cost criterion [18] for the estimation error (4.9) as

$$J(\Omega) \coloneqq \frac{\varepsilon^2(t)}{2} = \frac{1}{2}\frac{(\mu(t) - \Omega^\top\phi(t))^2}{1 + \beta\|\phi(t)\|^2}. \tag{4.10}$$

The approach is to update $\Omega(t)$ such that (4.10) is minimized so that $\varepsilon(t) \to 0$ as $t \to \infty$; we use gradient descent

$$\dot{\Omega}(t) = -\Gamma\nabla J(\Omega) \tag{4.11}$$

where $\Gamma = \Gamma^\top \in \mathbb{R}^{3\times3}$ is a positive definite scaling matrix called the adaptive gain and $\nabla J(\Omega)$ is the gradient of (4.10) with respect to Ω

$$\nabla J(\Omega) = (\mathrm{d}J_\Gamma)^\top = -\varepsilon(t)\phi(t). \tag{4.12}$$

Substituting (4.12) into (4.11) leads to the adaptive law

$$\dot{\Omega}(t) = \Gamma\varepsilon(t)\phi(t), \quad \Omega(0) = \Omega_0, \tag{4.13}$$

for updating the estimate $\Omega(t)$ starting from an arbitrary initial estimate $\Omega(0) = \Omega_0$.

The algorithm (4.13) ensures that estimation error $\varepsilon(t) \to 0$ as $t \to \infty$, but it does not necessarily guarantee that $\Omega(t) \to \Omega^\star$ with time. However in the next sections we will observe that such law will prove to be enough to satisfy control objectives from Problem 2 in an indirect adaptive path following control approach.

4.2 Adaptive Path Following of Circular Paths

Returning to the path following problem, consider a circle $\mathcal{C}$ in the inertial frame $\mathbb{O}$, represented parametrically by (3.24), where r and c are unknown. To apply the estimation algorithm from Section 4.1 the role of $p(t)$ will be played by the mobile manipulator's end-effector location, i.e., the first two components of the output (2.29),

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_b + \ell_1 \cos(\theta_1) + \ell_2 \cos(\theta_1 + \theta_2) \\ y_b + \ell_1 \sin(\theta_1) + \ell_2 \sin(\theta_1 + \theta_2) \end{bmatrix}. \tag{4.14}$$

Note from (4.14) that the agent's trajectory $p(t)$ will be a function of the evolution of the configuration variables $q \in \mathcal{Q}$, under the system's partially compensated dynamics (3.1). Furthermore, since we've assumed that q and v are measurable, $\dot{p}(t)$ is also available

$$\begin{aligned} \dot{p}(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathrm{d}h_q \, G(q) \, v, \\ &= \begin{bmatrix} -\left(\ell_1 \sin\left(\theta_1\right) + \ell_2 \sin\left(\theta_1 + \theta_2\right)\right) \dot{\theta}_1 - \ell_2 \sin\left(\theta_1 + \theta_2\right) \dot{\theta}_2 + \mathrm{v} \cos\left(\theta\right) \\ \left(\ell_1 \cos\left(\theta_1\right) + \ell_2 \cos\left(\theta_1 + \theta_2\right)\right) \dot{\theta}_1 + \ell_2 \cos\left(\theta_1 + \theta_2\right) \dot{\theta}_2 + \mathrm{v} \sin\left(\theta\right) \end{bmatrix}, \end{aligned} \tag{4.15}$$

with both (4.14) and (4.15), we also assume that sensors provide the signals (4.1) and (4.2). Therefore, the design of the so called indirect adaptive path following controller for circular paths, in short, will consist of a combination of our circular path parameter estimator from Section 4.1 and a modified version of our path following control from Subsection 3.3.1.

4.2.1 Path Following Output and Control Design

The basic idea is to use the path following controller from Subsection 3.3.1 but with an estimate of the circle's radius r and centre c. We run the adaptive law described by (4.6), (4.8), (4.9) and (4.13). The path following output is taken to be

$$y_{\mathrm{PF}} = h_{\mathrm{PF}}(y, t) := \begin{bmatrix} \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \hat{c}(t) \right\| - \hat{r}(t) \\ \operatorname{arctan2}(y_2 - \hat{c}_2(t), y_1 - \hat{c}_1(t)) \\ y_3 \\ y_4 \end{bmatrix}.$$

As in Section 3.4, we neglect the time-varying nature of h_{PF}. Specifically, when we compute the differential $\mathrm{d}h_{\mathrm{PF}}$, the estimates $\hat{r}(t)$ and $\hat{c}(t)$ are treated as constants. This results in

a simplified expression for the input-output feedback linearizing controller (3.10) and for the tangential η states in the coordinate transformation (3.14).

In order to satisfy property (ii) in Problem 2, we take the desired motion along the path to be generated by an exosystem of the form (3.33). If $w_0 > 0$, the desired motion is in the same direction as increasing λ in the parameterized circle (3.24). We apply the linear controllers (3.22) however, cf. Remark 3.2.3, we take the tangential controller to be

$$v_{\parallel} = F_{\parallel}(\eta_2 - \eta_2^{\text{ref}}). \tag{4.16}$$

We emphasize that in the linear feedback law $v_{\pitchfork}$ from (3.22), the true values of ξ (see Assumption 4.1) are used. Refer to Figure 4.1 for a block diagram of the control scheme presented in this section.

4.2.2 Simulation Results

To illustrate the effectiveness of our adaptive path following controller for circular paths, we simulate for an unknown target circle of radius $r = 1.9$ and origin $c = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$. The robot and adaptive law are initialized respectively at

$$q(0) = \begin{bmatrix} 0 \\ \pi/4 \\ 3 \\ 3 \\ 0 \end{bmatrix}, \quad v(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Omega(0) = \begin{bmatrix} 1 \\ 0.2 \\ 0 \end{bmatrix}.$$

with $\beta = 0.8$, $\Gamma = 20 I_3$, we again use a velocity tracking controller for the tangential control and we take the reference signal to be $\eta_2^{\text{ref}}(t) = 0.5$ and the linear feedback matrices (3.22) are

$$F_{\pitchfork} = \begin{bmatrix} -81 & -18 \end{bmatrix}, \quad F_{\parallel} = -28, \quad F_{\zeta} = \begin{bmatrix} -9 & -6 & 0 & 0 \\ 0 & 0 & -9 & -6 \end{bmatrix}.$$

Figures 4.2, 4.3 and 4.4 reveal that while $\hat{c}(t)$ converges to c, $\hat{r}(t)$ does not converge to r. Furthermore, even though we assumed our estimated circular path parameters $\hat{r}(t)$ and $\hat{c}(t)$ to be constant in our control design our adaptive algorithm still achieves control objectives of Problem 2.

