DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale
in Ingegneria dell'Automazione

Tesi di Laurea Magistrale

Posture regulation for unicycle-like robots with prescribed performance guarantees

Laureanda
Martina Zambelli

Relatore
Ch.mo Prof. Giorgio Picci

Anno Accademico 2012/2013

Dedicato ai miei genitori

Abstract

This thesis focuses on control of nonholonomic system with particular reference to the unicycle-like robots. These are common examples of WMRs (Wheeled Mobile Robots), increasingly present in industrial and service robotics, particularly when flexible motion capabilities are required.

The major objective of this study is to solve the regulation problem for the unicycle model while guaranteeing prescribed performance. Different controllers based on either polar coordinates or time-varying laws are proposed. The main contribution is the combination of the standard control laws (both with polar coordinates and time-varying laws) that allow to achieve posture regulation for the unicycle model, with the prescribed performance control technique that imposes time-varying constraints to the system coordinates. The study also illustrates two different approaches to bind linear or angular coordinates, one based on a particular error transformation, and the other arising from a specific potential function.

Simulations confirm the effectiveness of the proposed solutions.

Acknowledgements

I would like to express my gratitude to Dr. Dimos Dimarogonas, Assistant professor at the Automatic Control Department at KTH, and Dr. Yiannis Karayiannidis, researcher at the Computer Science Department at KTH, for supervising me in this project; their expertise, understanding, suggestions and useful critics added considerably to the depelopement of my thesis work.
My gratitude goes to Professor Giorgio Picci for giving me the opportunity to do my thesis in the Automatic Control Department of KTH.
I also thank Lukas Bühler for his opposition at my presentation at KTH.

My heartfelt thanks to my parents, Fernanda and Alessandro, and to my brother Marco, for their constant support and patience in all these years of university in Padova and in the six months of Erasmus in Stockholm.
Many thanks also to Alberto and to my friends in Verona, Padova, Roma and Stockholm, who enriched my personal journey, making familiar each place and unique each moment.

Vorrei esprimere la mia gratitudine al Dr. Dimos Dimarogonas, Assistant professor presso il Dipartimento di Automazione al KTH, e il Dr. Yiannis Karayiannidis, ricercatore presso il Dipartimento di Informatica al KTH, per la supervisione in questo progetto; la loro esperienza, la comprensione, i suggerimenti e le critiche utili hanno contribuito molto allo sviluppo del mio lavoro di tesi.
La mia gratitudine va al Professor Giorgio Picci per avermi dato l'opportunità di svolgere la mia tesi di laurea presso il Dipartimento di Automazione del KTH.
Ringrazio anche Lukas Bühler per la sua opposizione alla mia presentazione al KTH.

Un sentito grazie ai miei genitori, Fernanda e Alessandro, e a mio fratello Marco, per il loro costante supporto e la pazienza in tutti questi anni di università a Padova e nei sei mesi di Erasmus a Stoccolma.
Molte grazie anche ad Alberto e ai miei amici di Verona, Padova, Roma e Stoccolma, che hanno arricchito il mio percorso personale, rendendo familiare ogni luogo e unico ogni momento.

Contents

Chapter 1

Introduction

Over the past thirty years wheeled mobile robots (WMRs) have become increasingly important in a wide variety of applications such as transportation, security, inspection, planetary exploration, etc. WMRs are increasingly present in industrial and service robotics, particularly when flexible motion capabilities are required. Several mobility configurations (wheel number and type, their location and actuation, single- or multi-body vehicle structure) can be found in the applications. The most common for single-body robots are differential drive and synchro drive (both kinematically equivalent to a unicycle), tricycle or car-like drive, and omnidirectional steering.

Beyond the relevance in applications, the problem of autonomous motion planning and control of WMRs has some theoretical challenges. In particular, these systems are a typical example of nonholonomic mechanisms due to the perfect rolling constraints on the wheel motion (no longitudinal or lateral slipping).

Target problems for WMR are (i) regulation of position and orientation of the WMR to an arbitrary set point, (ii) tracking of a time-varying reference trajectory (the path following problem is a special case), and (iii) enhance robustness including the effects of the dynamic model during the control design.

With regard to the control of nonholonomic systems, one of the technical hurdles often cited is that the regulation problem cannot be solved via a smooth, time-invariant state feedback law due to the implications of Brockett's condition [1]. Brockett's theorem provides a very useful necessary condition for asymptotic stabilizability by continuous feedback. Intuitively, it means that, starting near zero and applying small controls, we must be able to move in all directions. Also, in other words, Brockett's condition states that smooth stabilizability of a driftless regular system requires a number of inputs equal to the number of states. Thus, to reach stabilization of these systems we can use either time-varying or discontinuous controllers.

Many solutions can be found in literature. A very common and simple model to analyze the stabilization of nonholonomic systems is the unicycle. Many solutions in literature refer to this model and it will be exploited also in this thesis.

There exist different approaches to control a nonholonomic system, such as a unicycle-like robot. See for example [2] for discontinuous control, [3] for dynamic feedback linearization technique, [4] for discontinuous backstepping, [5] for an approach involving potential function, [6] for chained form systems control and time-varying point-stabilization, [7] for control with polar coordinate, and also [8, 9] for an overview on nonholonomic systems and control of wheeled robots.

The solution with polar coordinates allows to achieve very natural trajectories for the unicycle vehicle. It is based on the change of variables from the Cartesian (x, y, θ) to the polar (r, γ, δ) coordinates. With these coordinates, control inputs v (the driving linear velocity) and ω (the steering angular velocity) can be designed. This type of control will be analyzed in this thesis, and modification will be made on it in order to achieve better transient performance.

The time-varying control permits to achieve convergence but the obtained vehicle behavior is characterized by noticeable oscillations around the desired position. This is an intrinsic issue for this type of controller, which involves oscillating functions in its design. This thesis will also analyze and modify the time-varying controller in order to achieve better transient performance, specifically for the convergence of the unicycle orientation.

The dynamic feedback linearization technique is used to obtain a linear system starting from the original one. This type of control is briefly recalled in this thesis as a first example of control combined with prescribed performance guarantees.

A different approach is the discontinuous control. It involves a different type of transformation of the nonholonomic system, based on σ-processes. As for other control techniques, this approach has to deal with singularities that are intrinsic either in the controller or in the system to be controlled. This approach is not part of this thesis.

The reader is referred to the literature for further details and other control techniques.

Prescribed performance controllers have recently been proposed in order to guarantee the system transient performance. While usually the problems are solved in the sense of asymptotic convergence of the position errors to zero, with the prescribed performance approach the aim is also to achieve system performance in the transient phase. The reader is referred to the recent literature, e.g. [10], [11], [12], [13].

Prescribed performance guarantees mean that components of the error evolve within predefined regions that are bounded by decaying functions of time. A transformation on the error components is applied. This transformation consists first on modulating the error through the decaying function of time, usually chosen as an exponential function; then a logarithmic function is applied to the modulated error to obtain a transformed error. The aforementioned transformations are based on preset values of convergence rate and overshoot of the response. Proving that the transformed error is bounded, then the error is guaranteed to stay within the predefined limits.

The cited literature is devoted mainly to robot joints or holonomic systems. This thesis applies the concept of prescribed performance on a nonholonomic system, namely the unicycle. Controllers based on polar coordinates are proposed. Prescribed performance are imposed to bind the distance of the unicycle from the desired position, the vehicle orientation, and eventually both the position and the orientation. Time-varying controllers are also designed in order to guarantee prescribed performance on the orientation. In this case, the controller is realized referring to a transformation of the error vector through a rotation matrix. This implies that not all the (Cartesian) coordinates are directly accessible, and the binding procedure is not immediate. The approach is the same used for the orientation bounds in the case of polar coordinates.

This thesis addresses the regulation problem for a mobile robot of the type of the unicycle. Different controllers are designed, in order to guarantee prescribed performance guarantees. The main results are obtained by mean of the Lyapunov analysis.

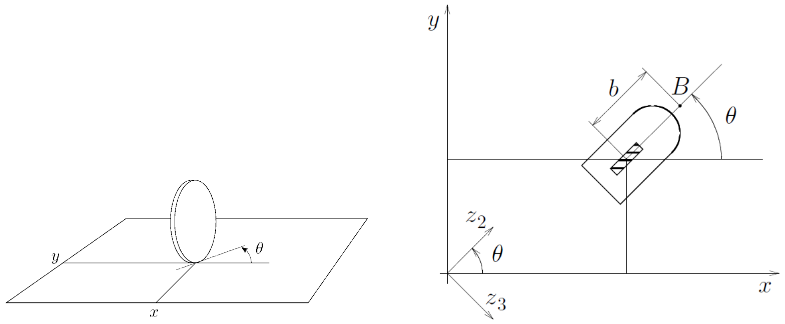
The thesis is organized as follows. In Section 2, we briefly recall the background on which we develop our controllers. In Section 3, we address the regulation by mean of the polar coordinates. We modify the original control law in order to achieve prescribed performance on position, orientation and eventually on both of them at the same time. In Section 4, we design a time-varying control law to stabilize the unicycle to the desired position and orientation; the exploited technique is similar to the trajectory tracking one; we modify the original controller in order to bind the error component into predefined regions. In Section 5 we present some simulations implemented in *ROS* environment. Conclusions follow. All the main proofs can be found in Appendix.

Chapter 2

Preliminaries

2.1 Unicycle model and control overview

Literature reference is for example [7]. A unicycle is a vehicle with a single orientable wheel. The unicycle is the simplest model of a nonholonomic wheeled mobile robot (WMR) and it corresponds to a single wheel rolling on the plane. Consider a disk rolling without slipping on the horizontal plane, while keeping its sagittal plane (the plane that contains the disk) in the vertical direction. The generalized coordinates are $q = (x, y, \theta) \in \mathcal{Q} = \mathbb{R}^2 \times SO^1$: (x, y) are the Cartesian coordinates of the contact point with the ground, measured in the fixed reference frame, and θ is the steering angle, which characterizes the orientation of the disk with respect to the x axis (Fig. 2.1).



(a) Generalized coordinates

(b) Top view of the unicycle

Figure 2.1: Relevant variables for the unicycle

The pure rolling constraint for the disk can be expressed in the Pfaffian form as

$$\dot{x}\sin\theta - \dot{y}\cos\theta = [\sin\theta \quad \cos\theta \quad 0]\dot{q} = 0.$$

This constrain is nonholonomic, because it implies no loss of accessibility in the configuration space of the disk. Thus, the constraints on the wheel state $q = (x, y, \theta)$ are of the type

$$A(q)\dot{q} = 0, \qquad A(q) = \begin{bmatrix} \sin\theta & -\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Considering the matrix

$$G(q) = [g_1(q) \quad g_2(q)] = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}$$

whose columns $g_1(q)$ and $g_2(q)$ are a basis of the null space of the matrix $A(q)$, the kinematic model of the unicycle can be expressed in the following form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{2.1}$$

where the inputs v and ω are, respectively, the *driving velocity* (the linear velocity of the wheel) and the *steering velocity* (the angular velocity of the wheel around the vertical axis). This type of system is said to be *driftless*. Thus, while there are $n = 3$ degree of freedom of the considered system, only $m = 2$ inputs are assumed as available controls.

2.2 Prescribed performance overview

The prescribed performance control technique has been introduced in [14]; see also [11, 12]. The goal of the prescribed performance controller is to guarantee that the error e evolves within certain a priori defined performance bounds defined by a decreasing function and an acceptable overshoot range. The performance bounds are defined by a function $\rho(t)$, called *performance function.*

Given an acceptable overshoot range M, the performance bounds $\forall t \geq 0$ for each element $e_i, \ i = 1, \ldots, n$ of the error are mathematically defined as:

$$\begin{aligned} -M_i\rho_i(t) < e_i < \rho_i(t), \quad & \text{if } e_{0i} \geq 0, \\ -\rho_i(t) < e_i < M_i\rho_i(t), \quad & \text{if } e_{0i} \leq 0, \end{aligned} \tag{2.2}$$

where $e_{0i} = e_i(0), \ i = 1, \ldots, n$, $0 \leq M \leq 1$, and $\rho(t)$ is smooth, bounded, strictly positive decreasing function of time and satisfying $\lim_{t\to\infty}\rho(t) = \rho_\infty > 0$. The performance function can be defined as:

$$\rho(t) = (\rho_0 - \rho_\infty)\exp(-lt) + \rho_\infty.$$

To unify the two control objectives, namely regulation and prescribed transient and steady state behavioral bounds on the error, an error transformation is used. At first the error is modulated by $\rho(t)$, and then a transformation function $T(\cdot)$ is applied.

The modulated error is defined as follows:

$$\hat{e}_i(t) \triangleq \frac{e_i}{\rho_i(t)}. \tag{2.3}$$

Then, the transformed error $\varepsilon(t) \in \mathbb{R}^n$ is defined through transformation functions $T_i : D_{\hat{e}_i} \to \mathbb{R},\ i = 1, \dots, n$:

$$\varepsilon_i(t) \triangleq T_i(\hat{e}_i(t)) \tag{2.4}$$

where the transformations $T_i(\cdot),\ i = 1, \dots, n$ define increasing bijective mappings of the performance domain:

$$D_{\hat{e}_i} \triangleq \{\hat{e}_i : \hat{e}_i \in (-M_i, 1)\} \quad \text{if } e_{0i} \geq 0,$$

$$D_{\hat{e}_i} \triangleq \{\hat{e}_i : \hat{e}_i \in (-1, M_i)\} \quad \text{if } e_{0i} \leq 0.$$

Differentiating (2.4) with respect to time we obtain:

$$\dot{\varepsilon}_i(t) = J_{Ti}(t)[\dot{e}_i + \alpha_i(t)e_i] \tag{2.5}$$

where $J_{Ti}(t)$ and $\alpha_i(t)$ are respectively

$$J_{Ti}(t) \triangleq \frac{\partial T_i}{\partial \hat{e}(t)} \frac{1}{\rho_i(t)} > 0$$

$$\alpha_i(t) \triangleq -\frac{\dot{\rho}_i(t)}{\rho_i(t)} > 0 \text{ with } \lim_{t \to +\infty} \alpha_i(t) = 0.$$

The transformation function is smooth and strictly increasing. Two transformation functions for (2.4) can be defined:

$$\begin{aligned} T_{ai}\left[\hat{e}_i(t)\right] &= \begin{cases} \ln\left(\frac{M_i + \hat{e}_i(t)}{1 - \hat{e}_i(t)}\right), & \text{if } e_{0i} \geq 0 \\ \ln\left(\frac{1 + \hat{e}_i(t)}{M_i - \hat{e}_i(t)}\right), & \text{if } e_{0i} \leq 0 \end{cases} \\ T_{bi}\left[\hat{e}_i(t)\right] &= \begin{cases} \ln\left(\frac{M_i + \hat{e}_i(t)}{M_i(1 - \hat{e}_i(t))}\right), & \text{if } e_{0i} \geq 0 \\ \ln\left(\frac{M_i(1 + \hat{e}_i(t))}{M_i - \hat{e}_i(t)}\right), & \text{if } e_{0i} \leq 0 \end{cases} \end{aligned} \tag{2.6}$$

If from the Lyapunov analysis ε_i is proved bounded ($\varepsilon_i \in \mathcal{L}_\infty$), then the aforementioned transformation is bounded as well and this means that e_i stays within the predefined bounds.
One way is to accommodate a potential of the form

$$\frac{1}{2}||\varepsilon||^2. \tag{2.7}$$

Prescribed performance can also be defined through a different potential of the form

$$\ln\big(\cos \hat{e}_i\big), \tag{2.8}$$

where e_i is the error component to bind.
While in the previous case we begin with the error transformation and then use a potential defined by the square of the transformed error ε, in this case we start from the potential. This approach is particularly convenient in the case of bounds on an angle, for example the orientation of the unicycle. Notice that the potential (2.8) is well defined as

$$\hat{e}_i \in \Big(-\frac{\pi}{2}, \frac{\pi}{2}\Big). \tag{2.9}$$

Employing this potential to define a candidate Lyapunov function V, it is possible to design a control law such that $\dot{V}$ is negative semidefinite. Thus, one can prove that V is bounded and $\ln\big(\cos \hat{e}_i\big)$ as well, hence e_i stays within the defined bounds.

