Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Adaptive Path Following for a Nonholonomic Mobile Manipulator ⋆

Orlando Barrera Perez Baris Fidan Christopher Nielsen

Faculty of Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.
E-mail:{orlando.barrera,fidan,cnielsen}@uwaterloo.ca

Abstract: We investigate an adaptive path following problem for a 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 path. First, we present a solution to the non-adaptive path following problem using the concept of a path following output. Then, we use an indirect adaptive control approach to design path following controllers for a feasible class of strictly convex paths.

Keywords: Mobile manipulators, path following, closed curves, adaptive control.

1. INTRODUCTION

In robotic applications, path following problems involve the design of feedback controllers that drive a certain physical feature of a robotic system towards a known geometric path with a predetermined desired motion about the path (Hladio et al. (2013)). In many of these applications, the robotic system lacks global information about the path to follow. For example, an autonomous car when driving on an unknown road may not know more than the next 20 or so metres of the road, or in a warehouse setting, where the routes the automated ground vehicles follow may change frequently in real-time.

Lack of complete global information about the path to follow introduces interesting challenges to the research and development of path following control. Motivated by these challenges, in what follows we build up to the adaptive path following problem and our goal is to design a path following controller that adapts to unknown and strictly convex paths for a 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 (Li et al. (2008, 2010); Wang et al. (2010); Meng et al. (2012)), i.e., in our case we assume our dynamics are known but take the path to be unknown.

We begin by proposing a solution to the non-adaptive path following problem for a nonholonomic mobile manipulator using the notion of a path following output (Li and Nielsen (2016)) to design an input-output feedback linearizing controller with its associated normal form for arbitrary, smooth, closed paths; we apply it to the circular path case. There are drawbacks associated with our proposed controller when applied to arbitrary closed paths; to overcome these drawbacks for strictly convex paths we apply a circular approximation to the path by means of an osculating circle representation 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, aligned with distance-measurement based target localization studied in (Dandach et al. (2009)), estimates the radius of the target circle. Thereafter, we design a path following controller with an indirect adaptive control approach 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 one developed for circular paths, however, this time with respect to the estimated osculating circle.

In this paper, 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 unit circle is denoted by $\mathbb{S}^1$. The differential of a function f evaluated at x is written as $\mathrm{d}f_x$.

2. PROBLEM FORMULATION

2.1 Mathematical Model

We study a mobile manipulator consisting of a two-link manipulator mounted on a differential drive mobile base robot. We assume that the manipulator's links have their masses concentrated at their endpoints. Referring to Figure 1, the configuration manifold is $\mathcal{Q} := \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{R}^2 \times \mathbb{S}^1$ and we choose coordinates $q := [\theta_1 \ \theta_2 \ x_b \ y_b \ \theta_h]^\top$. 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 and define $u := [\tau_1 \ \tau_2 \ a \ \alpha]^\top$.

⋆ Research supported by a grant from the Natural Sciences and Engineering Research Council of Canada. Orlando Barrera Perez was supported by Consejo Nacional de Ciencia y Tecnología, Government of Mexico.

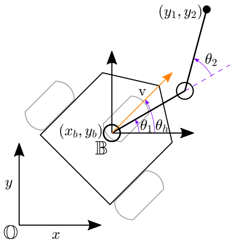

Fig. 1. Schematic diagram of two-link mobile manipulator.