4.3 Adaptive Path Following of Strictly Convex Paths

Next, we extend the adaptive path following controller of Section 4.2 to a more general class of paths. Given an *unknown* curve $\mathcal{C}$ with parametric representation (3.3) that is

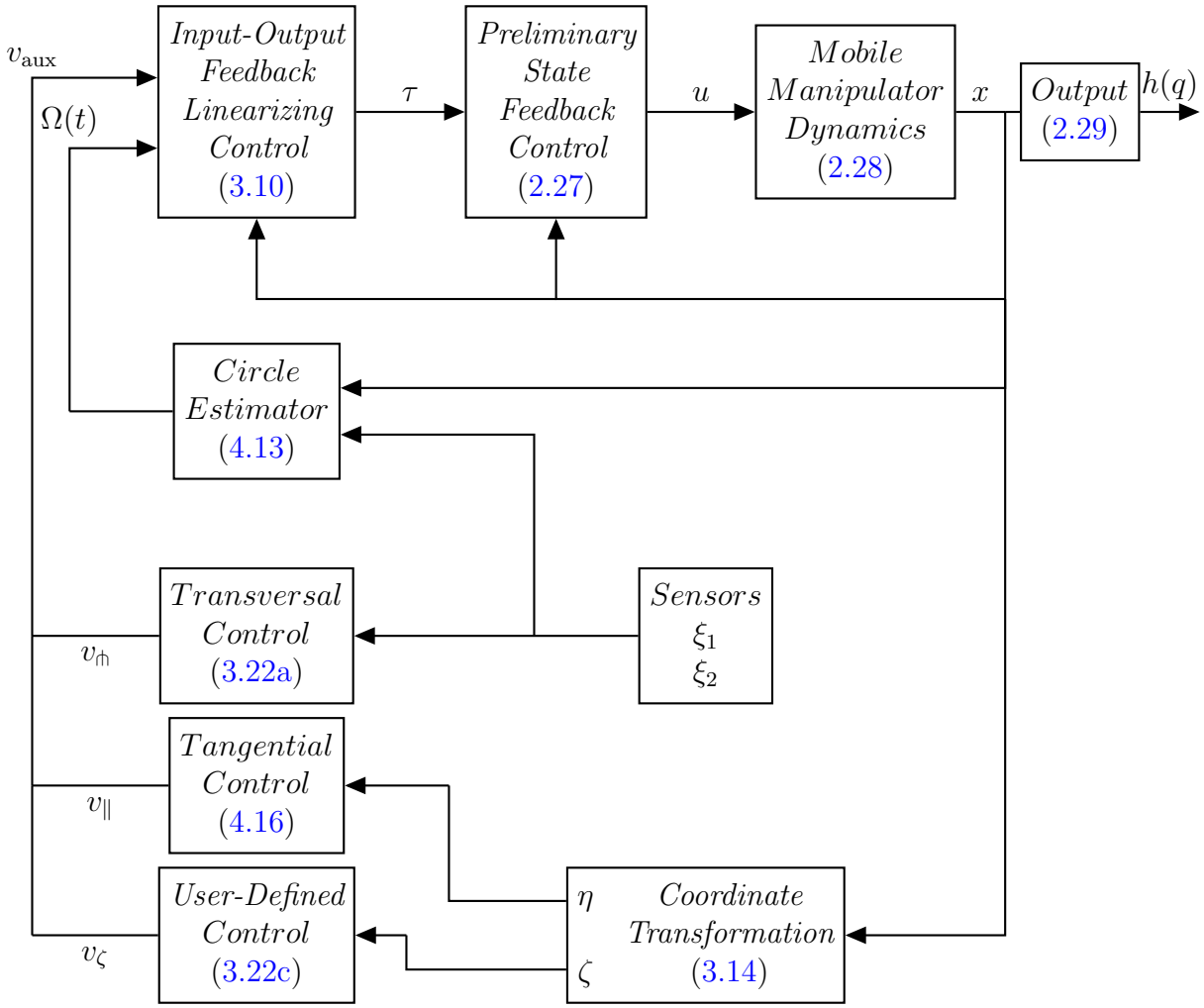

Figure 4.1: Block diagram of indirect adaptive path following controller for unknown circular paths.

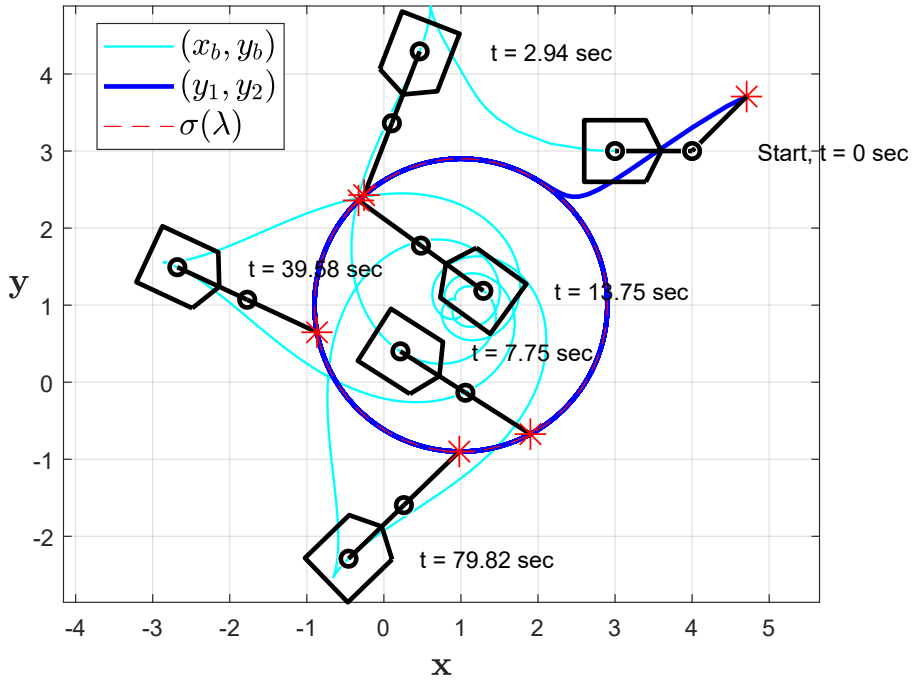

Figure 4.2: End-effector and mobile base trajectories for unknown circle.

strictly convex, i.e., its signed curvature (3.34) is strictly positive, we make the following assumption.

Assumption 4.2. Sensors provide the values $s(y_1(t), y_2(t))$ from (3.8) and $\dot{s}(y_1(t), y_2(t))$ for the unknown, strictly convex, path $\mathcal{C}$. ⧫

Assumption 4.2 essentially asks that a signed distance $s(y_1(t), y_2(t))$ from the mobile manipulator's end-effector to $\mathcal{C}$ be available for feedback, as well as its time-derivative $\dot{s}$. Under Assumption 4.2, and following the discussion from Section 3.4, to each $\lambda^\star(t) = \varpi(y_1(t), y_2(t))$ we associate the osculating circle to $\mathcal{C}$ at the point $\sigma(\lambda^\star(t))$. It has a parametric representation $\sigma_{\lambda^\star(t)}(s)$ given by (3.36) and an implicit representation $s_{\lambda^\star(t)}(y_1, y_2)$ given by (3.37) where the curvature $\kappa(\lambda^\star(t))$ and centre of curvature $\epsilon(\lambda^\star(t))$ are unknown. Let $r(\lambda) := 1/|\kappa(\lambda)|$ denote the radius of the osculating circle at $\sigma(\lambda)$.