The first thing to be defined is what we consider as *error*. Adopting the aforementioned transformations, the aim is to combine control objective (regulation) while guaranteeing prescribed performance bounds. In this thesis, controllers are designed by mean of polar coordinates and time-varying laws, while applying prescribed performance control concept. The proof of convergence of the error e to zero can be achieved by appropriate Lyapunov functions.

Instrumental results

We briefly present here some results which will be instrumental for the convergence proof of the proposed controllers.
A first critical term to be analyzed is the ratio of the transformation of the error component through the prescribed performance and the error itself:

$$\frac{\varepsilon}{e}$$

We here briefly show that this term turns out to be limited when choosing either T_a with $M = 0$ or T_b for all $M \in (0, 1)$.

Let's consider first $T(\cdot) = T_a(\cdot)$. If we take $M = 0$, then the error e, remaining bounded within prescribed performance bounds (PPB) and does not approach zero, not even asymptotically. Hence, we can have practical convergence, while avoiding the singularity. If $M \neq 0$ then calculating the limit for $e \to 0$ ($e_0 \geq 0$), $\frac{\varepsilon}{e} \to \infty$. The same result is obtained if $e_0 \leq 0$.

Let's consider now $T(\cdot) = T_b(\cdot)$. Applying L'Hôpital's rule, the limit for $e \to 0$ ($e_0 \geq 0$) yields to

$$E \triangleq \frac{1+M}{\rho M}.$$

The same result is obtained if $e_0 \leq 0$.

A graphical representation of this term is drawn in Figure 2.2: it depicts $\frac{\varepsilon}{e}$ with respect to e for a fixed certain time.

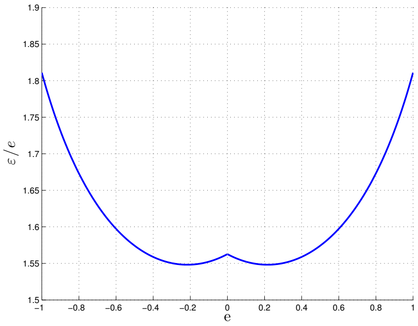


Figure 2.2: The term $\frac{\varepsilon}{e}$ is bounded and it is equal to E as $e = 0$. This plot is obtained setting $\rho_0 = 10,\ \rho_\infty = 0.1,\ L = 2,\ M = 0.8$.

Another relevant expression is the following inequality (see [10]):

$$\varepsilon J e \geq \mu \varepsilon^2 \tag{2.10}$$

with μ a positive constant. This relation is instrumental for convergence proof, in particular see Appendix C.1.

Other motivation

Another motivation for introducing prescribed performance control concept is that for nonholonomic system, as the unicycle model, it is not possible to prove exponential convergence. That is there are no guarantees that the error vanishes with exponential rate. This is related to the fact that the derivative of the Lyapunov function with respect to the time does not have all the coordinates as the Lyapunov function has. This means that a relation of the type $\dot{V} \leq -\nu V$ can not be obtained. Hence, V can not be expressed as

$$V \leq V(0)e^{-\nu t}$$

With prescribed performance approach, a predefined behavior can be achieved, given a maximum overshoot and a desired convergence rate. We design a controller that guarantees the fulfillment of prescribed performance constraints and the convergence to the desired position (thus solving the regulation problem), with the required rate of convergence.

Analysis of the two approaches to impose prescribe performance

This paragraph analyzes the two different approaches that can be followed in order to impose prescribed performance. One begins with the error transformation and then uses a potential defined by the square of the transformed error ε, the other starts from the potential of the form (2.8).

Let us call V_1 and V_2 the defined potentials, and consider $e, \hat{e}, \varepsilon$ as scalar quantities: this is reasonable in view of the controllers we will design in this work. In the first case we have

$$V_1 = \frac{1}{2}\varepsilon^2 \tag{2.11}$$

and in the second case

$$V_2 = -\ln\cos\hat{e}. \tag{2.12}$$

Note that this potential corresponds to the case we apply a transformation of the form

$$\varepsilon = \operatorname{sign}(\hat{e})\sqrt{\ln(\cos\hat{e})^{-2}}$$

As already mentioned, V_2 is particularly convenient when binding angle coordinates. Furthermore, following the first approach that yields to V_1 to bind angle coordinates, leads to find controllers which do not guarantee the convergence of all the variables according to Barbalat lemma.

The fact that the first approach does not solve the problem of regulation while binding an angle coordinate, whereas the second one is successful, is strictly related to the unicycle model and its dynamics.
We remark that this is a nonholonomic system, and the number of the coordinates is greater than the number of control inputs. In particular notice also that the steering velocity ω appears only in $\dot{\gamma}$ in the case of polar coordinate control and only in $\dot{e}_3$ in the case of time-varying control.

Calculating the first derivative with respect to time of the potentials, in the first case we have:

$$\dot{V}_1 = \frac{\partial V_1}{\partial \varepsilon}\dot{\varepsilon} = \varepsilon\dot{\varepsilon} = \varepsilon J(\dot{e} + \alpha e) = \varepsilon\frac{\partial T}{\partial \hat{e}}\frac{1}{\rho}(\dot{e} + \alpha e); \tag{2.13}$$

in the second case:

$$\dot{V}_2 = \frac{\partial V_2}{\partial \hat{e}}\dot{\hat{e}} = \frac{\sin\hat{e}}{\cos\hat{e}}\dot{\hat{e}} = \tan\hat{e}\frac{1}{\rho}(\dot{e} + \alpha e). \tag{2.14}$$

What differentiates the two cases is related to the terms

$$\varepsilon\frac{\partial T}{\partial \hat{e}} \qquad \text{and} \qquad \tan\hat{e}.$$

Also, notice that $\dot{e}$ multiplies in (2.13) and (2.14) respectively

$$\varepsilon \qquad \text{and} \qquad \tan \hat{e}.$$

In the first derivative of the Lyapunov function the following terms appear, respectively in the first and in the second case:

$$\omega\varepsilon \qquad \text{and} \qquad \omega \tan \hat{e}.$$

In order to design control laws that guarantee convergence to the desired posture, ω is defined so as to cancel out some spurious terms deriving from the other coordinates or error components. This implies that the steering velocity depends on terms of the form

$$\tilde{\omega}_1 = \frac{e}{\varepsilon} \qquad \text{and} \qquad \tilde{\omega}_2 = \frac{e}{\tan \hat{e}}$$

in the first and second case respectively.

The convergence proof for the unicycle system is based on Barbalat lemma; in particular we are interested to prove that the second derivative of the Lyapunov function is bounded and thus in particular that $\ddot{V}_1$ and $\ddot{V}_2$ are bounded. We are now taking into consideration the problem of binding the orientation of the unicycle, through $e = \gamma$ in the case of polar coordinates control, or $e = e_3$ in the case of time-varying control; we also recall that ω appears exactly only in the first derivative of those terms. Hence, in order to complete the convergence proof exploiting Barbalat lemma, $\dot{\omega}$ is needed to be bounded.
In other words, to complete the convergence proof, $\dot{\tilde{\omega}}_1$ and $\dot{\tilde{\omega}}_2$ have to be proved bounded. Calculating these first derivatives, in the first case we have

$$\dot{\tilde{\omega}}_1 = \frac{d}{dt}\frac{e}{\varepsilon} = \frac{\dot{e}}{\varepsilon} - e\frac{\dot{\varepsilon}}{\varepsilon^2} = \dot{e}\left[\frac{1}{\varepsilon} - \frac{eJ}{\varepsilon^2}\right] - \alpha J\left(\frac{e}{\varepsilon}\right)^2 \tag{2.15}$$

while in the second case

$$\dot{\tilde{\omega}}_2 = \frac{d}{dt}\frac{e}{\tan \hat{e}} = \frac{\dot{e}}{\tan \hat{e}} - e\dot{\hat{e}}\frac{1 + \tan^2 \hat{e}}{\tan^2 \hat{e}} = \dot{e}\left[-\hat{e} + \frac{\tan \hat{e} - \hat{e}}{\tan^2 \hat{e}}\right] - \alpha e\hat{e}. \tag{2.16}$$

In the first case, the term in the squared brackets is unbounded, and $\dot{\tilde{\omega}}_1$ as well. In the second case, all the terms are bounded and in particular the term $\frac{\tan \hat{e} - \hat{e}}{\tan^2 \hat{e}}$ is bounded as long as $\hat{e} \neq 0$ and

$$\lim_{\hat{e} \to 0} \frac{\tan \hat{e} - \hat{e}}{\tan^2 \hat{e}} = 0.$$

Hence, only $\dot{\tilde{\omega}}_2$ is proved bounded, and thus only the second approach is a feasible way to bind an angle coordinate by prescribed performance.

Details of this reasoning applied to the polar coordinates case and to the time-varying control one, can be found in Appendix A.

2.3 First example: Dynamic Feedback Linearization

This section introduces a first example of application of prescribed performance control to the DFL control technique that solves the regulation problem of the unicycle.

The reader is referred to [9] for a more detailed treatise of *DFL* technique. The unicycle system can always be transformed via feedback into simple integrators (input- output linearization and decoupling). The choice of the linearizing outputs is not unique.

Notice that in the case of linear systems, it is possible to prove exponential convergence. Thus, in this case prescribed performance control does not improve the performance of the obtained controller, unless the system is affected by disturbances.

Define the linearizing output vector as $\eta = (x, y)$ and introduce an integrator (whose state is denoted by ξ) on the linear velocity input

$$v = \xi, \quad \dot{\xi} = a$$

being a the *linear acceleration*, considered as new input.
Provided that $\xi \neq 0$, the unicycle can be expressed as a linear system.
In the new coordinates it is

$$\begin{aligned} z_1 &= x \\ z_2 &= y \\ z_3 &= \dot{x} \\ z_4 &= \dot{y} \end{aligned} \quad \Rightarrow \quad \begin{cases} \ddot{z}_1 = u_1 \\ \ddot{z}_2 = u_2 \end{cases}$$

and a PD controller on the Cartesian error

$$\begin{aligned} u_1 &= -k_{p1}x - k_{d1}\dot{x} \\ u_2 &= -k_{p2}y - k_{d2}\dot{y} \end{aligned} \tag{2.17}$$

can yield exponential convergence, while $k_{p1}, k_{p2}, k_{d1}, k_{d2}$ are positive constants.

So as to have a more compact notation, define

$$\begin{aligned} e &= \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad K_p = \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix} \quad K_v = \begin{bmatrix} k_{d1} & 0 \\ 0 & k_{d2} \end{bmatrix} \\ \varepsilon &= \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \qquad\qquad K_\varepsilon = \begin{bmatrix} k_{\varepsilon 1} & 0 \\ 0 & k_{\varepsilon 2} \end{bmatrix} \quad J_T = \begin{bmatrix} J_{T1} & 0 \\ 0 & J_{T2} \end{bmatrix} \end{aligned} \tag{2.18}$$

where $K_p, K_v, K_\varepsilon, J_T$ are positive definite matrices. Thus,

$$\ddot{z} = \begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \ddot{e} = u.$$

In order to introduce Prescribed performance, define the control law as

$$u = -K_v\big(\dot{e} + \alpha(t)e\big) - K_\varepsilon J_T\varepsilon - \dot{\alpha}(t)e - \alpha(t)\dot{e} \tag{2.19}$$

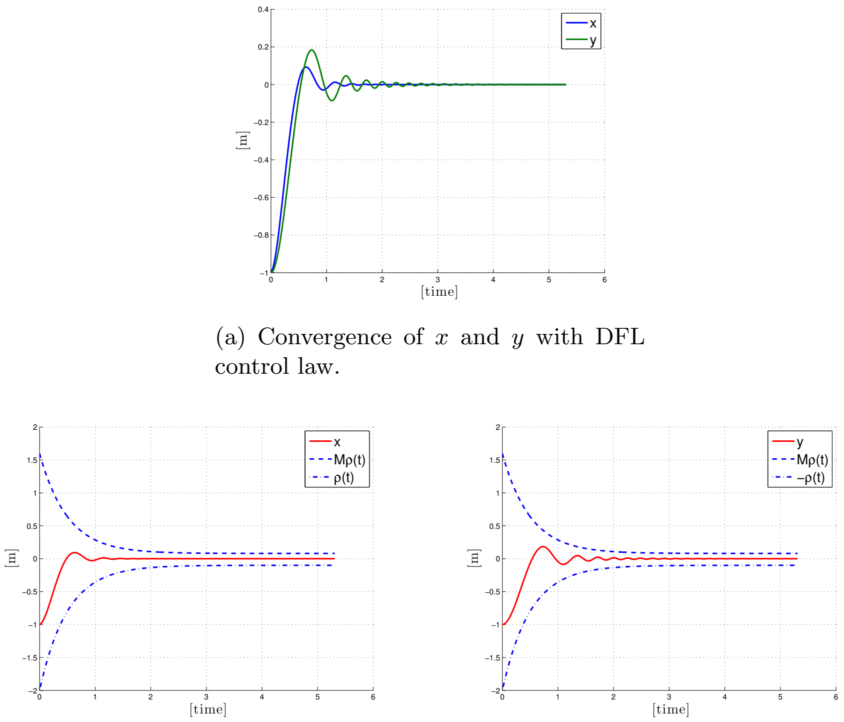
and consider the Lyapunov function

$$V = \frac{1}{2}\big|\big|\dot{e} + \alpha(t)e\big|\big|^2 + \frac{1}{2}\varepsilon^T K_\varepsilon \varepsilon. \tag{2.20}$$

Differentiating (2.20) with respect to time, substituting the control law (2.19) and operating some cancellations we have

$$\frac{dV}{dt} = -\dot{e}^T K_v \dot{e} - e^T \alpha(t)^T K_v \alpha(t) e \leq 0. \tag{2.21}$$

Exploiting Barbalat Lemma, it is possible to prove asymptotic convergence of $(e, \dot{e}, \varepsilon)$ to zero. Details can be found in Appendix B.
Figure 2.3.a shows the convergence of e, that is of x and y, to zero, while Figure 2.3.b-2.3.c display x and y together with their bounds, pointing out that the prescribed performance limits are fulfilled.