Following standard Euler-Lagrange modelling techniques, the system's model can be expressed as

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

where $v = [\dot{\theta}_1\ \dot{\theta}_2\ \mathrm{v}\ \dot{\theta}_h]^{\top} \in \mathbb{R}^4$, with v equal to the translational speed of the base, and

$$B(q) = G(q) = \begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \cos(\theta_h) & 0\\ 0 & 0 & \sin(\theta_h) & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Let ℓ_1, ℓ_2 be, respectively, length of link 1 and 2, m_b is the mass of base, m_1 is the mass of link 1, m_2 is the mass of link 2, and I_z is the moment of inertia of the mobile base about the axis of rotation at its centre of mass. Then

$$M(q) = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) & 0\\ M_{21}(q) & \ell_2^2 m_2 & M_{23}(q) & 0\\ M_{31}(q) & M_{32}(q) & m_b + m_1 + m_2 & 0\\ 0 & 0 & 0 & I_z \end{bmatrix}, \tag{3}$$

where

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

Lastly,

$$m(q,v) = [m_1(q,v)\ m_2(q,v)\ m_3(q,v)\ 0]^{\top} \tag{4}$$

where

$$\begin{aligned} m_1(q,v) =& -\ell_1\ell_2 m_2 \sin(\theta_2)(\dot{\theta}_2^2 + \dot{\theta}_1\dot{\theta}_2) + (\ell_1 m_1 \cos(\theta_1 - \theta_h)\\ & + \ell_1 m_2 \cos(\theta_1 - \theta_h) + \ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h))\mathrm{v}\dot{\theta}_h,\\ m_2(q,v) =& \ell_1\ell_2 m_2 \sin(\theta_2)\dot{\theta}_1^2 + \ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h)\mathrm{v}\dot{\theta}_h,\\ m_3(q,v) =& -(\ell_1 m_1 \cos(\theta_1 - \theta_h) + \ell_1 m_2 \cos(\theta_1 - \theta_h))\dot{\theta}_1^2\\ & - (\ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h))(\dot{\theta}_1^2 + \dot{\theta}_2^2)\\ & - 2\ell_2 m_2 \cos(\theta_1 + \theta_2 - \theta_h)\dot{\theta}_1\dot{\theta}_2. \end{aligned}$$

For this system, for all $q \in \mathcal{Q}$, $G^{\top}(q)B(q) = I_4$ so if we apply the preliminary state feedback

$$u = m(q,v) + M(q)\tau, \tag{5}$$

where $\tau \in \mathbb{R}^4$ is an auxiliary control input yet to be specified, we obtain the partially compensated system

$$\begin{aligned}\dot{q} &= G(q)v,\\ \dot{v} &= \tau.\end{aligned} \tag{6}$$

The output y of (6) is defined by a function $h : \mathcal{Q} \to \mathcal{Y}$, $\mathcal{Y} := \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^1$, as

$$\begin{bmatrix} y_1\\ y_2\\ y_3\\ y_4 \end{bmatrix} = h(q) := \begin{bmatrix} x_b + \ell_1\cos(\theta_1) + \ell_2\cos(\theta_1 + \theta_2)\\ y_b + \ell_1\sin(\theta_1) + \ell_2\sin(\theta_1 + \theta_2)\\ \theta_2\\ \operatorname{arctan2}(\sin(\theta_1 - \theta_h), \cos(\theta_1 - \theta_h)) \end{bmatrix}. \tag{7}$$

In (7), (y_1, y_2) 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 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.

2.2 Admissible Paths and Desired Motion Along Path

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

$$\sigma : \mathbb{S}^1 \to \mathbb{R}^2, \quad \sigma'(\lambda) \neq 0 \text{ for all } \lambda \in \mathbb{S}^1. \tag{8}$$

Assumption 1. *In addition to the parametric representation (8), there exists a known smooth function $s : U \subseteq \mathbb{R}^2 \to \mathbb{R}$, where U is an open and connected set, such that*

$$\mathcal{C} = \{(y_1, y_2) \in \mathbb{R}^2 : s(y_1, y_2) = 0\} \tag{9}$$

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

With this assumption, the path in the task space $\mathcal{Y}$ of the two-link mobile manipulator equals the set

$$\gamma := \{y \in \mathcal{Y} : s(y_1, y_2) = y_3 = y_4 = 0\} \tag{10}$$

which, under Assumption 1, is an embedded submanifold.