Under Assumption 4.2, we equate the value of $s(y_1(t), y_2(t))$ to the distance of the

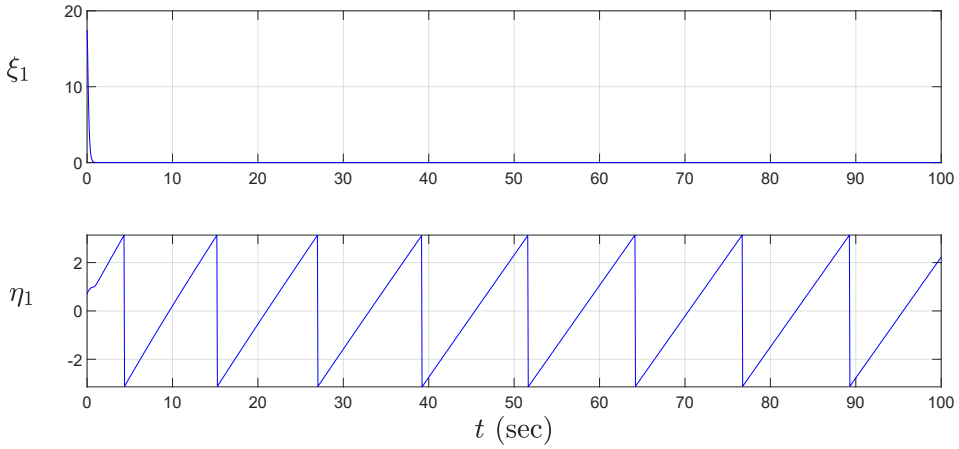

Figure 4.3: $\xi_1 = \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \hat{c}(t) \right\| - \hat{r}(t)$ and $\eta_1 = \arctan2(y_2 - \hat{c}_2(t), y_1 - \hat{c}_1(t))$ for unknown circle.

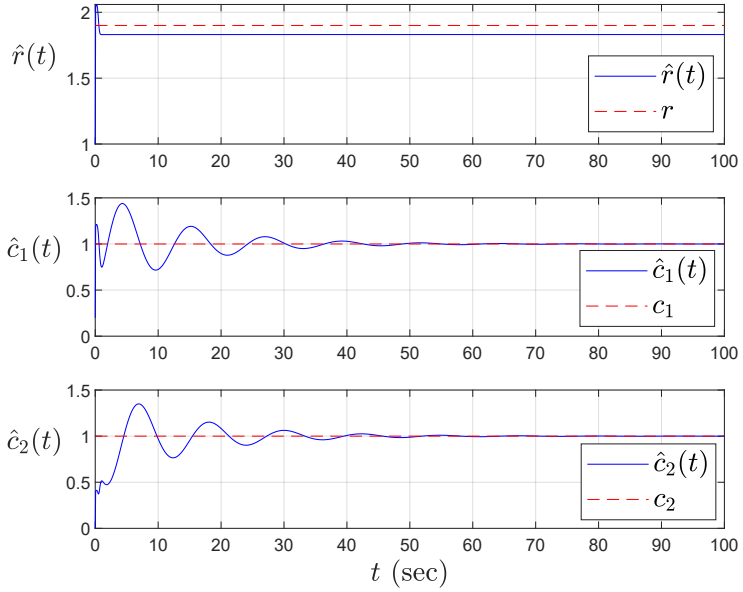

Figure 4.4: Circular path parameter convergence from adaptive law.

end-effector to the osculating circle

$$\xi_1(t) \coloneqq \left\| \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix}^\top - \epsilon(\lambda^\star(t)) \right\| - r(\lambda^\star(t)).$$

Similarly we identify $\dot{s}(y_1(t), y_2(t))$ to equal the rate of change of the distance to the osculating circle

$$\xi_2 \coloneqq \frac{\left(\begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix}^\top - \epsilon(\lambda^\star(t)) \right)^\top}{\left\| \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix}^\top - \epsilon(\lambda^\star(t)) \right\|} \left(\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} - \dot{\epsilon}(\lambda^\star(t)) \right) - \dot{r}(\lambda^\star(t)), \tag{4.17}$$

not the curve $\mathcal{C}$. Next we attempt to apply the parameter estimator from Section 4.1 keeping in mind that in the current scenario the unknown parameters $\Omega^\star(t)$ are no longer constant. Given the above ξ_1 and ξ_2 we proceed to derive the new linear parametric model by considering the same expression (4.3)

$$(\xi_1(t) + r(\lambda^\star))^2 = (p(t) - \epsilon(\lambda^\star))^\top (p(t) - \epsilon(\lambda^\star))$$

differentiating the left side of (4.3) with respect to time (we will drop the t and $\lambda^\star$ parameters in the interest of space), we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}(\xi_1 + r)^2 &= 2(\xi_1 + r)(\dot{\xi}_1 + \dot{r}) \\ &= 2(\xi_1 + r) \left(\xi_2 + \left. \frac{\mathrm{d}r(\lambda)}{\mathrm{d}\lambda^\star} \right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star}{\mathrm{d}t} \right) \\ &= 2 \left\| p - \epsilon \right\| \ \xi_2 + \underbrace{2 \left\| p - \epsilon \right\| \ \left. \frac{\mathrm{d}r(\lambda)}{\mathrm{d}\lambda^\star} \right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star}{\mathrm{d}t}}_{\coloneqq \varepsilon_1} \\ &= 2 \left\| p - \epsilon \right\| \ \xi_2 + \varepsilon_1 \\ &= 2(\xi_1 + r) \ \xi_2 + \varepsilon_1 \end{aligned} \tag{4.18}$$

substituting (4.17) into (4.18) and rearranging

$$\begin{aligned} 2(\xi_1 + r)\xi_2 + \varepsilon_1 = & 2 \left\| p - \epsilon \right\| \quad \dots \\ & \left(\frac{(p - \epsilon)^\top}{\left\| p - \epsilon \right\|} \dot{p} - \frac{(p - \epsilon)^\top}{\left\| p - \epsilon \right\|} \left. \frac{\mathrm{d}\epsilon(\lambda)}{\mathrm{d}\lambda} \right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star(t)}{\mathrm{d}t} - \left. \frac{\mathrm{d}r(\lambda)}{\mathrm{d}\lambda^\star} \right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star}{\mathrm{d}t} \right) + \varepsilon_1 \end{aligned}$$

$$
\begin{aligned}
=&2(p-\epsilon)^\top \dot{p} - \underbrace{2(p-\epsilon)^\top \left.\frac{\mathrm{d}\epsilon(\lambda)}{\mathrm{d}\lambda}\right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star(t)}{\mathrm{d}t}}_{:=\varepsilon_2} \quad \dots \\
&\underbrace{-2\left\|p-\epsilon\right\| \left.\frac{\mathrm{d}r(\lambda)}{\mathrm{d}\lambda^\star}\right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star}{\mathrm{d}t}}_{=\varepsilon_1} + \varepsilon_1 \\
=&2(p-\epsilon)^\top \dot{p} - \varepsilon_2. \qquad (4.19)
\end{aligned}
$$