(a) Convergence of x and y with DFL control law.

(b) x stays within prescribed bounds.

(c) y stays within prescribed bounds.

Figure 2.3: Results of Matlab simulation of the designed controller.

Chapter 3

Control with Polar Coordinates

3.1 Control with polar coordinates

A convenient way to formulate the regulation problem for a unicycle is to express it in polar coordinates. The reader is referred to [7].
Consider then the following change of variables:

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \gamma &= \operatorname{atan2}(y, x) - \theta + \pi \\ \delta &= \gamma + \theta. \end{aligned} \tag{3.1}$$

A graphical representation is illustrated in Fig. 3.1.

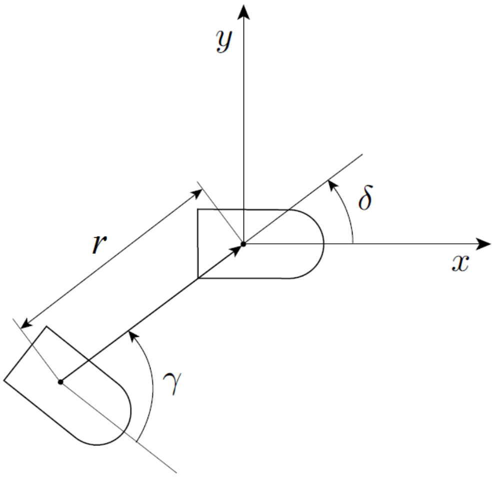


Figure 3.1: Regulation and polar coordinates for the unicycle

The first coordinate, r, represents the distance of the unicycle from the origin of the fixed world Cartesian frame, or in other words the measure of the pointing vector individuated from the hub of the vehicle and the origin of the Cartesian reference; the second one, γ, is the angle between the forward direction vector of the unicycle and the pointing vector; the third coordinate, δ, is the angle between the x-axis and the pointing vector.
In these coordinates, the kinematic model is expressed as:

$$\begin{aligned} \dot{r} &= -v\cos\gamma \\ \dot{\gamma} &= \frac{\sin\gamma}{r}v - \omega \\ \dot{\delta} &= \frac{\sin\gamma}{r}v, \end{aligned} \tag{3.2}$$

and the control law can be defined as

$$\begin{aligned} v &= k_1 r\cos\gamma \\ \omega &= k_2\gamma + k_1\frac{\sin\gamma\cos\gamma}{\gamma}(\gamma + k_3\delta), \end{aligned} \tag{3.3}$$

where $k_1 > 0$, $k_2 > 0$, $k_3 > 0$. The control inputs are bounded and well defined for all the values of γ.

Notice that there is a singularity for $r = 0$. Specifically, the coordinates γ and δ are not defined for $x = y = 0$. Also, the control law, once mapped back to the original coordinates, is discontinuous at the origin of the configuration space, and the behavior of the controlled system is not continuous with respect to the initial state.

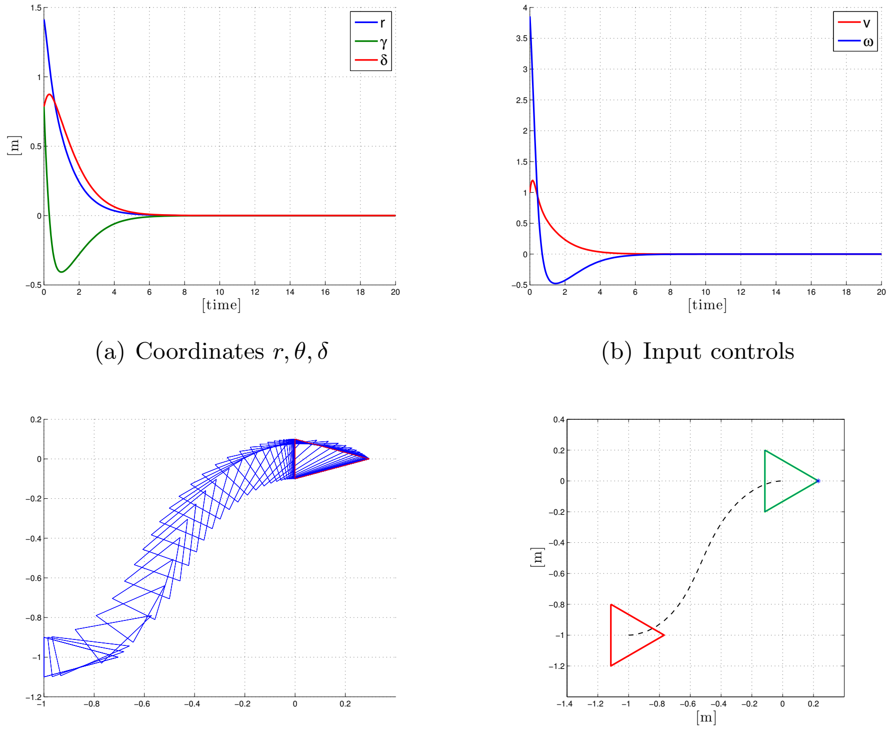
The Lyapunov function $V = \frac{1}{2}(r^2 + \gamma^2 + k_3\delta^2)$ allows to conclude that the kinematic model (3.2) under the action of the given control law asymptotically converges to the desired configuration $(r, \gamma, \delta)^T = (0, 0, 0)^T$. In fact, differentiating V with respect to the time and considering the closed-loop system with control inputs (3.3), the obtained $\dot{V}$ is non-increasing:

$$\dot{V} = -k_1 r^2\cos^2\gamma - k_2\gamma^2 \leq 0.$$

Observing the form of $\dot{V}$, notice that γ is guaranteed to be bounded and convergent to zero. Thus the cosine multiplying r^2 converges to one, hence also r is guaranteed to converge to zero.
More analytically, being $\dot{V} \leq 0$, the state is bounded in norm, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to a limit value. Exploiting Barbalat lemma, it is possible to conclude that $\dot{V}(t)$ tends to zero and thus also r and γ do. Also, analyzing the closed-loop system, $\dot{r}$ and $\dot{\delta}$ converge to zero, δ converges to some finite limit $\bar{\delta}$ while $\dot{\gamma}$ tends to the finite limit $-k_1k_3\bar{\delta}$ and is uniformly continuous. This finite limit must be zero according to Barbalat Lemma and thus also δ converges to zero.

Matlab simulation

In Figure 3.2 we report the unicycle behavior under the control law designed by mean of polar coordinates reference system.
One can notice that all the coordinates converge and we obtain a natural movement for the vehicle.



(a) Coordinates r, θ, δ

(b) Input controls

(c) Unicycle movement

(d) Unicycle trajectory

Figure 3.2: Unicycle behavior with initial conditions $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad) and $k_1 = 1,\ k_2 = 3,\ k_3 = 2$.

3.2 Prescribed performance on the distance vector

In this section, we define a control law for the posture regulation of the unicycle, utilizing polar coordinates while guaranteeing prescribed performance for the convergence of the first coordinate.

We define the error as $e = r$ and its transformation $\varepsilon(\hat{e}) = T(\hat{e})$. Consider the Lyapunov function

$$V = \frac{1}{2}(\varepsilon^2 + \gamma^2 + k_3\delta^2). \tag{3.4}$$

Define the control law

$$\begin{aligned} v &= k_1\varepsilon\cos\gamma + k_3\alpha(t)e\cos\gamma \\ \omega &= k_2\gamma + \Big(k_1\frac{\varepsilon}{e} + k_3\alpha(t)\Big)\frac{\sin\gamma\cos\gamma}{\gamma}(\gamma + k_3\delta) + k_3\varepsilon J\alpha(t)e\frac{\sin^2\gamma}{\gamma}. \end{aligned} \tag{3.5}$$

Then the first derivative of the Lyapunov function wrt time is:

$$\dot{V} = -k_1\varepsilon^2 J_T\cos^2\gamma - k_2\gamma^2 - \varepsilon J\alpha(t)e(k_3 - 1). \tag{3.6}$$

Exploiting the relation $\varepsilon Je \geq \mu\varepsilon^2$ with $\mu > 0$, and provided that $k_3 \geq 1$, it is possible to conclude that $\dot{V}$ is non-increasing. Details can be found in Appendix C.1.

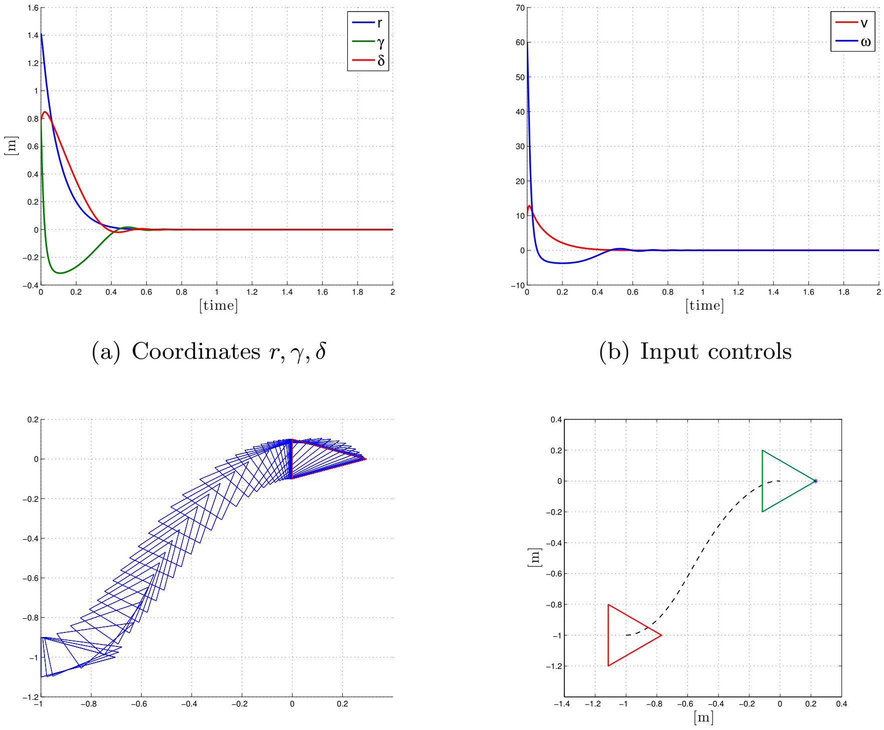
Proposition 3.1 *Consider the polar coordinate description (3.2) of the unicycle and the feedback control (3.5) with k_1, k_2, k_3 positive constants and $k_3 \geq 1$. The closed-loop system (3.2)-(3.5) is then globally asymptotically driven to the posture $(r, \gamma, \delta) = (0, 0, 0)$. Also, the polar coordinate r respects the prescribed limits.*

Proof. The proof can be found in Appendix C.1.

MatLab simulations

Simulations confirm the analysis developed in the previous paragraph. In Fig. 3.3 is reported the unicycle behavior with initial conditions $(x_0, y_0, \theta_0) = (-1, -1, 0)$(m,m,rad).

From Figure 3.3.a one can notice that all the polar coordinates converge to the desired values, and the convergence is faster than in the previous case. The input signals vanish in short time as well, although higher values are required for the initial steering velocity. However, this fact is related to the control coefficients k_1, k_2, k_3: setting these coefficients equal to those used for the original controller simulation, the resulting behavior is less regular, but still faster than the original one. In other words, on equal terms, the achieved performance of the modified control law is faster than the original one. Refer to Figure 3.5 for simulation comparison. In Figure 3.3.d a view of the vehicle trajectory is depicted.



(a) Coordinates r, γ, δ

(b) Input controls

(c) Unicycle movement

(d) Unicycle trajectory

Figure 3.3: Unicycle behavior with initial conditions $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad) and $k_1 = 0.02,\ k_2 = 20,\ k_3 = 2$. PP bounds are imposed on r

Figure 3.4 shows that with the designed control law the first coordinate r evolves within the *prescribed performance* predefined bounds, modulated by the function $\rho(t)$.

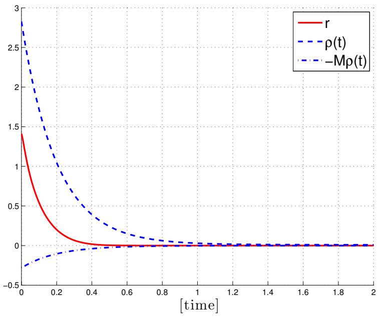
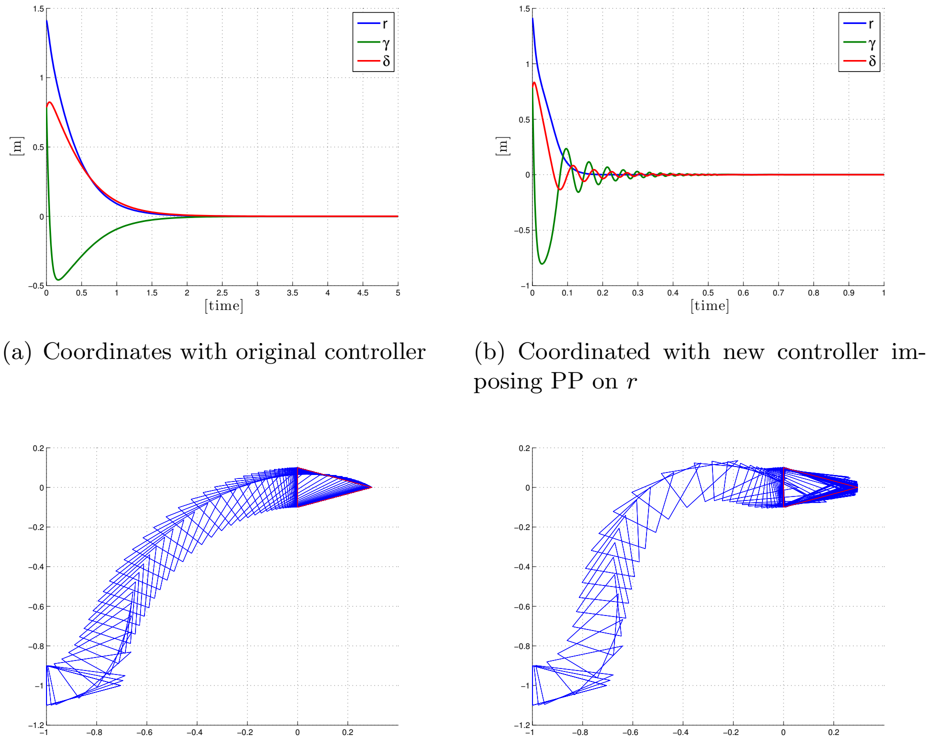


Figure 3.4: The error $e = r$ stays within prescribed performance bounds.