The dynamic task in a path following problem to make the output y along the path γ in a pre-specified way, e.g., having the end-effector traverse the entire curve $\mathcal{C}$. 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 2. *The desired motion along the assigned path (8) is described by a timing law $\lambda^{\mathrm{ref}} : \mathbb{R} \to \mathbb{S}^1$. The timing law is produced by a known exogenous system*

$$\dot{w}(t) = Sw(t), \qquad w(0) \in \mathbb{R}^{n_r}, \tag{11a}$$

$$\lambda^{\mathrm{ref}}(t) = \arg(\exp(jQw(t)) \tag{11b}$$

with $S \in \mathbb{R}^{n_r \times n_r}$, $Q \in \mathbb{R}^{1 \times n_r}$. ♦

2.3 Problem Statement

We state the two main problems considered in this paper.

Problem 1 (Path following). Consider the mobile manipulator model (6),(7). Suppose we are given a closed path $\mathcal{C}$ satisfying Assumption 1 and a desired motion along the path that satisfies Assumption 2. Find a state-feedback controller τ such that the closed-loop system enjoys the following properties:

(i) The output is driven towards the path (10). In particular, there exists an open set in $\mathcal{Q} \times \mathbb{R}^4$ such that any initial condition in this set results in $y(t) \to \gamma$ as $t \to \infty$ with $(q(t), v(t))$ bounded.

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

(ii) The path (10) is output invariant. In particular, if $(q(0), v(0)) \in \mathcal{Q} \times \mathbb{R}^4$, are such that $y(0) \in \gamma$ and $\dot{y}(0) \in \mathsf{T}_{y(0)}\gamma$, then, for all $t \geq 0$, $y(t) \in \gamma$.
(iii) The output asymptotically converges to the desired motion along the path.

In Section 4 we consider an adaptive version of this problem for circular paths.

Problem 2 (Adaptive path following). Consider the mobile manipulator model (6),(7) and a circle $\mathcal{C}$ of unknown radius and unknown location in the plane. Further suppose that the controller has access to the signed distance from the end-effector to $\mathcal{C}$ as well as the rate of change of this distance. Find a dynamic control law τ such that the closed-loop system enjoys the following:

(i) There exists an open set in $\mathcal{Q} \times \mathbb{R}^4$ such that, for any initial condition in the set results in
$$y(t) \to \gamma \quad \text{as} \quad t \to \infty \text{ (attractivity)}$$
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. NON-ADAPTIVE PATH FOLLOWING

3.1 Path Following Output and Control Design

To solve the aforementioned problems, it is convenient to define the so-called path following output (Li and Nielsen (2016)) 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 1. Next, define a function $\varpi : U \subset \mathbb{R}^2 \to \mathbb{S}^1$ by

$$\varpi(y_1, y_2) := \arg\min_{\lambda \in \mathbb{S}^1} \left\| [y_1 \;\, y_2]^\top - \sigma(\lambda) \right\|. \qquad (12)$$

Intuitively, $\varpi(y_1, y_2)$ equals the parameter $\lambda^\star \in \mathbb{S}^1$ with the property that $\sigma(\lambda^\star)$ is the closest point on the curve $\mathcal{C}$ to (y_1, y_2). Combining the function (12) with the function (9) from Assumption 1, we define the path following output to be $h_{\mathrm{PF}} : U \times \mathbb{S}^1 \times \mathbb{S}^1 \subseteq \mathcal{Y} \to \mathbb{R} \times \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$,

$$y_{\mathrm{PF}} = h_{\mathrm{PF}}(y) := \begin{bmatrix} s(y_1, y_2) \\ \varpi(y_1, y_2) \\ y_3 \\ y_4 \end{bmatrix} = h_{\mathrm{PF}} \circ h(q). \qquad (13)$$

Driving the output (7) to the path (10) is equivalent, under mild technical conditions, to driving $y_{\mathrm{PF},1}$, $y_{\mathrm{PF},3}$ and $y_{\mathrm{PF},4}$ to zero. Converging to the desired motion along the path is equivalent to driving $y_{\mathrm{PF},2}$ to $\lambda^{\mathrm{ref}}(t)$.