Therefore from (4.18) and (4.19) we obtain

$$
2(\xi_1(t) + r(\lambda^\star))\xi_2(t) + \varepsilon_1(t) = 2(p(t) - \epsilon(\lambda^\star))^\top \dot{p}(t) - \varepsilon_2(t) \qquad (4.20)
$$

further rearranging (4.20), we get

$$
p^\top(t)\dot{p}(t) - \xi_1(t)\xi_2(t) = r(\lambda^\star)\xi_2(t) + \epsilon^\top(\lambda^\star)\dot{p}(t) + \varepsilon_1(t) + \varepsilon_2(t). \qquad (4.21)
$$

From (4.21) we are led to the linear parametric model (cf. (4.7))

$$
\mu(t) = \Omega^{\star\top}(t)\phi(t) + \varepsilon_1(t) + \varepsilon_2(t) \qquad (4.22)
$$

where

$$
\begin{aligned}
\mu(t) &\coloneqq \begin{bmatrix} y_1(t) & y_2(t) \end{bmatrix} \begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} - \xi_1(t)\xi_2(t) \in \mathbb{R}, \\
\Omega^\star(t) &\coloneqq \begin{bmatrix} r(\lambda^\star(t)) \\ \epsilon(\lambda^\star(t)) \end{bmatrix} \in \mathbb{R}^3, \\
\phi(t) &\coloneqq \begin{bmatrix} \xi_2(t) \\ \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} \in \mathbb{R}^3.
\end{aligned}
$$

Again the perturbations $\varepsilon_1(t)$ and $\varepsilon_2(t)$ in (4.22) are given by

$$
\varepsilon_1(t) = 2\left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \epsilon(\lambda^\star(t)) \right\| \left.\frac{\mathrm{d}r(\lambda)}{\mathrm{d}\lambda}\right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star(t)}{\mathrm{d}t}, \qquad (4.23)
$$

$$
\varepsilon_2(t) = 2\left(\begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \epsilon(\lambda^\star(t)) \right)^\top \left.\frac{\mathrm{d}\epsilon(\lambda)}{\mathrm{d}\lambda}\right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star(t)}{\mathrm{d}t}. \qquad (4.24)
$$

The expressions (4.23), (4.24) show that if $r(\lambda^\star(t))$ and $\epsilon(\lambda^\star(t))$ are slowly time-varying, then $\varepsilon_1(t) \approx 0$, $\varepsilon_2(t) \approx 0$ and it is reasonable to expect that the estimation algorithm from

Section 4.1 will still work. In this context, slow time variation means that the curve $\mathcal{C}$ has almost constant curvature $r'(\lambda) \approx 0$ and that the closest point on $\mathcal{C}$ to the robot's end-effector doesn't change too quickly, $\dot{\lambda}^\star(t) \approx 0$. In other words, $\mathcal{C}$ can be slightly deformed from a circle and the robot shouldn't move too fast. The estimation algorithm for $\Omega(t) \coloneqq [\hat{r}(t) \quad \hat{\epsilon}^\top(t)]^\top$ is therefore

$$\dot{\Omega}(t) = \Gamma \left(\frac{\mu(t) - \Omega^\top(t)\phi(t)}{1 + \beta \|\phi(t)\|^2} \right) \phi(t), \quad \Omega(0) = \Omega_0, \quad \beta > 0. \tag{4.25}$$

4.3.1 Path Following Output and Control Design

With equation 4.25 as our osculating circle parameter estimator, we proceed to use the same path following controller proposed in Section 4.2. The path following output is taken to be

$$y_{\text{PF}} = h_{\text{PF}}(y, t) \coloneqq \begin{bmatrix} \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \hat{\epsilon}(t) \right\| - \hat{r}(t) \\ \operatorname{arctan2}(y_2 - \hat{\epsilon}_2(t), y_1 - \hat{\epsilon}_1(t)) \\ y_3 \\ y_4 \end{bmatrix}.$$

As in Section 4.2, we neglect the time-varying nature of h_{PF}. Specifically, when we compute the differential $\mathrm{d}h_{\text{PF}}$, the estimates $\hat{r}(t)$ and $\hat{\epsilon}(t)$ are treated as constants. This results in the same simplifications to the path following controller discussed in Section 4.2 for the input-output feedback linearizing controller (3.10) and for the tangential η states in the coordinate transformation (3.14).

To enforce that the closest-point on the path $\mathcal{C}$ to the end-effector does not change too rapidly, we choose the reference for the tangential velocity η_2 in much the same way as in Section 3.4. Specifically, we take $\eta_2^{\text{ref}}(t) = c_\parallel \hat{\kappa}(t)$ where $|c_\parallel|$ is chosen to be small. We apply the linear controllers (3.22) however, since we are doing velocity tracking in the tangential subsystem, we apply the tangential controller

$$v_\parallel = F_\parallel(\eta_2 - c_\parallel \hat{\kappa}(t)). \tag{4.26}$$

We emphasize that in the linear feedback law $v_{\pitchfork}$ from (3.22), the values of s and $\dot{s}$ from Assumption 4.2 are used in place of ξ_1 and ξ_2. Refer to Figure 4.5 for a block diagram of the control scheme presented in this section.