(a) Coordinates with original controller

(b) Coordinated with new controller imposing PP on r

(c) Unicycle movement with original controller

(d) Unicycle movement with new controller imposing PP on r

Figure 3.5: Simulation comparison: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad), $k_1 = 3,\ k_2 = 20,\ k_3 = 5$.

3.3 Bounds on the orientation

This section explores the problem of putting prescribed performance bounds on the angles γ and/or δ.
First of all, we notice that while putting (PP) bounds on r is a reasonable and intuitive way to proceed, imposing bounds on the angles needs some more comments. We will first discuss about the angle γ and then we will briefly comment the case with δ.

3.3.1 Bounds on γ or δ

We recall that γ is the angle that the robot's frame makes with the environment (fixed) frame, i.e. the angle between the vehicle direction and the pointing vector that connects the unicycle position to the origin of the fixed frame.

From a physical point of view, imposing bounds on γ for example in order to keep it in $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ implies also that the vehicle has constraints in its motion. In particular, if γ is constrained to stay in $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, the vehicle must depart from the 2nd or 3rd quadrant, so that the motion can satisfy the constraints on γ while exploiting a linear velocity which makes it go forward. Also, we have to take care of δ in order to make it converge to zero as well.

From a mathematical point of view, trying to apply the prescribed performance transformation T to the angle coordinates and carrying on an analysis similar to that presented in the previous sections, yields to an unbounded second derivative of γ (or δ). This fact does not allow to conclude for $\dot{\gamma}$ to be uniformly continuous, thus to prove the convergence (exploiting Barbalat Lemma) of $\dot{\gamma}$ to zero, and eventually the convergence of δ to zero. This analysis is given in Appendix C.2.

Similarly to what said for γ, bounds on δ do not find a trivial physical motivation, and the effect is to limit the movement of the vehicle.
Analytical details can be found in Appendix C.2.

3.3.2 Overview of a practical possible solution

A reasonable approach to bind angle coordinates implies that we consider some precise configuration and we have an a priori knowledge of the initial configuration.
Consider for example the following bounds:

$$\gamma \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \quad \bar{\gamma} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \quad \delta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$

where $\bar{\gamma} \triangleq \gamma - \pi$. We can split this constrains as follows:

$$\begin{aligned} A', A'' : \gamma, \bar{\gamma} \in \big(-\frac{\pi}{2}, 0\big) \quad \cup \quad B', B'' : \gamma, \bar{\gamma} \in \big(0, \frac{\pi}{2}\big) \\ C : \delta \in \big(-\frac{\pi}{2}, 0\big) \quad \cup \quad D : \delta \in \big(0, \frac{\pi}{2}\big) \end{aligned} \tag{3.7}$$

and design a driving velocity input that can either drive the vehicle forward (v_F) or backward (v_B).
Notice that C means that the vehicle is in the 2nd or 4th quadrant, while D means that the vehicle is in the 1st or 3rd.

Thus, we have 16 possible feasible combinations:
– C,A',v_F: 2rd quadrant, forward motion;
– C,A',v_B: 4th quadrant, backward motion;
– C,A",v_B: 2rd quadrant, backward motion;
– C,A",v_F: 4th quadrant, forward motion;
– C,B',v_F: 2rd quadrant, forward motion;
– C,B',v_B: 4th quadrant, backward motion;
– C,B",v_B: 2rd quadrant, backward motion;
– C,B",v_F: 4th quadrant, forward motion;
– D,A',v_F: 3rd quadrant, forward motion;
– D,A',v_B: 1st quadrant, backward motion;
– D,A",v_B: 3rd quadrant, backward motion;
– D,A",v_F: 1st quadrant, forward motion;
– D,B',v_F: 3rd quadrant, forward motion;
– D,B',v_B: 1st quadrant, backward motion;
– D,B",v_B: 3rd quadrant, backward motion;
– D,B",v_F: 1st quadrant, forward motion;

Notice also that not all of this configurations allow to have a final orientation $\theta = 0$: e.g. case (D,B",v_B) where the final vehicle orientation will be $\theta = \pi$.

We remark that prescribed performance bounds on the angle variable (only on γ, only on δ or on both) set by mean of the transformation $T(\hat{e})$, lead either to find controllers which do not guarantee the convergence of all the variables, or to have positive terms in the first derivative of the Lyapunov function.

3.3.3 Bounds on the angle γ through a different Lyapunov function

As already mentioned in the Preliminaries section, another way to impose prescribed performance is to use a different Lyapunov function of the form (2.8). This approach is particularly convenient when dealing with angle coordinates.
We now set bounds on γ, and hence indirectly on the orientation of the

unicycle, following this different approach.
First we take the candidate Lyapunov function defined as

$$V = \frac{1}{2}r^2 - \ln\cos\gamma + \frac{k_3}{2}\delta^2. \tag{3.8}$$

This function is positive definite for a specified range of value of γ, namely $\gamma \in (-\pi/2, \pi/2)$. This means that if the vehicle departs from a position with $\gamma \in (-\pi/2, \pi/2)$, then this angle coordinate will evolve within the predefined set of value, and it will never leave it.
Define the control input as

$$\begin{aligned} v &= k_1 r \cos\gamma \\ \omega &= k_2 \tan\gamma + k_1\big(k_3\delta + \tan\gamma\big)\cos^2\gamma \end{aligned} \tag{3.9}$$

Then the first derivative of (3.8) is negative semidefinite:

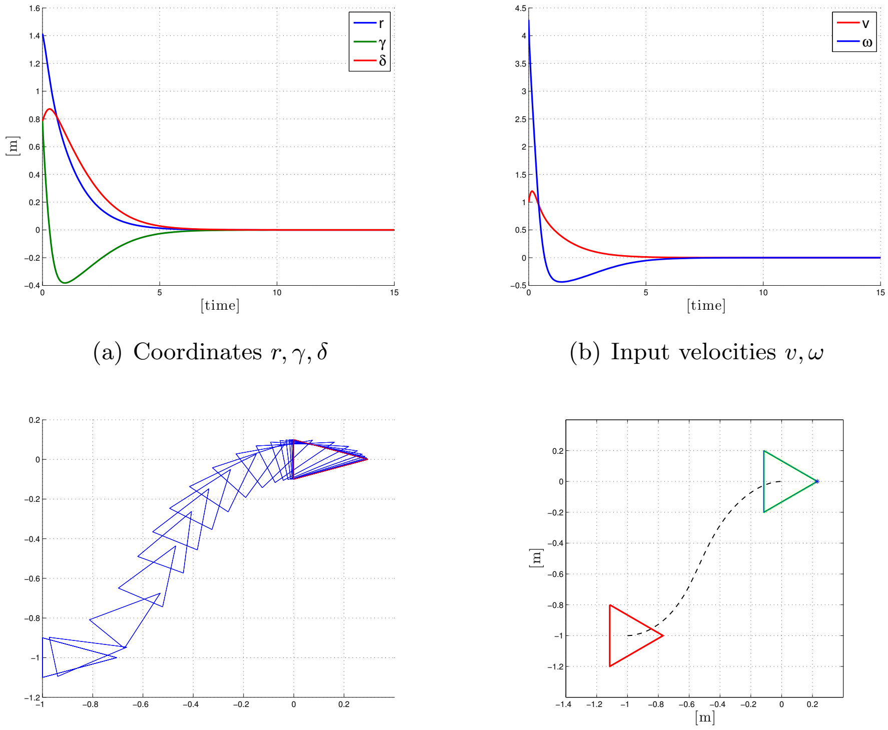
$$\dot{V} = -k_1 r^2 \cos^2\gamma - k_2 \tan^2\gamma \leq 0. \tag{3.10}$$

Proposition 3.2 *Consider the polar coordinate description (3.2) of the unicycle and the feedback control (3.9) with k_1, k_2, k_3 positive constants. The closed-loop system (3.2)-(3.9) is then globally asymptotically driven to the posture $(r, \gamma, \delta) = (0, 0, 0)$. Also, the polar coordinate γ respects the prescribed limits.*

Note that being $\gamma \in (-\pi/2, \pi/2)$, the cosine in $\dot{V}$ is never zero.
The proof for the coordinates convergence can be carried on adopting LaSalle theorem and Barbalat lemma, as for the previous designed controllers. The control law (3.9) designed with the particular Lyapunov function defined by (3.8) guarantees that the angle coordinate γ stays within the predefined set $(-\pi/2, \pi/2)$, as γ_0 is chosen in this range of values.
Details and a sketch of the proof of Proposition 3.2 can be found in Appendix C.3.

Matlab simulation

Matlab simulations are reported in Figures 3.6-3.7. Note that all the coordinates converge to the desired position and the vehicle performs a natural maneuver. Also the input controllers vanish in short time and remain bounded. Moreover, the coordinate γ never leaves the defined set of values, namely $(-\pi/2, \pi/2)$.



(a) Coordinates r, γ, δ

(b) Input velocities v, ω

(c) Unicycle behavior

(d) Unicycle trajectory

Figure 3.6: Unicycle behavior. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3) = (1, 3, 2)$.

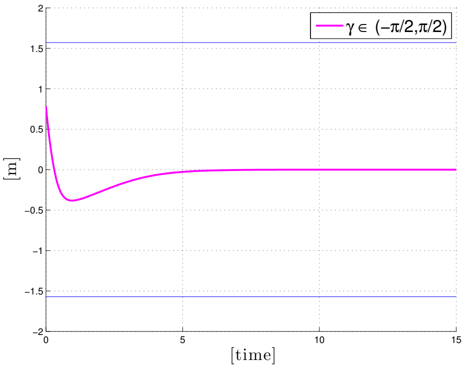


Figure 3.7: The coordinate γ never leaves the defined set of values, namely $(-\pi/2, \pi/2)$.

Time-varying bounds on γ

In order to achieve faster convergence, we define time-varying bounds on the orientation. We now introduce a time-varying positive transformation function, namely $\rho(t)$, such that

$$\gamma \quad \mapsto \quad \hat{\gamma} = \frac{\gamma}{\rho(t)}.$$

The modulating function is defined, in the same way as in the prescribed performance analysis, as a smooth, bounded, strictly positive decreasing function of time and satisfying $\lim_{t\to\infty} \rho(t) = \rho_\infty > 0$:

$$\rho(t) = (\rho_0 - \rho_\infty)\exp(-Lt) + \rho_\infty. \tag{3.11}$$

To unify the convergence and the time-varying bounds we consider a Lyapunov function, defined as in the previous paragraph but depending on $\hat{\gamma}$ instead of γ:

$$V = \frac{1}{2}r^2 - \ln\cos\hat{\gamma} + \frac{k_3}{2}\delta^2, \quad \text{with } \gamma \in \left(-\frac{\pi}{2}\rho(t), \frac{\pi}{2}\rho(t)\right) \tag{3.12}$$

This function is positive definite in the defined set of values that depends on time. This fact permits to define more strict bounds, that evolve together with the coordinate γ.
Define the control velocity input as

$$\begin{aligned} v &= k_1 r\cos\gamma \\ \omega &= k_2\tan\hat{\gamma} + \gamma\alpha(t) + k_1\rho(t)\Big(k_3\delta + \frac{1}{\rho(t)}\tan\gamma\Big)\cos\gamma\cos\hat{\gamma}\frac{\sin\gamma}{\sin\hat{\gamma}} \end{aligned} \tag{3.13}$$

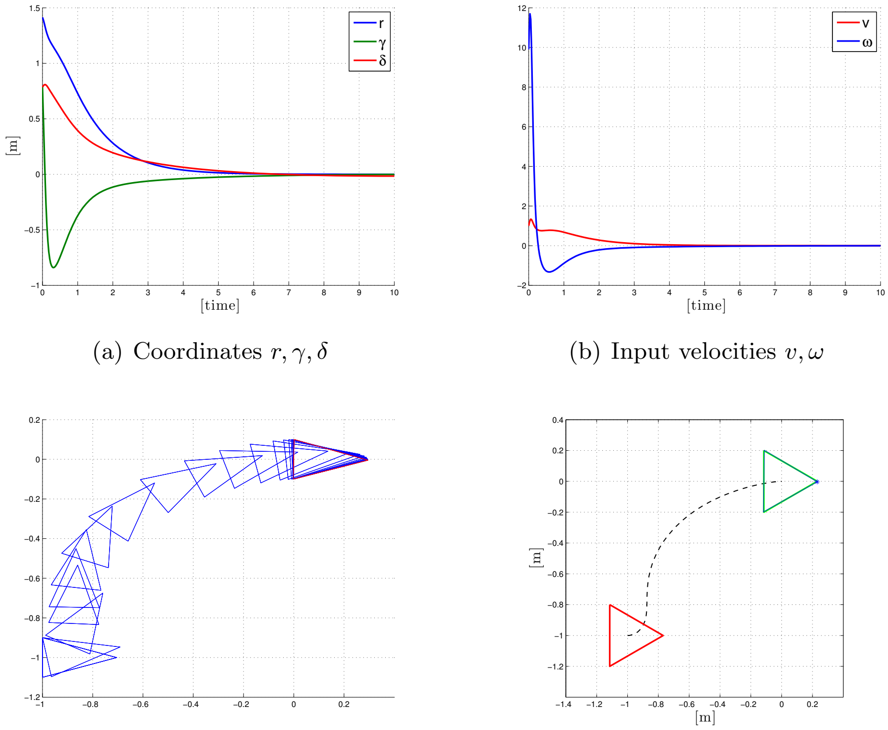
Then the first derivative of (3.12) wrt to time is negative semidefinite:

$$\dot{V} = -k_1 r^2\cos^2\gamma - \frac{k_2}{\rho(t)}\tan^2\hat{\gamma} \leq 0. \tag{3.14}$$

The control inputs are bounded and well defined. Details can be found in Appendix C.4.
The proof for the convergence of the coordinates can be carried on exploiting LaSalle theorem and Barbalat Lemma. Proving that V is bounded allows also to conclude that $\ln\cos\hat{\gamma}$ is bounded and hence γ respects the predefined limits.