Consider the mobile manipulator (6) with the path following output (13). It is straightforward to show that this system 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

$$\tau := \left(\mathrm{d}h_{\mathrm{PF}}|_{h(q)} \; \mathrm{d}h_q \; G(q) \right)^{-1} \left(\frac{\partial}{\partial q} \left(\mathrm{d}h_{\mathrm{PF}}|_{h(q)} \; \mathrm{d}h_q \; G(q)v \right) G(q)v + v_{\mathrm{aux}} \right), \quad (14)$$

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

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. In the case of (6),(13), it turns out that 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 contains the path (10), i.e., the normal form is valid in a neighbourhood of the *entire path*.

Define the coordinate transformation

$$T(q, v) := \begin{bmatrix} \theta_h \\ h_{\mathrm{PF}} \circ h(q) \\ \mathrm{d}h_{\mathrm{PF}}|_{h(q)} \; \mathrm{d}h_q \; G(q)v \end{bmatrix}. \qquad (15)$$

The last 8 components of the transformation (15) are simply y_{PF} and $\dot{y}_{\mathrm{PF}}$.

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

The function (15) *maps the set* $W \times \mathbb{R}^4 \subset \mathcal{Q} \times \mathbb{R}^4$ *diffeomorphically onto its image.*

The proof of this result is omitted but can be found in (Barrera Perez, O., 2020, Proposition 3.2.1). Proposition 1 together with the feedback control law (14) show that on the set $W \times \mathbb{R}^4$, the system (6),(13) is feedback equivalent to the input-output feedback linearized system

$$\dot{z} = [0 \;\, 0 \;\, 0 \;\, 1] \left(\mathrm{d}h_{\mathrm{PF}}|_{h(q)} \; \mathrm{d}h_q \; G(q) \right)^{-1} \begin{bmatrix} \xi_2 \\ \eta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix}, \qquad (16a)$$

$$\dot{\xi} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{\pitchfork}, \qquad (16b)$$

$$\dot{\eta} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \eta + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{\parallel}, \qquad (16c)$$

$$\dot{\zeta} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \zeta + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} v_{\zeta}, \qquad (16d)$$

where we have decomposed the auxiliary input as $v_{\mathrm{aux}} := [v_{\pitchfork} \; v_{\parallel} \; v_\zeta^\top]^\top$. With respect to Problem 1, 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 (16b) and $\zeta = 0$ is an equilibrium point of the ζ-subsystem (16d).

The tangential subsystem (16c) governs the portion of the manipulator dynamics that produces observable motion along the path. They can be used to express the desired motion along the path. Define the error
$$e_1 := \arg(\exp(j(\eta_1 - \lambda^{\mathrm{ref}}))).$$
Simple calculations using (16c) and the exosystem model (11) give that $\dot{e}_1 = \eta_2 - QSw$, thus letting $e = [e_1 \;\, e_2]^\top :=$

$[e_1\ \dot{e}_1]^\top \in \mathbb{S}^1 \times \mathbb{R}$, the tangential dynamics (16c) can be replaced with

$$\begin{aligned}\dot{e} &= \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} e + \begin{bmatrix}0\\1\end{bmatrix} v_\parallel - \begin{bmatrix}0\\ QS^2\end{bmatrix} w \\ \dot{w} &= Sw \qquad \text{(exosystem).}\end{aligned} \tag{17}$$

Driving $e(t)$ to zero is equivalent to tracking the desired motion along the path.

The transformed manipulator system (17) dynamics suggest the linear control laws

$$v_\pitchfork = F_\pitchfork \xi, \quad v_\parallel = F_\parallel e + QS^2 w, \quad v_\zeta = F_\zeta \zeta \tag{18}$$

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 (17) are exponentially stable.