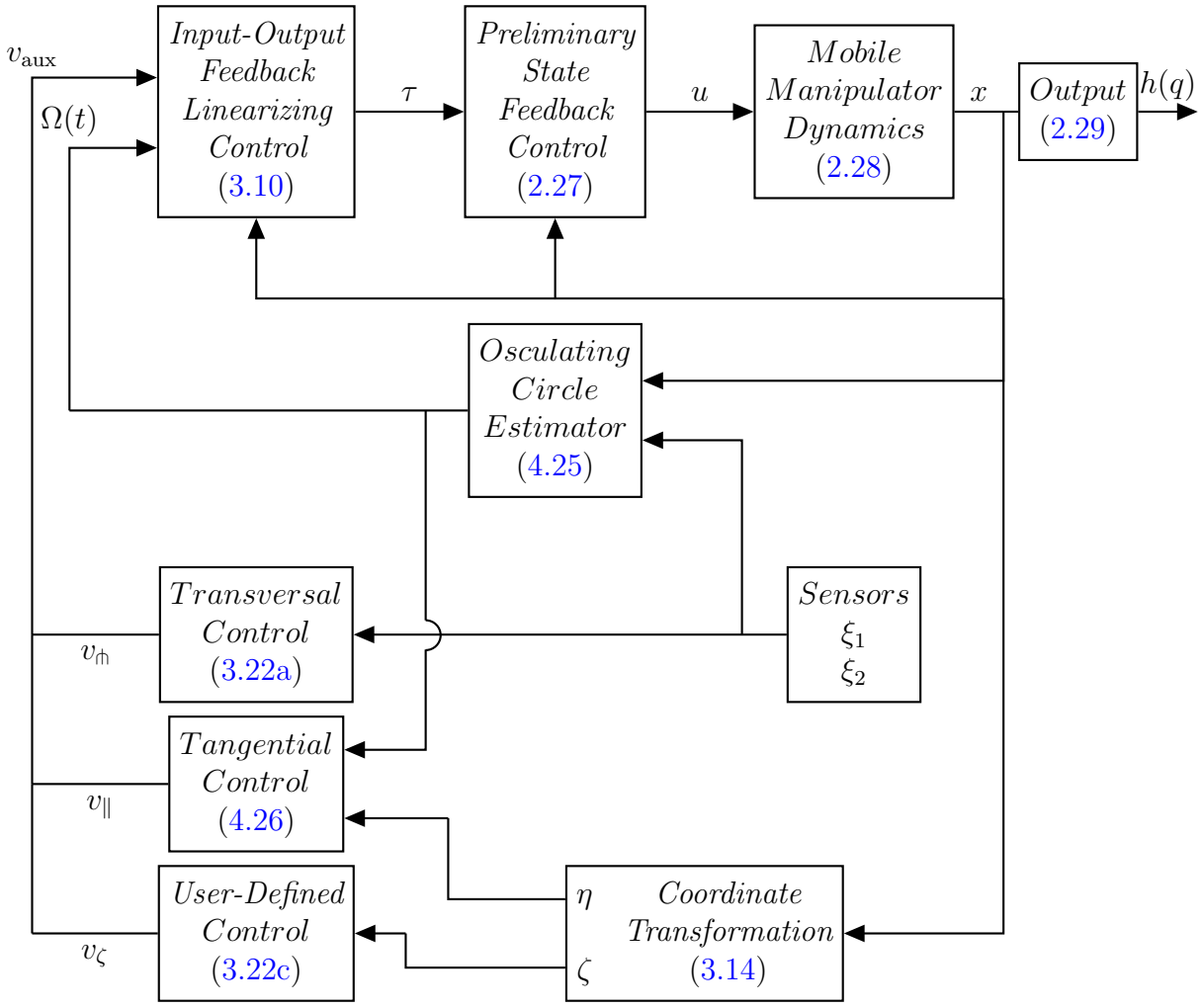

Figure 4.5: Block diagram of indirect adaptive path following controller for unknown strictly convex curves.

4.3.2 Simulation Results

To illustrate the effectiveness of our adaptive path following controller for strictly convex paths here we present a simulation of out adaptive path following system for and unknown ellipse

$$\sigma(\lambda) = \begin{bmatrix} 3\cos(\lambda) \\ \sin(\lambda) \end{bmatrix} \tag{4.27}$$

The robot is initialized at

$$q(0) = \begin{bmatrix} 0 \\ \pi/4 \\ 3 \\ 3 \\ 0 \end{bmatrix}, \quad v(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Omega(0) = \begin{bmatrix} 1 \\ 0.2 \\ 0 \end{bmatrix}.$$

with $\beta = 0.8$, $\Gamma = 700 I_3$, we take the reference signal to be $\eta_2^{ref}(t) = c_{\|}\hat{\kappa}(\lambda^{\star}(t))$ with $c_{\|} = 0.5$. We again use a velocity tracking controller for the tangential control and the linear feedback matrices (3.22) are

$$F_{\pitchfork} = \begin{bmatrix} -81 & -18 \end{bmatrix}, \quad F_{\|} = -28, \quad F_{\zeta} = \begin{bmatrix} -9 & -6 & 0 & 0 \\ 0 & 0 & -9 & -6 \end{bmatrix}.$$

The resulting motion of the end-effector and mobile base are shown in Figure 4.6 and the osculating circle parameter estimates are shown in Figure 4.8. Figures 4.6, 4.7 and 4.8 reveal that while our approach does not deliver online parameter convergence of the osculating circle, our indirect adaptive path following control approach still achieves control objectives of Problem 2 for the unknown ellipse.

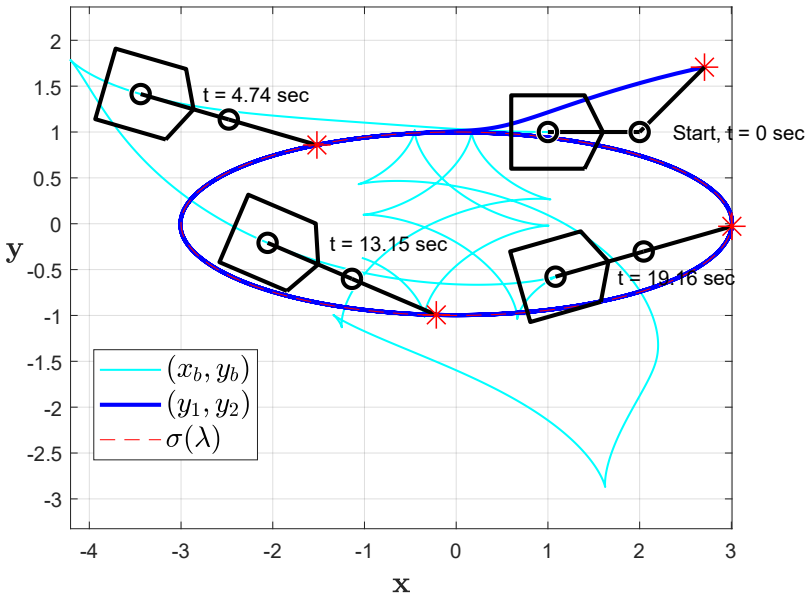

Figure 4.6: End-effector and mobile base trajectories for unknown ellipse.