Matlab simulation

Matlab simulations are reported in Figures 3.8-3.9.
One can notice (from Fig. 3.8.a) that the convergence of the γ coordinate evolves faster than in the previous case, although its maximum oscillating amplitude is bigger. We also notice that the convergence of δ is slower in this case, and the control requires higher initial values for the steering velocity input. These facts are related to the modulating function, which affects also the evolution of ω. Moreover, since γ is vanishing faster, the coordinate δ converges later to zero in order to achieve the desired orientation $\theta_d = 0$.
The performances are also affected by the parameters. Tuning the constant parameters k_1, k_2, k_3 and especially modifying the requirements for the time-varying bounds, that is replacing ρ_0, ρ_∞, L with other values, one can achieve different behaviors of the unicycle.



(a) Coordinates r, γ, δ

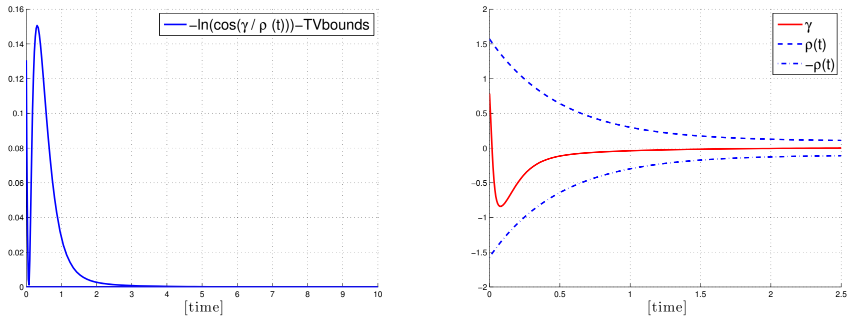
(b) Input velocities v, ω

(c) Unicycle behavior

(d) Unicycle trajectory

Figure 3.8: Unicycle behavior. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3) = (1, 0.05, 7)$, $\rho_0 = \pi/2, \rho_\infty = 0.1, L = 2$.

Figure 3.9 shows the bounded coordinate behavior. Picture 3.9.a plots the evolution of $-\ln \cos \hat{\gamma}$ in the time: we are confirmed that this part of Lyapunov function converges to zero and also has a fast dynamics, so that the bounded coordinate can quickly reach convergence. Picture 3.9.b shows γ evolution in the time together with the bounds defined by the modulating function $\rho(t)$, pointing out that this bounds are fully satisfied.



(a) γ by the logarithmic function $-\ln \cos \hat{\gamma}$ (b) γ stays in the bounds designed by the modulating function $\rho(t)$

Figure 3.9: Bounds. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3) = (1, 0.05, 7)$, $\rho_0 = \pi/2, \rho_\infty = 0.1, L = 2$.

3.4 Bounds on both radial and angle coordinate

In this paragraph we combine the control laws defined in the previous sections. The first control law (defined by the equations in (3.5)) allows to set prescribed performance bounds on r while the second one (defined by the equations in (3.13)) permits to bind the angle γ and hence, indirectly, the orientation of the unicycle ($\theta = \delta - \gamma$).

The subscript r will be used for the terms referring to the first polar coordinate transformed by mean of *prescribed performance* bounds, and the subscript γ for the terms referring to the homonym angle coordinate, transformed as shown in the previous section.
Let's consider the transformation for the first coordinate

$$r \;\mapsto\; \varepsilon(\hat{e}) = T(\hat{e}), \quad \hat{e} = \frac{r}{\rho_r(t)}, \quad \rho_r(t) = (\rho_{0_r} - \rho_{\infty_r}) \exp(-L_r t) + \rho_{\infty_r}$$

defined by *prescribed performance* through the modulating function $\rho_r(t)$, and the transformation for the second coordinate

$$\gamma \;\mapsto\; \hat{\gamma} = \frac{\gamma}{\rho_\gamma(t)}, \quad \rho_\gamma(t) = (\rho_{0_\gamma} - \rho_{\infty_\gamma}) \exp(-L_\gamma t) + \rho_{\infty_\gamma}$$

with the modulating function $\rho_\gamma(t)$.

We define the candidate Lyapunov function, inspired both by (3.4) and (3.12), as

$$V = \frac{1}{2}\varepsilon^2 - \ln\cos\hat{\gamma} + \frac{k_3}{2}\delta^2. \tag{3.15}$$

This function is positive definite in a set of value that depends on time:

$$\gamma \in \Big(-\frac{\pi}{2}\rho_\gamma(t), \frac{\pi}{2}\rho_\gamma(t)\Big)$$

We design the control law as

$$\begin{aligned} v &= k_1 \cos\gamma \varepsilon_r J_r + k_3 \alpha_r r \cos\gamma; \\ \omega &= k_2 \tan\hat{\gamma} + \alpha_\gamma \gamma + \rho_\gamma \Big(k_3\delta + \frac{\tan\hat{\gamma}}{\rho_\gamma}\Big)\Big(k_1 J_r \frac{\varepsilon_r}{r} + \alpha_r\Big) \cos\hat{\gamma}\cos\gamma \frac{\sin\gamma}{\sin\hat{\gamma}} \end{aligned} \tag{3.16}$$

Differentiating V wrt to time and substituting the defined controllers we obtain

$$\dot{V} = -k_1 \varepsilon^2 J_r^2 \cos^2\gamma - \frac{k_2}{\rho_\gamma}\tan^2\hat{\gamma} \leq 0 \tag{3.17}$$

where the time dependence of ρ_γ from the time is implied, that is $\rho_\gamma = \rho_\gamma(t)$.

The control inputs (3.16) are well defined and bounded, as shown in Appendix C.4.

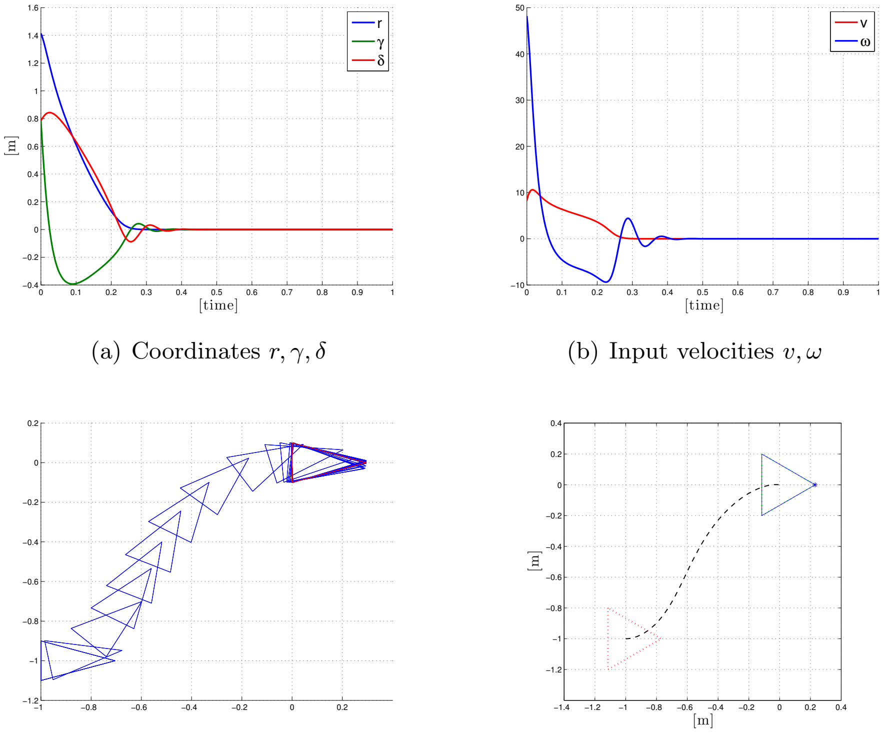
The control law (3.16), designed with the particular Lyapunov function defined by (3.15) by means also of the *prescribed performance* transformation for the first polar coordinate r and the time-varying transformation through $\rho_\gamma(t)$ of the first angle coordinate, guarantees the convergence to the desired position and orientation while satisfying the predefined bounds. Specifically, proving that V is bounded, it is possible to conclude that ε (as well as the transformation $T(\hat{e})$) and $\ln\cos\hat{\gamma}$ are also bounded. Hence, r and γ respect the predefined limits.

Proposition 3.3 *Consider the polar coordinate description (3.2) of the unicycle and the feedback control (3.16) with k_1, k_2, k_3 positive constants. The closed-loop system (3.2)-(3.16) is then globally asymptotically driven to the posture $(r, \gamma, \delta) = (0, 0, 0)$. Also, the polar coordinates r and γ respect the prescribed limits.*

Details and proof of convergence for Proposition 3.3 can be found in Appendix C.5.

Matlab simulation

Matlab simulations are reported in Figures 3.10-3.11.



(a) Coordinates r, γ, δ

(b) Input velocities v, ω

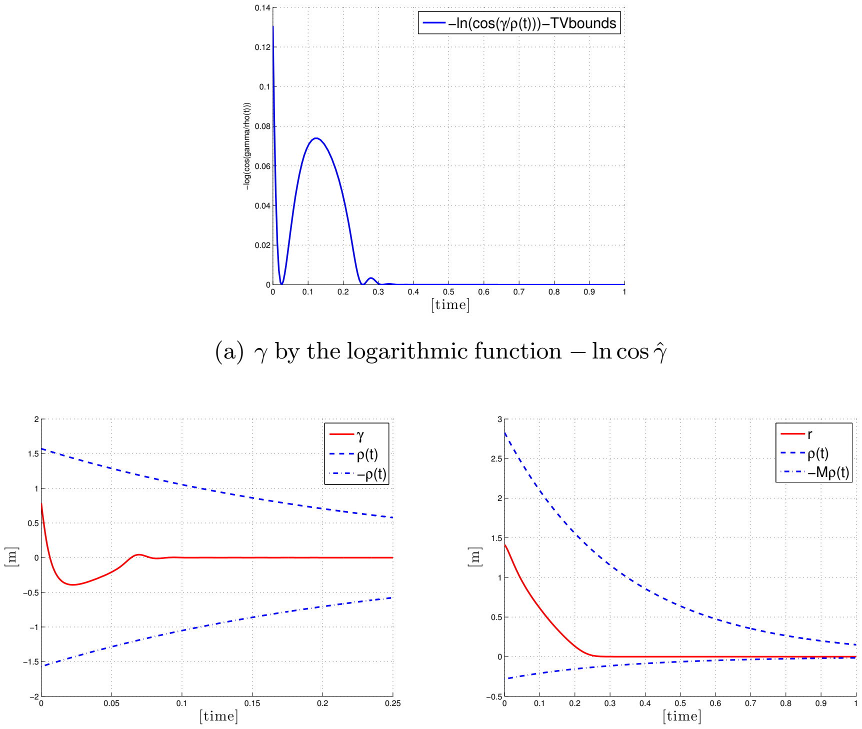
(c) Unicycle behavior

(d) Unicycle trajectory

Figure 3.10: Unicycle behavior. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3) = (1, 20, 2)$, $\rho_{0_\gamma} = \pi/2$, $\rho_{\infty_\gamma} = 0.01$, $L_\gamma = 4$; $\rho_{0_r} = 2|r_0|$, $\rho_{\infty_r} = 0.01$, $L_r = 3$, $M_r = 0.1$;

One can easily observe that the convergence is much faster than in the previous cases. All the coordinates converge faster; the maximum oscilating amplitude of γ is bigger in this case with respect to the behavior obtained with the control law defined in (3.5), but smaller with respect to the case obtained with the control law (3.13). This controller requires also higher initial values for the steering velocity input compared to the other cases. This is due also to the parameters and to the modulating function, which affects the steering velocity definition.

Figure 3.11 shows the bounded coordinate behavior. Picture 3.11.a plot the evolution of $-\ln\cos\hat{\gamma}$ in the time: we are confirmed that this part of Lyapunov function converges to zero and also has a fast dynamics, so that the bounded coordinate can quickly reach convergence. Pictures 3.11.b,3.11.c show γ and r evolution respectively, together with the bounds defined by the modulating functions $\rho_\gamma(t)$ and $\rho_r(t)$, pointing out that these bounds are fully satisfied.



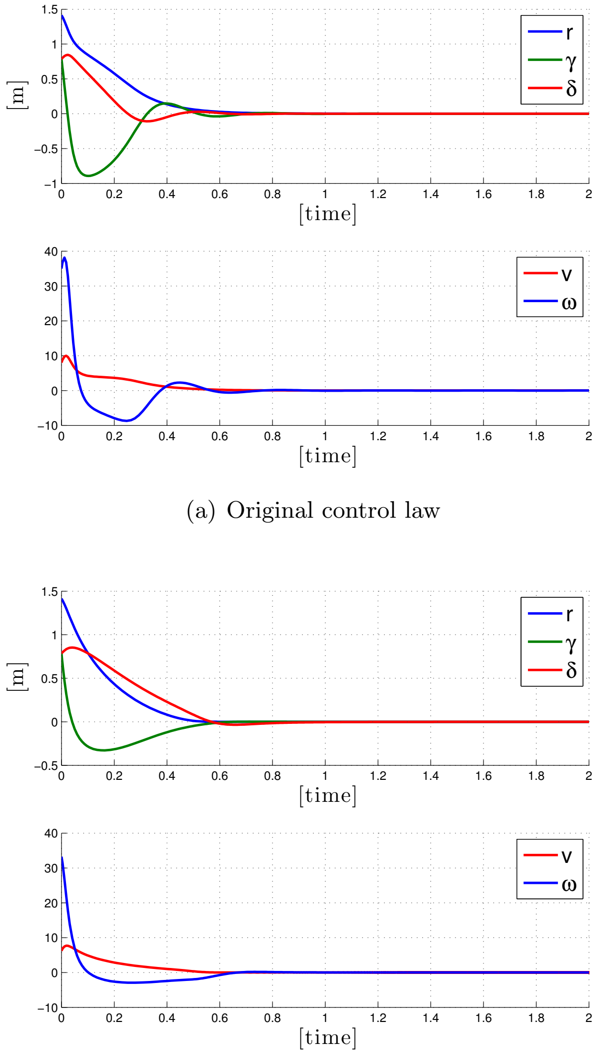
(a) γ by the logarithmic function $-\ln\cos\hat{\gamma}$

(b) γ stays in the bounds designed by the modulating function $\rho_\gamma(t)$

(c) r stays in the bounds designed by the modulating function $\rho_r(t)$

Figure 3.11: Bounds. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3) = (1, 20, 2)$, $\rho_{0_\gamma} = \pi/2$, $\rho_{\infty_\gamma} = 0.01$, $L_\gamma = 4$; $\rho_{0_r} = 2|r_0|$, $\rho_{\infty_r} = 0.01$, $L_r = 3$, $M_r = 0.1$;

In Figure 3.12 it is possible to compare the inputs needed to drive the vehicle to the desired posture, on equal convergence rate. Notice that the initial velocities are greater in the case that the original controller is used. Also, applying the control law designed in order to guarantee prescribed performance on both position and orientation, the obtained inputs and coordinates evolutions are smoother and better distributed over the time.