Remark 1. *The transformed system* (16) *allows us to consider desired motions along* $\mathcal{C}$ *other than the type described in Section 2.2, e.g., velocity tracking. In that case the tangential controller* $v_\parallel$ *is taken to be* $v_\parallel = F_\parallel(\eta_2 - \eta_2^{\text{ref}}) + QSw$, $F_\parallel < 0$. ▲

In summary, our proposed path following controller consists of Equations (3), (4) and (5) which define the preliminary feedback u, Equation (11) which produces the desired motion along the path, Equation (14) which input-output feedback linearizes system (6),(13) and the coordinate transformation (15) which is needed to implement the linear feedbacks in Equation (18).

3.2 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 (13) do not, in general, have closed-form expressions. Furthermore, even when closed-form expressions are available, the expressions in the coordinate transformation (15), which are needed in order to implement the linear feedback laws (18), can be complicated. In this subsection we propose a path following controller for arbitrary closed curves that applies a circular path following controller to the osculating circle associated to the closest point of the closed curve.

Given a strictly convex curve $\mathcal{C}$ with parametric representation (8), its signed curvature at $\lambda \in \mathbb{S}^1$ is

$$\kappa(\lambda) = \frac{\sigma_1''(\lambda)\sigma_2'(\lambda) - \sigma_2''(\lambda)\sigma_1'(\lambda)}{\|\sigma'(\lambda)\|^3}. \tag{19}$$

Since $\mathcal{C}$ is assumed to be strictly convex, for all $\lambda \in \mathbb{S}^1$, $\kappa(\lambda) > 0$. The centre of curvature $\epsilon(\lambda)$ of σ at the point $\sigma(\lambda)$ is defined to be

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

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

$$\sigma_\lambda(s) := \frac{1}{|\kappa(\lambda)|} \begin{bmatrix}\cos(s)\\ \sin(s)\end{bmatrix} + \epsilon(\lambda) \tag{20}$$

and implicitly by

$$s_\lambda(y_1, y_2) := \| [y_1\ y_2]^\top - \epsilon(\lambda)\| - |\kappa(\lambda)|^{-1}. \tag{21}$$

The idea proposed in this section is to first compute the parameter $\lambda^\star$ returned by the function (12) for the *given* curve $\mathcal{C}$. As mentioned at the start of this subsection, 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 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 (20) and implicit representation (21). To this circle we associate a path following output

$$y_{\text{PF}} = h_{\text{PF}}(y) = \begin{bmatrix} \| [y_1\ y_2]^\top - \epsilon(\lambda^\star)\| - |\kappa(\lambda^\star)|^{-1} \\ \operatorname{arctan2}(y_2 - \epsilon_2(\lambda^\star), y_1 - \epsilon_1(\lambda^\star)) \\ y_3 \\ y_4 \end{bmatrix}. \tag{22}$$

In (22), $\lambda^\star$ is a function of (y_1, y_2) but in applying the circular path following controller 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 (14) and the coordinate transformation (15).

We also modify the definition of the desired motion along the path because the second component in (22) doesn't return a value for the parameter of the actual path $\mathcal{C}$. In particular, at each moment in time, $y_{\text{PF},2}(t)/|\kappa(\lambda^\star(t))|$ equals the arclength along the osculating circle from $\sigma_{\lambda^\star}(0)$ to $\sigma_{\lambda^\star}(y_{\text{PF},2}(t))$. Thus, in order to get close to unit-speed traversal along the path, we take the reference for $\eta_2 \approx \dot{y}_{\text{PF},2}$ to be $\eta_2^{\text{ref}}(t) = \kappa(\lambda^\star(t))$.

4. ADAPTIVE PATH FOLLOWING

In this section we consider Problem 2; the mobile manipulator is required to follow a circular path of unknown radius $r > 0$ and unknown centre $c = [c_1\ c_2]^\top$ in the plane.