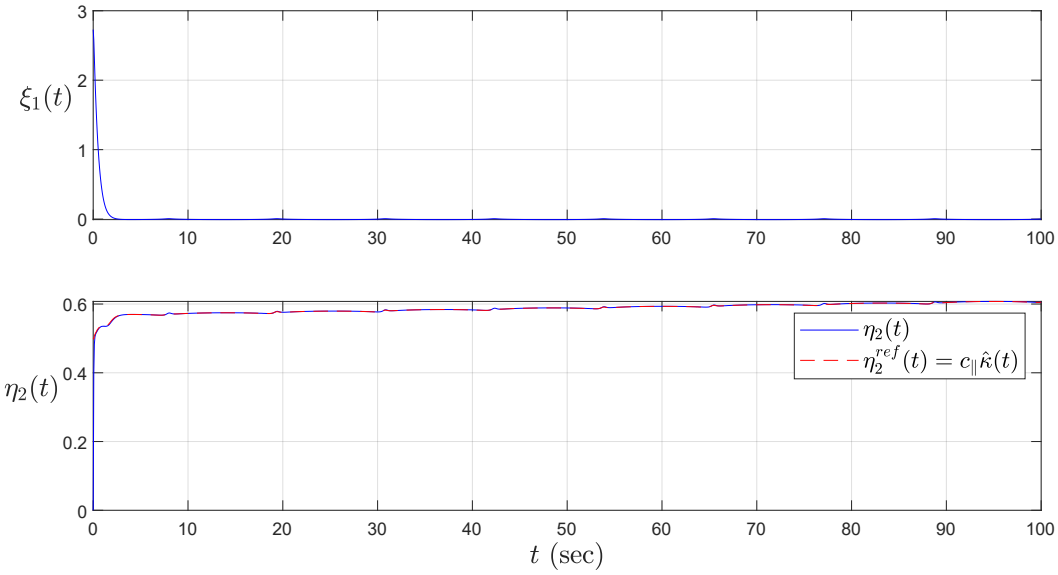

Figure 4.7: $\xi_1 = \left\| \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\top - \hat{\epsilon}(t) \right\| - \hat{r}(t)$ and $\eta_2(t)$ tracking $c_{\|}\hat{\kappa}(t)$.

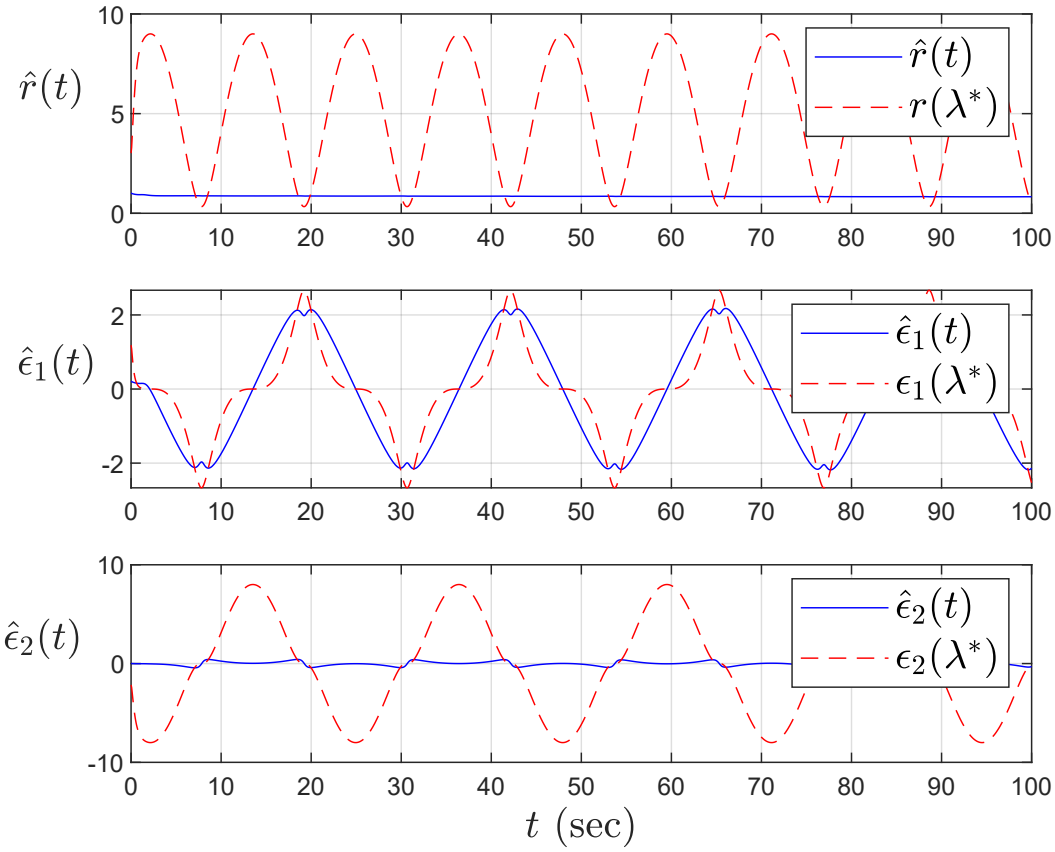

Figure 4.8: Osculating circle parameter convergence of adaptive law.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis we proposed a solution to the adaptive path following problem and designed path following controllers that adapt to unknown circular and strictly convex closed curves in the plane. In terms of design, we used an indirect adaptive control approach, i.e., given parameter estimates of an unknown path, we update our control in real-time.

We began by presenting a solution to the path following problem of general known closed curves and applied it to the case of known circular and elliptical paths. We also presented a second solution by means of a circular approximation of closed paths via osculating circle. We then proposed a solution to the adaptive path following problem for the case of unknown circular and strictly convex paths. First, we derived an adaptive law that estimates circular path parameters in real-time. Given the estimates from our adaptive law, we used our previously defined path following controller for circular paths and at each time instant updated the path following controller accordingly. We found that our path following controller that uses the circle approximation allowed us to extend our results to the case of unknown strictly convex closed curves.

5.2 Future work

Future work of the research presented in this thesis include the derivation of stability proofs of the control schemes for path following discussed in Chapter 3 and Chapter 4,

simulation suggests that an extension to the non-convex case is possible for controller from Section 3.4. The adaptive path following problem could also benefit from an extension and stability proofs to the adaptation of unknown non-convex paths with no line segments. All the proposed controllers in this thesis could be further validated via implementation on a physical nonholonomic mobile manipulator. Lastly the adaptive path following controllers of this thesis and any future work could be applied for obstacle avoidance.