(a) Original control law

(b) Control law guaranteeing PP bounds on r and γ

Figure 3.12: Inputs comparison, on equal convergence rate.

Chapter 4

Time-varying Control

4.1 Time-varying control

A feasible solution for posture stabilization for nonholonomic WMRs is based on time-varying feedback. Refer to [7].
The posture stabilization problem can be obtained using a fictitious time-varying reference asymptotically vanishing at the origin. Asymptotic stabilization of a state tracking error can be achieved provided that the nominal feedforward commands $v_d(t)$ and $\omega_d(t)$ do not both vanish in finite time. This two desired inputs introduce a time-varying signal in the feedback control law:

$$\begin{aligned} v &= v_d \cos e_3 - u_1 \\ \omega &= \omega_d - u_2, \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} u_1 &= -k_1(v_d(t), \omega_d(t))e_1 \\ u_2 &= -\bar{k}_2 v_d(t)\frac{\sin e_3}{e_3}e_2 - k_3(v_d(t), \omega_d(t))e_3, \end{aligned} \tag{4.2}$$

with constant $\bar{k}_2 > 0$ and positive continuous gain functions $k_1(\cdot,\cdot)$, $k_3(\cdot,\cdot)$, and e defined as

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}.$$

The error dynamics can be expressed as

$$\begin{aligned} \dot{e}_1 &= v_d \cos e_3 - v + e_2\omega \\ \dot{e}_2 &= v_d \sin e_3 + e_1\omega \\ \dot{e}_3 &= \omega_d - \omega \end{aligned} \tag{4.3}$$

and its derivation is reported extensively in Appendix D.1.

In order to achieve posture stabilization, we set $\forall t \; y_d(t) = 0$ and $\theta_d(t) = 0$ (and thus $\omega_d(t) = 0$), while v_d is defined by

$$v_d(t) = \dot{x}_d(t) = -k_4 x_d(t) + g(e,t), \tag{4.4}$$

being $g(e,t)$ the *heating function.* This is a C^2-function uniformly bounded with respect to t, together with its partial derivative. For further details see [7]. The *heating function* $g(e,t)$ plays a key role in guaranteeing asymptotic stability. It sustains motion as long as the error is not zero and also determines the transient behavior. Possible choices for its definition are:

- $g(e,t) = \|e\|^2 \sin t$
- $g(e,t) = \frac{\exp(k_5 e_2)-1}{\exp(k_5 e_2)+1} \sin t, \quad k_5 > 0$, if $k_1(\cdot,\cdot),\; k_3(\cdot,\cdot)$ are strictly positive.

Merging the previous equations, the resulting control law can also be rewritten as

$$\begin{aligned} v &= v_d \cos(\theta_d - \theta) + k_1(v_d, \omega_d)\left[\cos\theta(x_d - x) + \sin\theta(y_d - y)\right] \\ \omega &= \omega_d + \bar{k}_2 v_d \frac{\sin(\theta_d - \theta)}{\theta_d - \theta}\left[\cos\theta(x_d - x) - \sin\theta(y_d - y)\right] + k_3(v_d, \omega_d)(\theta_d - \theta) \end{aligned} \tag{4.5}$$

The proof for the stabilization related to this controller is based on the use of the Lyapunov function

$$V = \frac{\bar{k}_2}{2}\left(e_1^2 + e_2^2\right) + \frac{e_3^2}{2}, \tag{4.6}$$

whose time derivative along the solutions of the closed-loop system is non-increasing since

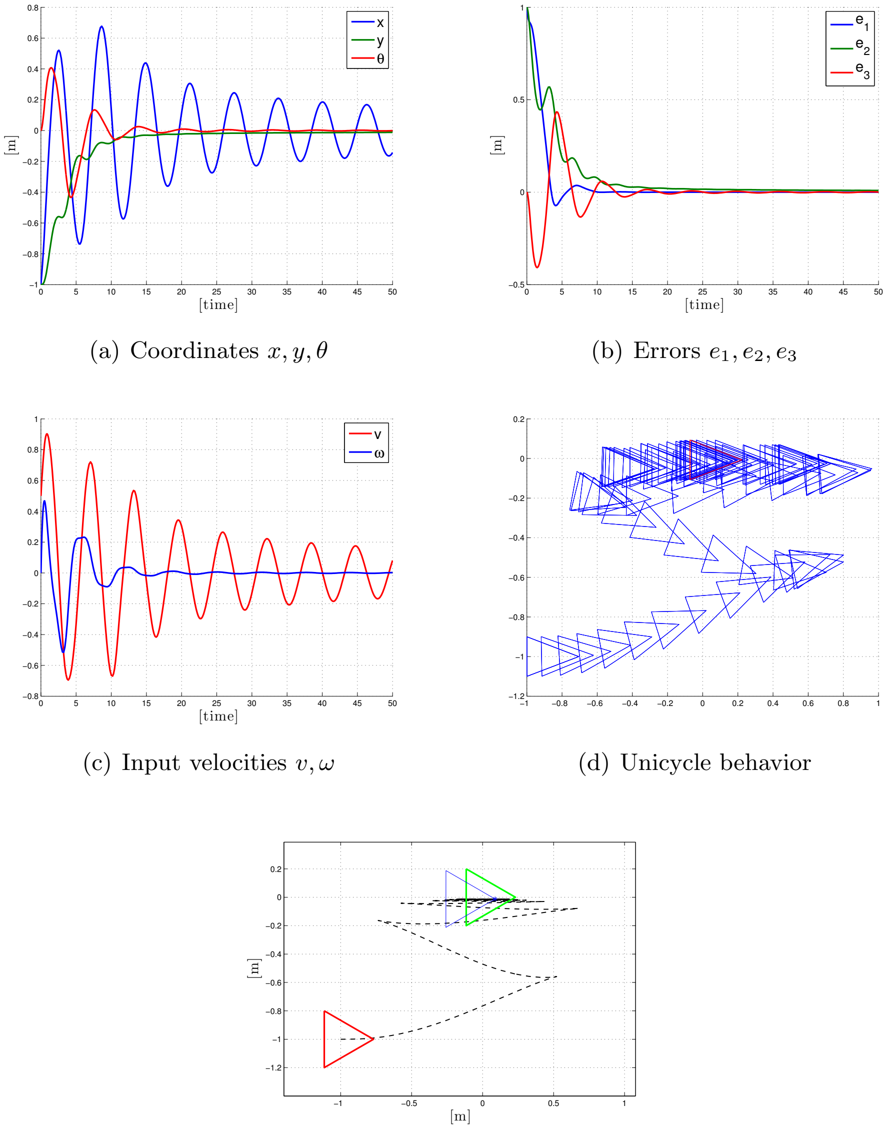
$$\dot{V} = -k_1 \bar{k}_2 e_1^2 - k_3 e_3^2 \leq 0. \tag{4.7}$$

For more details the reader is referred to [7].

We test the time-varying control (4.1), with desired motion given by eq. (4.4), initialized at $x_d(0) = 0$, and heating function

$$g(e,t) = \frac{\exp(k_5 e_2) - 1}{\exp(k_5 e_2) + 1} \sin t.$$

Matlab simulation is depicted in Fig. 4.1. The gains has been set as $k_1 = 0.5,\; \bar{k}_2 = 2,\; k_3 = 1,\; k_4 = 1,\; k_5 = 50$ and the initial conditions as $q(0) = (-1, -1, 0)$ (m,m,rad).



(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

(e) Unicycle trajectory

Figure 4.1: Unicycle behavior with Time-Varying control

4.2 Time-varying control without heating function

The behavior of the unicycle driven by the time-varying control is intrinsically oscillating. This is strictly related to the action of the heating function, which in fact is a modulated sine function.
In order to have different performances, one can change the definition of the desired driving velocity $v_d = \dot{x}_d$, that is to use a different dynamics to describe the desired behavior of x_d.

Define the dynamics of x_d as a damped oscillator:

$$\ddot{x}_d + \bar{k}_d \dot{x}_d + \bar{k}_N^2 x_d = 0 \tag{4.8}$$

where $\bar{k}_d$ represents the damping constant, $\bar{k}_N^2$ the natural frequency, and $\bar{k}_d < 2\bar{k}_N$ (strong damping condition). The second order dynamics can be rewritten as a first order system:

$$\begin{cases} \dot{x}_d = v_{dx} \\ \dot{v}_{dx} = -\bar{k}_d v_{dx} - \bar{k}_N^2 x_d \end{cases} \tag{4.9}$$

Consider the control inputs

$$u_1 = -k_1(v_d(t), \omega_d(t)) e_1 \tag{4.10}$$

$$u_2 = -\bar{k}_2 v_d(t) \frac{\sin e_3}{e_3} e_2 - k_3(v_d(t), \omega_d(t)) e_3 \tag{4.11}$$

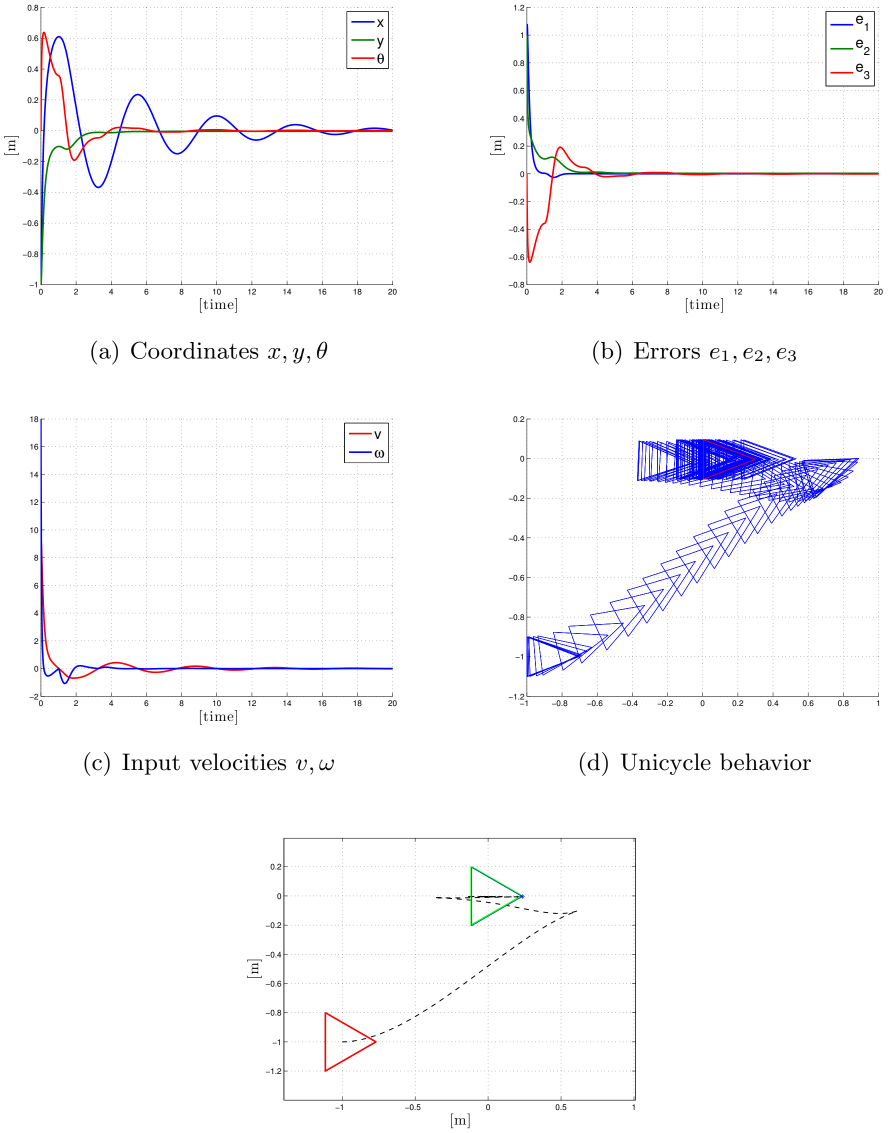
with

$$k_1(v_d(t), \omega_d(t)) = k_3(v_d(t), \omega_d(t)) = 2\zeta \sqrt{\omega_d^2(t) + b v_d^2(t)}$$

$$\bar{k}_2 = b > 0 \quad \zeta \in (0, 1)$$

and set again $y_d(t) = 0, \dot{y}_d(t) = 0$ and so $\omega_d(t) = 0$.

The unicycle behavior under the defined controller is shown in Fig. 4.2. With this controller we can achieve a different behavior and get a shorter transient. The convergence of the error components and of the Cartesian coordinates is faster. We notice however that we have to use higher gains to achieve convergence, and hence the required initial values for the input velocities are higher. The vehicle still needs some settling maneuvers nearby the desired position, due to the oscillating nature of the designed desired linear velocity. However, they are less noticeable with respect to the previous case based on the *heating* function.
The unicycle behavior can be modified or adapted by tuning the parameters $\zeta, b, \bar{k}_2, \bar{k}_d, \bar{k}_N^2$.



(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

(e) Unicycle trajectory

Figure 4.2: Unicycle behavior with Time-Varying control without PP bounds and without *heating function*. Settings: $\zeta = 0.9$, $b = 18$, $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $\bar{k}_d = 0.4$, $\bar{k}_N^2 = 2$

4.3 Control based on different Lyapunov function

In this section, we adopt a different candidate Lyapunov function to design a control law guaranteeing a fair solution for the regulation problem. This approach opens a new way to combine regulation problem and performance bounds guarantees.