4.1 Circular Path Parameter Estimation

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

$$\xi_1(t) := ||p(t) - c|| - r, \tag{23}$$

$$\xi_2(t) := \dot{\xi}_1(t). \tag{24}$$

are measurable. ♦

Under Assumption 3, we now derive a parametric model for our circular path parameter estimation algorithm. Differentiating the identity

$$(\xi_1(t) + r)^2 = (p(t) - c)^\top (p(t) - c).$$

with respect to time and rearranging

$$p^\top(t)\dot{p}(t) - \xi_1(t)\xi_2(t) = r\xi_2(t) + c^\top \dot{p}(t). \tag{25}$$

Since the only unknowns in (25) are r and c, it is in linear parametric model form. Define

$$\mu(t) := p^\top(t)\dot{p}(t) - \xi_1(t)\xi_2(t) \in \mathbb{R}, \tag{26}$$
$$\Omega^* := \begin{bmatrix} r \\ c \end{bmatrix} \in \mathbb{R}^3, \quad \phi(t) := \begin{bmatrix} \xi_2(t) \\ \dot{p}(t) \end{bmatrix} \in \mathbb{R}^3.$$

so that (25) can be expressed as

$$\mu(t) = \Omega^{*\top}\phi(t) \tag{27}$$

The estimation model has the same form as (27), with the unknown vector Ω^* replaced with its time-varying estimate $\Omega(t) := [\hat{r}(t)\ \hat{c}^\top(t)]^\top$; the estimation model is

$$\hat{\mu}(t) := \Omega^\top(t)\phi(t). \tag{28}$$

Define the estimation error to be

$$\varepsilon(t) := \frac{\mu(t) - \hat{\mu}(t)}{1 + \beta\|\phi(t)\|^2}, \qquad \beta > 0. \tag{29}$$

Equation (29) defines a signal that indirectly reflects the difference between $\Omega(t)$ and Ω^*.

To update the estimate $\Omega(t)$, define an instantaneous cost criterion for the estimation error (29) as

$$J(\Omega) := \frac{1}{2}\frac{(\mu(t) - \Omega^\top\phi(t))^2}{1 + \beta\|\phi(t)\|^2}. \tag{30}$$

The approach is to update $\Omega(t)$ such that (30) is minimized so that $\varepsilon(t) \to 0$ as $t \to \infty$; we use gradient descent

$$\dot{\Omega}(t) = -\Gamma\nabla J(\Omega) \tag{31}$$

where $\Gamma = \Gamma^\top \in \mathbb{R}^{3\times 3}$ is a positive definite gain matrix and $\nabla J(\Omega)$ is the gradient of (30) with respect to Ω:

$$\nabla J(\Omega) = -\varepsilon(t)\phi(t). \tag{32}$$

Substituting (32) into (31) leads to the adaptive law

$$\dot{\Omega}(t) = \Gamma\varepsilon(t)\phi(t), \quad \Omega(0) = \Omega_0, \tag{33}$$

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

4.2 Adaptive Path Following for Circular Paths

Returning to the path following problem, consider a circle $\mathcal{C}$ in the inertial frame $\mathbb{O}$, represented parametrically as

$$\sigma(\lambda) = r\begin{bmatrix} \cos(\lambda) \\ \sin(\lambda) \end{bmatrix} + c, \tag{34}$$

where r and c are unknown. The role of $p(t)$ in Assumption 3 and in the algorithm from Subsection 4.1 is played by the mobile manipulator's end-effector location.We assume that sensors provide the signals (23) and (24).

The basic idea is to use a circular path following controller from Subsection 3.1 but use an estimate of the circle's radius r and centre c. We run the adaptive law described by (26), (28), (29) and (33). The path following output is taken to be

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

As in Section 3.2, 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 (14) and for the tangential η states in the coordinate transformation (15).