References

[1] A. Pedro Aguiar, João P. Hespanha, and Petar V. Kokotović. Performance limitations in reference tracking and path following for nonlinear systems. *Automatica*, 44(3):598 – 610, 2008. 1

[2] A. Akhtar, C. Nielsen, and S. L. Waslander. Path following using dynamic transverse feedback linearization for car-like robots. *IEEE Transactions on Robotics*, 31(2):269–279, April 2015. 1, 31

[3] S. A. Al-Hiddabi and N. H. McClamroch. Tracking and maneuver regulation control for nonlinear nonminimum phase systems: application to flight control. *IEEE Transactions on Control Systems Technology*, 10(6):780–792, Nov 2002. 3

[4] Victor H Andaluz, Flavio Roberti, Juan Marcos Toibero, Ricardo Carelli, and Bernardo Wagner. Adaptive dynamic path following control of an unicycle-like mobile robot. In *International Conference on Intelligent Robotics and Applications*, pages 563–574. Springer, 2011. 4

[5] David Baraff. Linear-time dynamics using Lagrange multipliers. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996*, pages 137–146, 1996. 9

[6] Anthony M Bloch, PS Krishnaprasad, Jerrold E Marsden, and Richard M Murray. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136(1):21–99, 1996. 6

[7] Guy Campion and Georges Bastin. Indirect adaptive state feedback control of linearly parametrized non-linear systems. *International Journal of Adaptive Control and Signal Processing*, 4(5):345–358, 1990. 1

[8] Yongcan Cao. Uav circumnavigating an unknown target under a gps-denied environment with range-only measurements. *Automatica*, 55:150 – 158, 2015. 2

[9] Sandra H Dandach, Bars Fidan, Soura Dasgupta, and Brian D O Anderson. Systems & Control Letters A continuous time linear adaptive source localization algorithm , robust to persistent drift. *Systems & Control Letters*, 58(1):7–16, 2009. 2

[10] K.D. Do, Z.P. Jiang, and J. Pan. Robust adaptive path following of underactuated ships. *Automatica*, 40(6):929 – 944, 2004. 4

[11] Samet Güler and Baris Fidan. Target capture and station keeping of fixed speed vehicles without self-location information. *European Journal of Control*, 43:1 – 11, 2018. 2

[12] R. Gill, D. Kulic, and C. Nielsen. Robust path following for robot manipulators. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3412–3418, Nov 2013. 4

[13] V. Guillemin and A. Pollack. *Differential Topology*. Prentice Hall, New Jersey, 1974. 23

[14] Andre Hladio, Christopher Nielsen, and David Wang. Path following for a class of mechanical systems. *IEEE Transactions on Control Systems Technology*, 21(6):2380–2390, 2013. 1, 21, 23

[15] Qiang Huang, Kazuo Tanie, and Shigeki Sugano. Coordinated motion planning for a mobile manipulator considering stability and manipulation. *The International Journal of Robotics Research*, 19(8):732–742, 2000. 2

[16] S. K. Ider and F. M.L. Amirouche. Coordinate reduction in the dynamics of constrained multibody systems-a new approach. *Journal of Applied Mechanics, Transactions ASME*, 55(4):899–904, 1988. 9

[17] P. Ioannou and J. Sun. Theory and design of robust direct and indirect adaptive-control schemes. *International Journal of Control*, 47(3):775–813, 1988. 1

[18] Petros Ioannou and Baris Fidan. *Adaptive control tutorial*. SIAM, 2006. 1, 2, 41, 42

[19] A. Isidori. *Lectures in Feedback Design for Multivariable Systems*. Springer, 2017. 2, 23

[20] Gerhard Kreisselmeier. A robust indirect adaptive-control approach. *International Journal of Control*, 43(1):161–175, 1986. 1

[21] T. W. Lee and D. C. H. Yang. On the Evaluation of Manipulator Workspace. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105(1):70–77, 03 1983. 2

[22] Y. Li and C. Nielsen. Synchronized closed path following for a differential drive and manipulator robot. *IEEE Transactions on Control Systems Technology*, 25(2):704–711, 2016. 2, 4, 20

[23] Z. Li, S. S. Ge, M. Adams, and W. S. Wijesoma. Adaptive robust output-feedback motion/force control of electrically driven nonholonomic mobile manipulators. *IEEE Transactions on Control Systems Technology*, 16(6):1308–1315, Nov 2008. 1

[24] Z. Li, Y. Yang, and J. Li. Adaptive motion/force control of mobile under-actuated manipulators with dynamics uncertainties by dynamic coupling and output feedback. *IEEE Transactions on Control Systems Technology*, 18(5):1068–1079, Sep. 2010. 1

[25] Alicja Mazur. Hybrid adaptive control laws solving a path following problem for non-holonomic mobile manipulators. *International Journal of Control*, 77(15):1297–1306, 2004. 4

[26] Alicja Mazur. Path following for nonholonomic mobile manipulators. In Krzysztof Kozłowski, editor, *Robot Motion and Control 2007*, pages 279–292, London, 2007. Springer London. 4

[27] Alicja Mazur. Trajectory tracking control in workspace-defined tasks for nonholonomic mobile manipulators. *Robotica*, 28(1):5768, 2010. 2, 3

[28] Wei Meng, Chen Guo, and Yang Liu. Robust adaptive path following for under-actuated surface vessels with uncertain dynamics. *Journal of Marine Science and Application*, 11(2):244–250, 2012. 1, 4

[29] Christopher Nielsen, Cameron Fulford, and Manfredi Maggiore. Path following using transverse feedback linearization: Application to a maglev positioning system. *Automatica*, 46(3):585 – 590, 2010. 1

[30] Christopher Nielsen and Manfredi Maggiore. Output stabilization and maneuver regulation: A geometric approach. *Systems & control letters*, 55(5):418–427, 2006. 4

[31] François G. Pin and Jean Christophe Culioli. Optimal positioning of combined mobile platform-manipulator systems for material handling tasks. *Journal of Intelligent and Robotic Systems*, 6(2-3):165–182, 1992. 3

[32] A. Pressley. *Elementary Differential Geometry.* Springer, New York, U.S.A., 2001. 35

[33] Homayoun Seraji. A unified approach to motion control of mobile manipulators. *The International Journal of Robotics Research*, 17(2):107–118, 1998. 3

[34] Michael Shneier and Roger Bostelman. Literature review of mobile robots for manufacturing. 2015. 3

[35] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010. 2, 6, 9

[36] R. Skjente, T.I. Fossen, and P.V. Kokotović. Robust output maneuvering for a class of nonlinear systems. *Automatica*, 40(3):373–383, March 2004. 17, 18

[37] M W Spong, S Hutchinson, and M Vidyasagar. *Robot Modeling and Control.* John Wiley and Sons, Hoboken, 2006. 2, 7

[38] M. Steinfeld and C. Nielsen. Synchronized path following for lti systems and closed paths: Laboratory implementation. In *IEEE International Conference on Control and Automation*, pages 435–440, 2018. 2, 20

[39] Yi Wei Daniel Tay, Biranchi Panda, Suvash Chandra Paul, Nisar Ahamed Noor Mohamed, Ming Jen Tan, and Kah Fai Leong. 3D printing trends in building and construction industry: a review. *Virtual and Physical Prototyping*, 12(3):261–276, 2017. 3

[40] Xiao-fei Wang, Zao-jian Zou, Tie-shan Li, and Wei-lin Luo. Adaptive path following controller of underactuated ships using serret-frenet frame. *Journal of Shanghai Jiaotong University (Science)*, 15(3):334–339, 2010. 1

[41] Xu Zhang, Mingyang Li, Jian Hui Lim, Yiwei Weng, Yi Wei Daniel Tay, Hung Pham, and Quang-Cuong Pham. Large-scale 3d printing by a team of mobile robots. *Automation in Construction*, 95:98 – 106, 2018. 3