Let's consider the error dynamics (4.3) and take a different Lyapunov function, defined as

$$V = \frac{1}{2}\big(e_1^2 + e_2^2\big) + k_3\big(1 - \cos e_3\big) > 0. \tag{4.12}$$

One can observe that this Lyapunov function is similar to the natural candidate Lyapunov function used to describe the pendulum. That is obtained from the total energy $E = E_p + E_k$ (where E_p is the potential energy and E_k the kinematic energy). In the pendulum case the Lyapunov function, and thus the total energy, is given by

$$E = mgl(1 - \cos\phi) + \frac{1}{2}ml^2\phi^2,$$

where m and l are respectively the mass and the length of the pendulum, g the gravity acceleration and ϕ the oscillation amplitude angle. In our case, the Lyapunov function has not a direct physical interpretation. However, we can notice that the cosine function acts again on the angle that describes the system ($e_3 = \theta$).
Defining the control inputs as

$$\begin{aligned} v &= k_1 e_1 + v_d \cos e_3 \\ \omega &= \omega_d + \frac{1}{k_3} v_d e_2 + \sin e_3 \end{aligned} \tag{4.13}$$

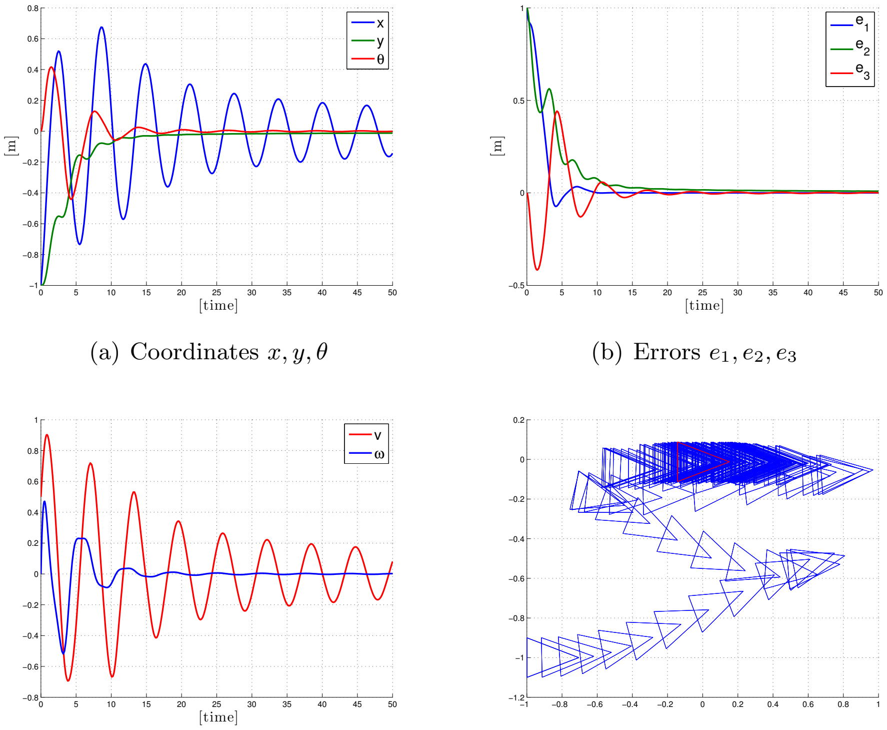
substituting them into the expression of $\dot{V}$ and canceling out some terms we obtain

$$\dot{V} = -k_1 e_1^2 - k_3 \sin^2 e_3 \leq 0. \tag{4.14}$$

Lyapunov analysis allows to prove the convergence of the three error components to zero, and thus the convergence of the Cartesian coordinates to the desired position.

Matlab simulations

The control law (4.13) has been implemented both with and without the heating function. In Figure 4.3 we report the unicycle behavior under the action of the presented controller exploiting the heating function to define the desired velocity. Notice that in this case the behavior is equivalent to the original one. In Figure 4.4 we report the unicycle behavior under the action of the same controller and a desired velocity defined without the heating function, but the dynamics expressed in (4.9).



(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

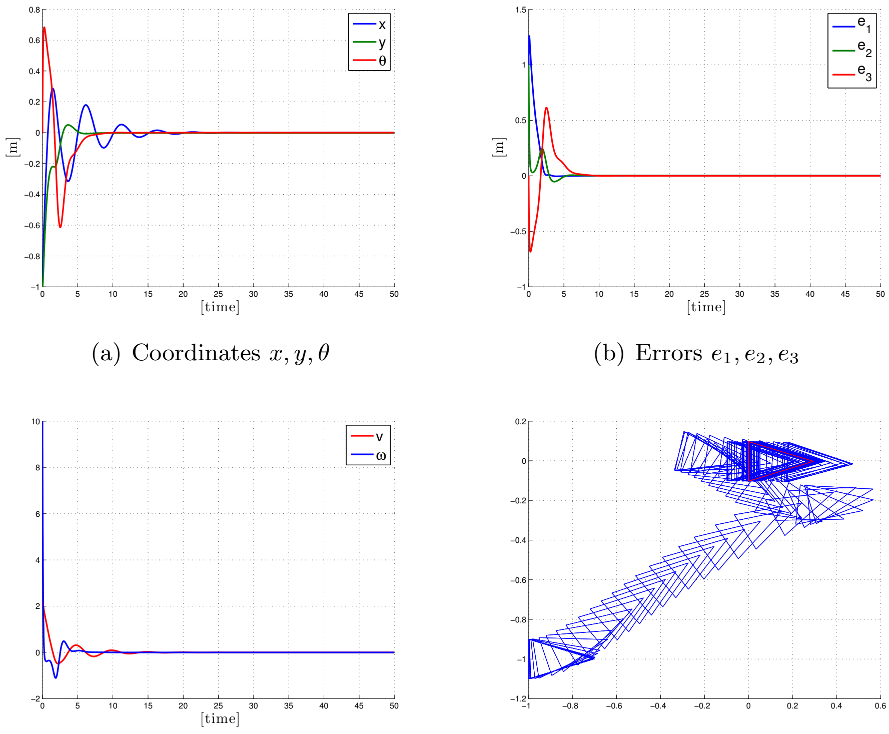
(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.3: Unicycle behavior with Time-Varying control law (4.13), without PP bounds and with *heating function*. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $k_1 = 0.5,\ \bar{k}_2 = 2,\ k_3 = 1,\ k_4 = 1,\ k_5 = 50$

The convergence of the Cartesian coordinates and of the error vector components is guaranteed in both cases. The velocity control inputs are bounded and vanish in short time. The convergence is faster in the case of the controller without heating function: this is due to the pronounced oscillating behavior obtained with the heating function. However, different performance can be achieved modifying or adapting the parameters, namely

$k_1, \bar{k}_2, k_3, k_4, k_5$ in the first case and $k_1, \bar{k}_2, k_3, \bar{k}_d, \bar{k}_N^2$ in the second one.



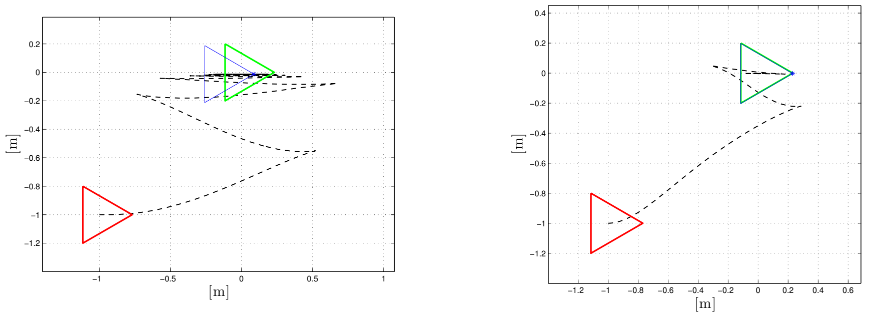
(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.4: Unicycle behavior with Time-Varying control law (4.13), without PP bounds and without *heating function*. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $k_1 = 0.5,\ \bar{k}_2 = 1,\ k_3 = 0.1,\ \bar{k}_d = 0.48,\ \bar{k}_N^2 = 1.6$



(a) Control designed with heating function.

(b) Control designed without heating function.

Figure 4.5: Unicycle trajectories.

4.4 Bounds on orientation

In the previous section we have shown that a candidate Lyapunov function depending on the cosine of the third error component permits to design a fair control law to regulate the unicycle to the desired position. We already revealed also in the Preliminaries section that this kind of Lyapunov function is a fair approach to bind error components as angles.

In this section we put bounds on the unicycle orientation, exploiting another different Lyapunov function, similar to that one used in the previous section, and of the form (2.8).

Consider the error dynamics defined as

$$\begin{aligned}\dot{e}_1 &= \omega_d e_2 + u_1 - e_2 u_2\\ \dot{e}_2 &= \omega_d e_1 + \sin e_3 v_d + e_1 u_2\\ \dot{e}_3 &= u_2\end{aligned} \tag{4.15}$$

with u_1, u_2 the inputs to design.

Take the Lyapunov function

$$V = \frac{k_2}{2}\left(e_1^2 + e_2^2\right) - k_3 \ln\left(\cos e_3\right) \tag{4.16}$$

which is positive definite and well defined in a proper set of values of e_3, namely $e_3 \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. This Lyapunov function operates so that if e_3 starts within $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ then its evolution remains limited by the constraints given by $\ln\left(\cos e_3\right)$.

Define now

$$\begin{aligned}u_1 &= -k_1 e_1\\ u_2 &= -\frac{k_2}{k_3} v_d e_2 \cos e_3 - \tan e_3\end{aligned} \tag{4.17}$$

Substituting the designed controllers in the expression of $\dot{V}$ and canceling out some terms we obtain

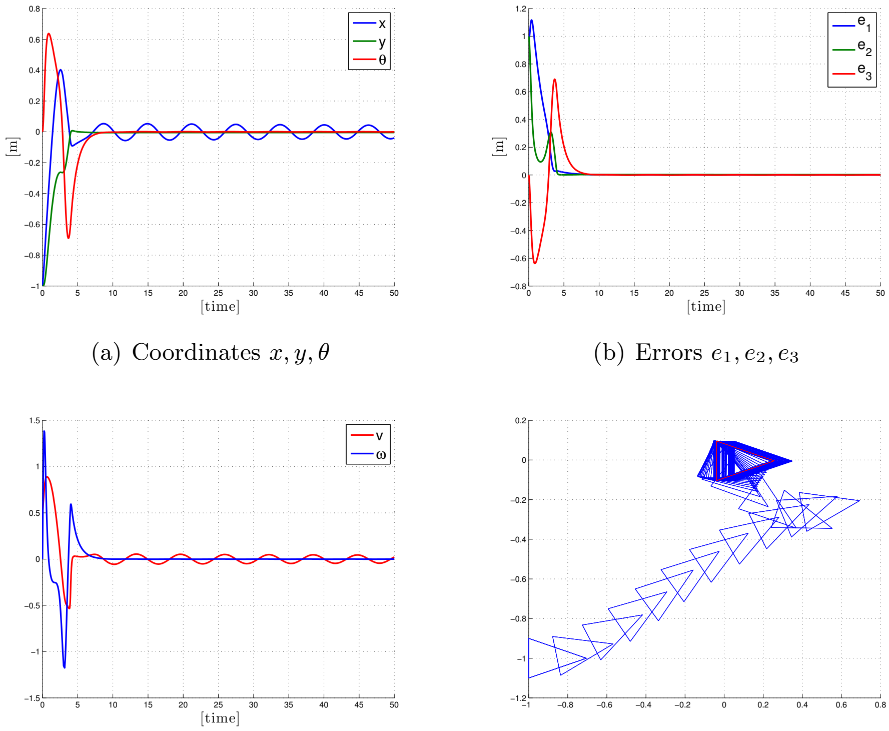
$$\dot{V} = -k_1 k_2 e_1^2 - k_3 \tan^2 e_3 \leq 0. \tag{4.18}$$

Proposition 4.1 *Consider the unicycle description (2.1), the error dynamics (4.3) and the feedback control (4.1) with control inputs defined as (4.17) and k_1, k_2, k_3 positive constants. The closed-loop system (2.1)-(4.1) is then globally asymptotically driven to the posture* $(x, y, \theta) = (0, 0, 0)$. *Also, the error component e_3 respects the prescribed limits.*

The proof for convergence of Proposition 4.1 can be carried on adopting LaSalle theorem and Barbalat lemma, as previously done for other controllers and system of coordinates.

Matlab simulation

Figures 4.6-4.8 show the unicycle behavior under the action of the designed control law. For the simulation represented in Fig. 4.12 and Fig. 4.13, the heating function and the dumped oscillator dynamics are exploited respectively.



(a) Coordinates x, y, θ

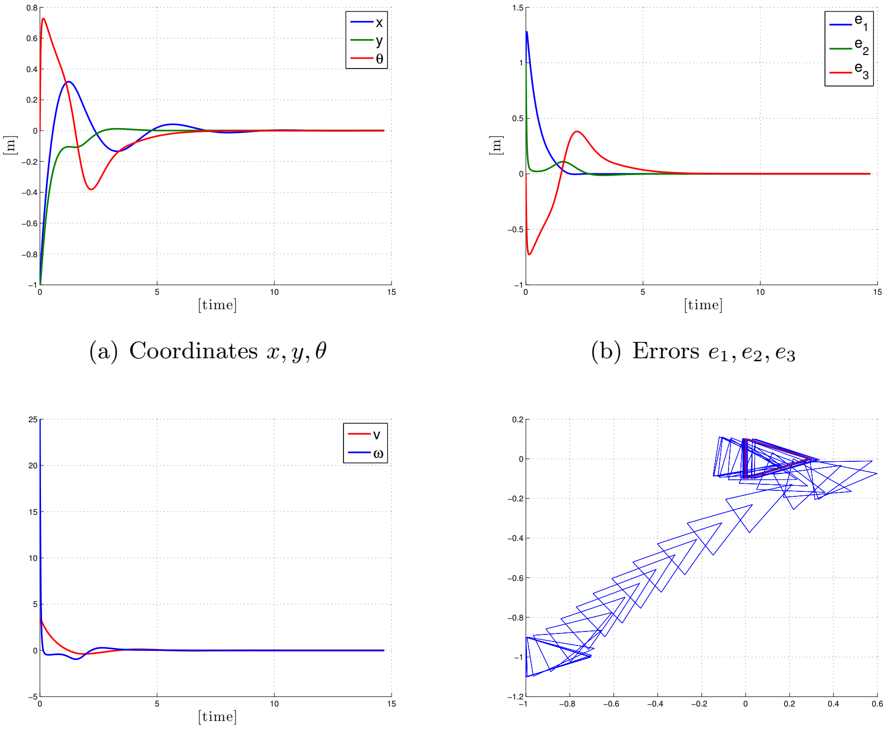
(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.6: Unicycle behavior. Desired linear velocity designed with heating function. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3, k_4, k_5) = (0.5, 1, 0.1, 1, 50)$.

Notice that in the second case a faster convergence is achieved but the initial values of velocity inputs are higher. The convergence in both cases is faster than in the very first presented time-varying controller, and also the pronounced oscillating behavior is less evident. The unicycle presents the best behavior under the time-varying control law designed by mean of the different Lyapunov function and without the heating function: it is not affected by high oscillations, the achieved movement is quite natural and the convergence is pretty fast. However, notice that the achieved performances can be modified or adapted tuning the parameters which regulate the unicycle behavior. Namely, k_1, k_2, k_3, k_4, k_5 in the case with heating function, and $k_1, k_2, k_3, \bar{k}_d, \bar{k}_N^2$ in the case without heating function.