In order to satisfy property (ii) in Problem 2, we take the desired motion along the path to be $\eta_2^{\mathrm{ref}}(t) = w_0$. If $w_0 > 0$, the desired motion is in the same direction as increasing λ in the parameterized circle (34). We apply the linear controllers (18) however, cf. Remark 1, we take the tangential controller to be $v_\parallel = F_\parallel(\eta_2 - \eta_2^{\mathrm{ref}})$. We emphasize that in the linear feedback law $v_\pitchfork$ from (18), the true values of ξ (see Assumption 3) are used.

4.3 Adaptive Path Following for 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 (8) that is strictly convex, i.e., its signed curvature (19) is strictly positive, we make the following assumption.

Assumption 4. *Sensors provide the values $s(y_1(t), y_2(t))$ from (13) and $\dot{s}(y_1(t), y_2(t))$ for the unknown, strictly convex, path $\mathcal{C}$.* ♦

Under Assumption 4, and following the discussion from Subsection 3.2, 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 (20) and an implicit representation $s_{\lambda^\star(t)}(y_1(t), y_2(t))$ given by (21) 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, we equate the value of $s(y_1(t), y_2(t))$ to the distance of the end-effector to the osculating circle

$$\xi_1(t) := \|[y_1(t)\ y_2(t)]^\top - \epsilon(\lambda^\star(t))\| - r(\lambda^\star(t)).$$

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

$$\xi_2 := \frac{\left([y_1\ y_2]^\top - \epsilon(\lambda^\star)\right)^\top}{\|[y_1\ y_2]^\top - \epsilon(\lambda^\star)\|}\left(\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} - \dot{\epsilon}(\lambda^\star)\right) - \dot{r}(\lambda^\star),$$

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 $\Omega^\star(t) := [r(\lambda^\star(t))\ \epsilon^\top(\lambda^\star(t))]^\top$ are no longer constant. Repeating the calculations performed in Section 4.1 we are led to the linear parametric model (cf. (27))

$$\mu(t) = \Omega^{\star\top}(t)\phi(t) + \varepsilon_1(t) + \varepsilon_2(t) \tag{35}$$

where

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

The perturbations $\varepsilon_1(t)$ and $\varepsilon_2(t)$ in (35) are given by

$$\varepsilon_1(t) = 2\left\|[y_1\ y_2]^\top - \epsilon(\lambda^\star(t))\right\| \left.\frac{\mathrm{d}r(\lambda)}{\mathrm{d}\lambda}\right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star(t)}{\mathrm{d}t}, \tag{36}$$
$$\varepsilon_2(t) = 2\left([y_1\ y_2]^\top - \epsilon(\lambda^\star(t))\right)^\top \left.\frac{\mathrm{d}\epsilon(\lambda)}{\mathrm{d}\lambda}\right|_{\lambda^\star(t)} \frac{\mathrm{d}\lambda^\star(t)}{\mathrm{d}t}. \tag{37}$$

The expressions (36), (37) 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 Subsection 4.1 will still work. 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 should not move too fast. The estimation algorithm for $\Omega(t) := [\hat{r}(t) \ \hat{\epsilon}^\top(t)]^\top$ is therefore

$$\dot{\Omega}(t) = \Gamma \left(\frac{\mu(t) - \Omega^\top(t)\phi(t)}{1 + \beta \|\phi(t)\|^2} \right) \phi(t), \Omega(0) = \Omega_0, \beta > 0. \tag{38}$$

With equation (38) 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

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

As in Section 4.2, we neglect the time-varying nature of h_{PF}. Specifically, when we compute the differential $\mathrm{d}h_{\mathrm{PF}}$, 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.

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 3.2. Specifically, we take $\eta_2^{\mathrm{ref}}(t) = c_\parallel \hat{\kappa}(t)$ where $|c_\parallel|$ is chosen to be small. We apply the linear controllers (18) but, since we are doing velocity tracking in the tangential subsystem, we apply the tangential controller $v_\parallel = F_\parallel(\eta_2 - \eta_2^{\mathrm{ref}})$. We emphasize that in the linear feedback law $v_\pitchfork$ from (18), the values of s and $\dot{s}$ from Assumption 4 are used in place of ξ_1 and ξ_2.