[42] Ray Y. Zhong, Xun Xu, Eberhard Klotz, and Stephen T. Newman. Intelligent manufacturing in the context of industry 4.0: A review. *Engineering*, 3(5):616 – 630, 2017. 3

APPENDICES

Appendix A

Angular variables, the circle

For an angular variable θ, the quantities θ and $\theta + 2\pi$ represent the same position on the unit circle. Thus, geometrically, we can view angular variables as belong to a set that is diffeomorphic to a circle. Consider the unit circle in $\mathbb{R}^2$

$$\mathbb{S}^1 \coloneqq \left\{ x \in \mathbb{R}^2 : \|x\|_2^2 = 1 \right\}. \tag{A.1}$$

This set is example of a *manifold*, more specifically, we've expressed the manifold $\mathbb{S}^1$ as being *embedded* in the plane. In this thesis we write $\theta \in \mathbb{S}^1$ for angular variables. The manifold $\mathbb{S}^1$ is *compact* and one-dimensional. Thus, when we write $\theta \in \mathbb{S}^1$ what we actually mean is that some local choice of coordinates has been made and the quantity θ can be treated as a scalar, i.e., an angle. This is because the set $\mathbb{S}^1$ can almost be covered with a single coordinate chart (U, φ) with

$$U \coloneqq \mathbb{S}^1 \backslash \{(-1, 0)\}$$

and $\varphi : U \to (-\pi, \pi)$ defined as $\varphi(x) = \arctan(x_2, x_1)$, i.e., the four quadrant arctangent. It's clear that $U \neq \mathbb{S}^1$ so this coordinate chart doesn't cover all of $\mathbb{S}^1$, however, in this thesis we will ignore this fact. In summary, when we write $\theta \in \mathbb{S}^1$ we should think of θ taking values in the open interval $(-\pi, \pi)$ even though this doesn't incude all the points on the unit circle $\mathbb{S}^1$.

Appendix B

Zero dynamics

Recall a definition from nonlinear control systems. Consider the nonlinear control system

$$\dot{x}(t) = f(x(t), u(t)) \tag{B.1a}$$
$$y(t) = h(x(t)) \tag{B.1b}$$

with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^m$ and output $y(t) \in \mathbb{R}^p$. We assume that $f(0,0) = 0$ and $h(0) = 0$. Let

$$\mathcal{Z} \coloneqq \{x \in \mathbb{R}^n : h(x) = 0\} \qquad \text{(output zeroing set)}.$$

Under the assumption that, for all $x \in \mathcal{Z}$, $\mathrm{d}h_x$ has full rank, this set is an (embedded) submanifold of the state-space $\mathbb{R}^n$. Let $\mathcal{Z}_0 \subseteq \mathcal{Z}$ denote the connected component of $\mathcal{Z}$ which contains $x = 0$.

Example B.0.1. If $h : \mathbb{R}^2 \to \mathbb{R}$, $(x_1, x_2) \mapsto x_2(x_1^2 + x_2^2 - 1)$, then

$$\mathcal{Z} = \left\{x \in \mathbb{R}^2 : x_2 = 0 \text{ or } x_1^2 + x_2^2 - 1 = 0\right\}$$

and $\mathcal{Z}_0 = \{x \in \mathbb{R}^2 : x_2 = 0\}$. In this case

$$\mathrm{d}h_x = \begin{bmatrix} 2x_1x_2 & 3x_2^2 \end{bmatrix}$$

isn't full rank on all of $\mathcal{Z}$, but, nevertheless, $\mathcal{Z}_0$ is an embedded submanifold of $\mathbb{R}^2$. ▲

Now let $\mathcal{Z}^\star \subseteq \mathcal{Z}_0$ denote the largest, in terms of set inclusion, controlled invariant set contained in $\mathcal{Z}_0$ which contains $x = 0$. Then $\mathcal{Z}^\star$ is called the **zero dynamics manifold**

of (B.1) near $x = 0$. This set has the property that, if $x(0) \in \mathcal{Z}^\star$, there exists a control law $u^\star : \mathcal{Z}^\star \to \mathbb{R}^m$ such that the state remains on $\mathcal{Z}^\star$ for all t on an open interval containing 0 and that the corresponding output on over this time interval is identically equal to zero. The dynamics of (B.1) restricted to $\mathcal{Z}^\star$ are called the **zero dynamics** of (B.1).

Example B.0.2. To illustrate these ideas in the special case when (B.1) happens to be linear, consider

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} c_1 & 1 \end{bmatrix}\end{aligned}$$

where $c_1 \neq 0$. In this case it is easy to check that

$$\mathcal{Z} = \mathcal{Z}_0 = \operatorname{Ker} \begin{bmatrix} c_1 & 1 \end{bmatrix} = \operatorname{span} \begin{bmatrix} 1 \\ -c_1 \end{bmatrix}.$$

In this case the set $\mathcal{Z}$ itself can be made invariant through (linear) state feedback since it's an (A, B)-invariant subspace, i.e., $A\mathcal{Z} \subseteq \mathcal{Z} + \operatorname{Im}(B)$. Thus, in this example, $\mathcal{Z} = \mathcal{Z}^\star$ and one choice of $u^\star : \mathcal{Z}^\star \to \mathbb{R}$ to make $\mathcal{Z}^\star$ invariant is

$$u^\star(x) = Fx = \begin{bmatrix} -3c_1 + 2 + c_1^2 & 0 \end{bmatrix}.$$

If we let $A_F \coloneqq A + BF$, then $\mathcal{Z}^\star$ is an A_F-invariant subspace. To obtain the system's zero dynamics we need to restrict the system to $\mathcal{Z}^\star$. If $V : \mathcal{Z}^\star \to \mathbb{R}^2$ is the insertion map, then the restriction of A_F to $\mathcal{Z}^\star$ is the map which makes the following diagram commute

$$\begin{array}{ccc} \mathbb{R}^2 & \xrightarrow{A_F} & \mathbb{R}^2 \\ \uparrow{\scriptstyle V} & & \uparrow{\scriptstyle V} \\ \mathcal{Z}^\star & \xrightarrow{A|_{\mathcal{Z}^\star}} & \mathcal{Z}^\star. \end{array}$$

Solving the equation $A_F V = V\, A|_{\mathcal{Z}^\star}$ yields $A|_{\mathcal{Z}^\star} = -c_1$ and the system's zero dynamcis are

$$\dot{z}(t) = -c_1 z(t).$$

These dynamics are exponentially stable if $c_1 > 0$; unstable if $c_1 < 0$ and stable in the sense of Lyapunov if $c_1 = 0$.

In this example it is interesting to note that the zero dynamics are independent of the choice of $u^\star$ and that the eigenvalues of $A|_{\mathcal{Z}^\star}$ equal the zeros of the transfer function $C(sI - A)^{-1}B$. ▲