5. SIMULATION TESTS

We simulate, in MATLAB the adaptive path following controller from Subsection 4.3 for the ellipse $\sigma(\lambda) = [3\cos(\lambda) \ \sin(\lambda)]^\top$. The robot and adaptive law are initialized at $q(0) = [0 \ \pi/4 \ 1 \ 1 \ 0]^\top$, $v(0) = 0$, $\Omega(0) = [1 \ 0.2 \ 0]^\top$ with $\beta = 0.8$, $\Gamma = 700 \ I_3$, we take the reference signal to be $\eta_2^{\mathrm{ref}}(t) = c_\parallel \hat{\kappa}(\lambda^\star(t))$ with $c_\parallel = 0.5$. We again use a velocity tracking controller for the tangential control and the linear feedback matrices (18) are

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

The resulting motion of the end-effector and mobile base are shown in Figure 2 and the parameter estimates are shown in Figure 3. Figures 2 and 3 reveal that while our approach does not deliver online parameter convergence of the osculating circle, it still solves Problem 2 for the convex curve.

6. FUTURE DIRECTIONS

This preliminary work suggests several avenues of future research. It would be fruitful to implement the proposed controllers on a hardware platform. Direct adaptive control approaches should be investigated as well as a more general class of paths that includes time-varying paths.

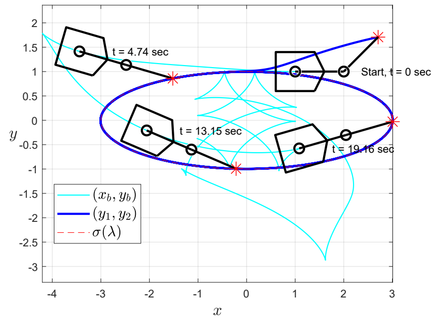

Fig. 2. (y_1, y_2) and (x_b, y_b) for unknown ellipse.

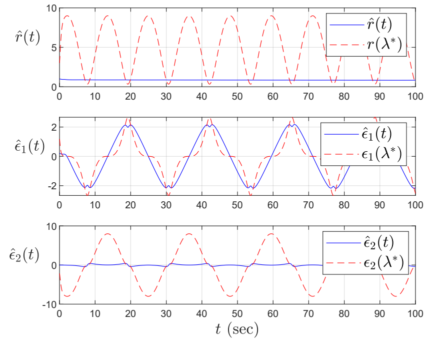

Fig. 3. Osculating circle parameter convergence.

REFERENCES

Barrera Perez, O. (2020). *Adaptive Path Following for an Underactuated Nonholonomic Mobile Manipulator.* Master's thesis, University of Waterloo, http://hdl.handle.net/10012/15496.

Dandach, S.H., Fidan, B., Dasgupta, S., and Anderson, B.D.O. (2009). A continuous time linear adaptive source localization algorithm , robust to persistent drift. *Systems & Control Letters*, 58(1), 7–16.

Hladio, A., Nielsen, C., and Wang, D. (2013). Path following for a class of mechanical systems. *IEEE Trans on Control Systems Technology*, 21(6), 2380–2390.

Li, Y. and Nielsen, C. (2016). Synchronized closed path following for a differential drive and manipulator robot. *IEEE Trans on Control Systems Technology*, 25(2), 704–711.

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

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

Meng, W., Guo, C., and Liu, Y. (2012). Robust adaptive path following for underactuated surface vessels with uncertain dynamics. *Journal of Marine Science and Application*, 11(2), 244–250.

Wang, X.f., Zou, Z.j., Li, T.s., and Luo, W.l. (2010). Adaptive path following controller of underactuated ships using serret-frenet frame. *Journal of Shanghai Jiaotong University (Science)*, 15(3), 334–339.