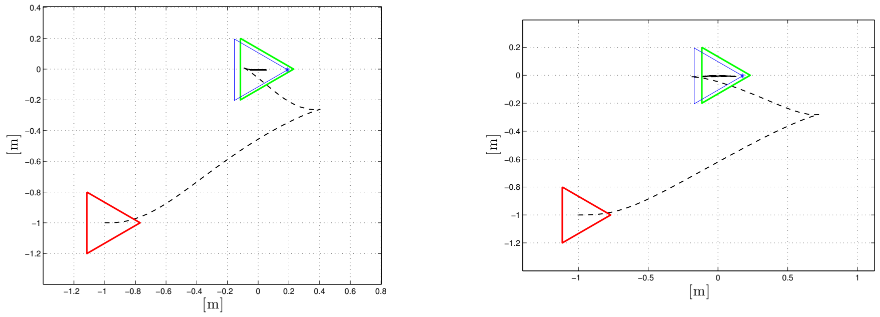
(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.7: Unicycle behavior. Desired linear velocity designed without heating function. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3, \bar{k}_d, \bar{k}_N^2) = (2, 2, 0.08, 1, 2)$.



(a) Control designed with heating function.

(b) Control designed without heating function.

Figure 4.8: Unicycle trajectories.

4.4.1 Time invariant bounds on the orientation

Given that adopting the candidate Lyapunov function (4.16) we are guaranteed that e_3 stays within the range $(-\pi/2, \pi/2)$, the next step consists to reduce this interval by mean of a time invariant error transformation.
We introduce a time-invariant (constant) positive coefficient, namely $\bar{\rho}$, such that

$$e_3 \quad \mapsto \quad \hat{e}_3 = \frac{e_3}{\bar{\rho}}$$

This transformation allows to define more strict bounds on the interval of variation of e_3, and hence of the vehicle orientation. In particular now we have

$$e_3 \in \Big(-\frac{\pi}{2}\bar{\rho}, \frac{\pi}{2}\bar{\rho}\Big).$$

Notice that this interval of values is still constant.
Define the Lyapunov function as

$$V = \frac{k_2}{2}\big(e_1^2 + e_2^2\big) - k_3 \ln\big(\cos \hat{e}_3\big) \tag{4.19}$$

which is positive definite for a specified range of value of e_3, namely $e_3 \in (-\frac{\pi}{2}\bar{\rho}, \frac{\pi}{2}\bar{\rho})$. This means that if we start from a position with $e_3 \in (-\frac{\pi}{2}\bar{\rho}, \frac{\pi}{2}\bar{\rho})$, then this angle coordinate will evolve within the predefined set of value, without ever leaving it.
Define the input controllers

$$\begin{aligned} u_1 &= -k_1 e_1 \\ u_2 &= -\frac{k_2\bar{\rho}}{k_3} v_d e_2 \frac{\cos \hat{e}_3}{\sin \hat{e}_3} \sin e_3 - \tan \hat{e}_3 \end{aligned} \tag{4.20}$$

Differentiating V wrt time and substituting u_1 and u_2 with the expressions in (4.20) we obtain

$$\dot{V} = -k_1 k_2 e_1^2 - \frac{k_3}{\bar{\rho}} \tan^2 \hat{e}_3 \leq 0. \tag{4.21}$$

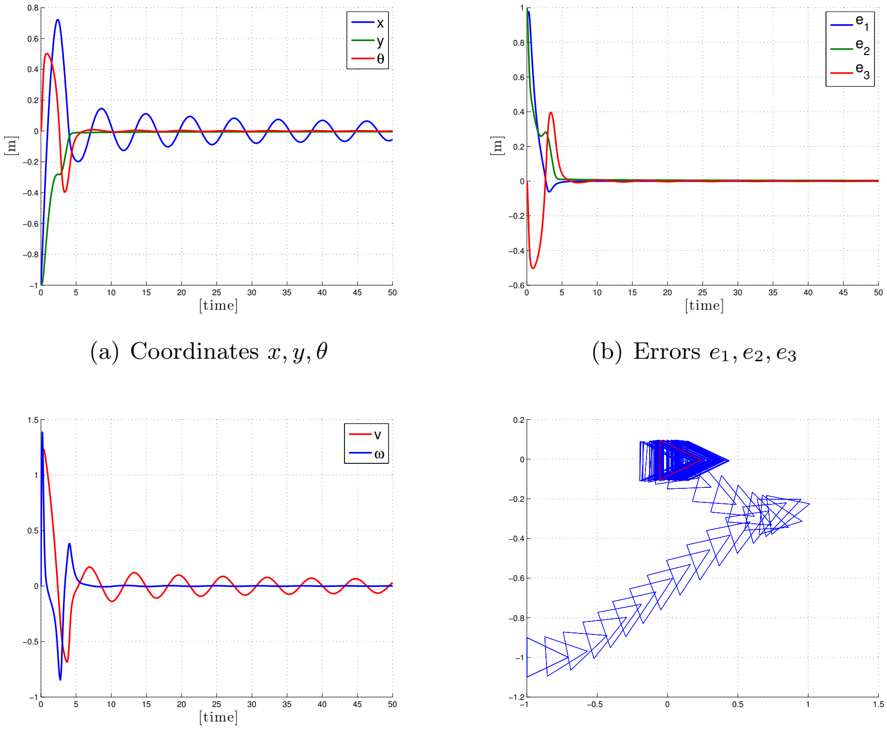
Note that u_2 is bounded and well defined.

Proposition 4.2 *Consider the unicycle description (2.1), the error dynamics (4.3) and the feedback control (4.1) with control inputs defined as (4.20) and k_1, k_2, k_3 positive constants. The closed-loop system (2.1)-(4.1) is then globally asymptotically driven to the posture* $(x, y, \theta) = (0, 0, 0)$. *Also, the error component e_3 respects the prescribed limits.*

As in the previous case, the proof for the convergence of the error components and for the Cartesian coordinates can be carried on exploiting LaSalle theorem and Barbalat Lemma. Details and convergence proof for Proposition 4.2 can be found in Appendix D.2.

Matlab simulation

Figures 4.9-4.11 show the unicycle behavior under the designed control law. For this simulation we set the departure position as $(x_0, y_0, \theta_0) = (-1, -1, 0)$(m,m,rad) and $\bar{\rho} = 0.5$. For the simulation represented in Fig. 4.9 the heating function is exploited to define the desired linear velocity. Fig. 4.10 the dumped oscillator dynamics has been used.
In both cases the convergence of coordinates and error components is guaranteed. Also the control inputs vanish and are bounded.
The initial steering velocity values are higher compared to the unbounded case. The bounded term presents now lower oscillating amplitude, but the convergence is not faster than the one achieved without the constant $\bar{\rho}$ transformation. In the next section we will enhance this performance by mean of a time-varying transformation.



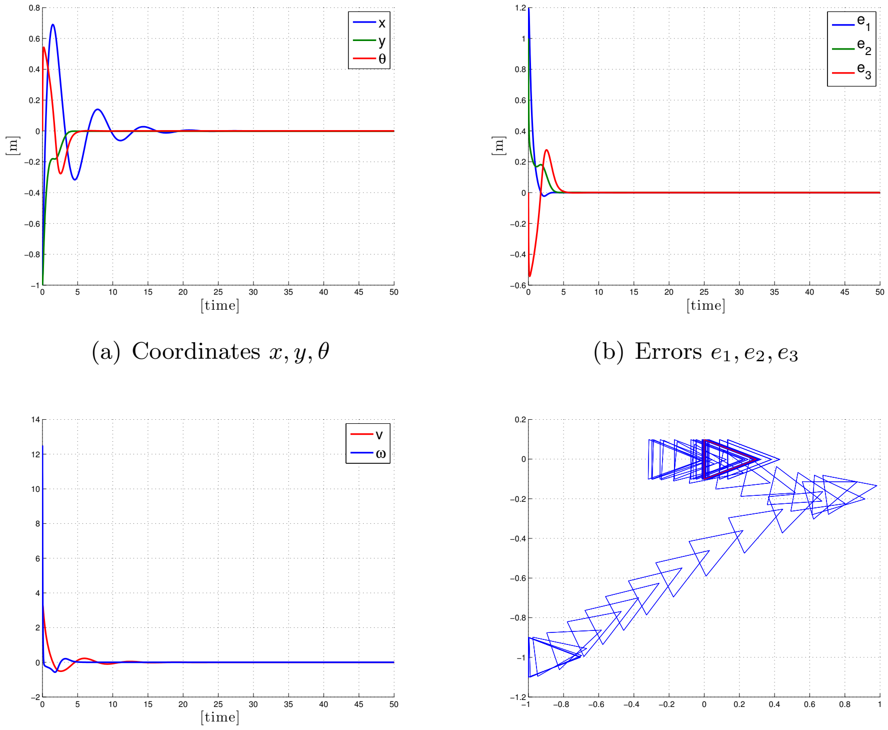
(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.9: Unicycle behavior. Desired linear velocity designed with heating function. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3, k_4, k_5) = (1, 5, 0.1, 1, 50)$.



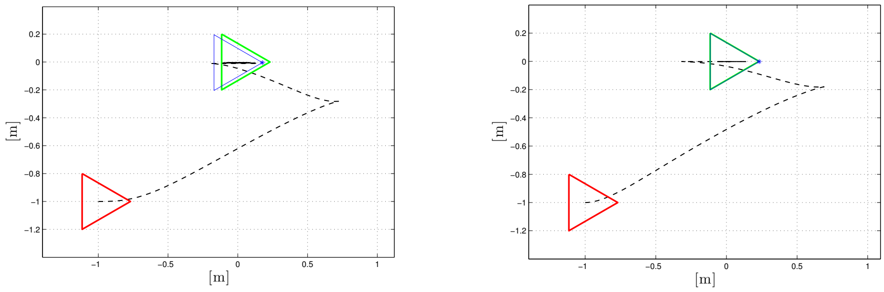
(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.10: Unicycle behavior. Desired linear velocity designed without heating function. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3, \bar{k}_d, \bar{k}_N^2) = (2, 3, 0.7, 0.2\bar{\rho}, 0.4\bar{\rho})$.



(a) Control designed with heating function.

(b) Control designed without heating function.

Figure 4.11: Unicycle trajectories.

4.4.2 Time-varying bounds on the orientation

In order to achieve faster convergence and set more narrow bounds on the orientation, we introduce a time varying transformation through the modulating function $\rho(t)$, such that

$$e_3 \quad \mapsto \quad \hat{e}_3 = \frac{e_3}{\rho(t)}, \quad \rho(t) = (\rho_0 - \rho_\infty)\exp(-Lt) + \rho_\infty.$$

Define the Lyapunov function as

$$V = \frac{k_2}{2}\big(e_1^2 + e_2^2\big) - k_3 \ln\big(\cos \hat{e}_3\big). \tag{4.22}$$

This Lyapunov function is positive definite for a specified range of value of e_3, namely

$$e_3 \in \Big(-\frac{\pi}{2}\rho(t), \frac{\pi}{2}\rho(t)\Big).$$

Notice that the interval of values is now time-varying.
If $e_{30} \in (-\frac{\pi}{2}\rho(0), \frac{\pi}{2}\rho(0))$, then this angle coordinate will evolve within the predefined set of value, without ever leaving it.

Define the input controllers

$$\begin{aligned} u_1 &= -k_1 e_1 \\ u_2 &= -e_3\alpha(t) - \frac{k_2\rho(t)}{k_3} v_d e_2 \frac{\cos \hat{e}_3}{\sin \hat{e}_3} \sin e_3 - \tan \hat{e}_3 \end{aligned} \tag{4.23}$$

Differentiating V wrt time, substituting the the designed controllers in (4.23) and canceling out some terms we obtain

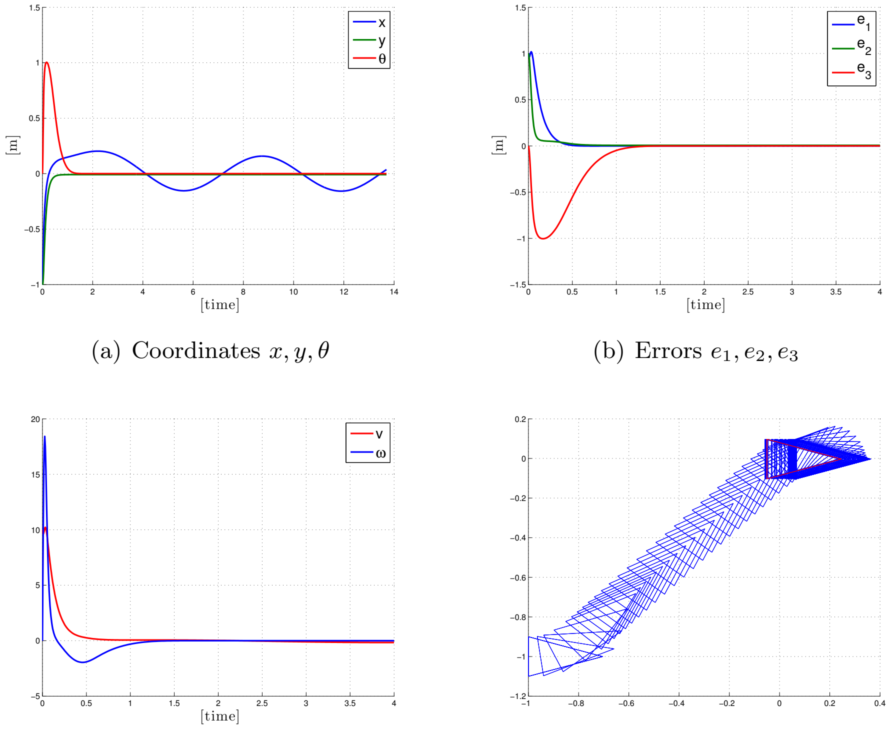
$$\dot{V} = -k_1 k_2 e_1^2 - \frac{k_3}{\rho(t)} \tan^2 \hat{e}_3 \leq 0. \tag{4.24}$$

Proposition 4.3 *Consider the unicycle description (2.1), the error dynamics (4.3) and the feedback control (4.1) with control inputs defined as (4.23) and k_1, k_2, k_3 positive constants. The closed-loop system (2.1)-(4.1) is then globally asymptotically driven to the posture $(x, y, \theta) = (0, 0, 0)$. Also, the error component e_3 respects the prescribed limits.*

As in the previous case, the proof for the convergence of the error components and of the Cartesian coordinates can be carried on exploiting LaSalle theorem and Barbalat Lemma. Details can be found in Appendix D.3.

Matlab simulation

Figures 4.12-4.14 show the unicycle behavior under the designed control law. For this simulation we set the departure position as $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad), $\rho_0 = \frac{\pi}{2}, \rho_\infty = 0.1, L = 3$, and the coefficient are set as $k_1 = 2, k_2 = 8.5, k_3 = 0.2, k_4 = 0.5/\rho_0, k_5 = 50$ and $\bar{k}_d = \rho_0 k_4, \bar{k}_N^2 = 2\rho_0 k_4$.
For the simulation represented in Fig. 4.12 and Fig. 4.13, the heating function and the dumped oscillator dynamics are exploited respectively.



(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

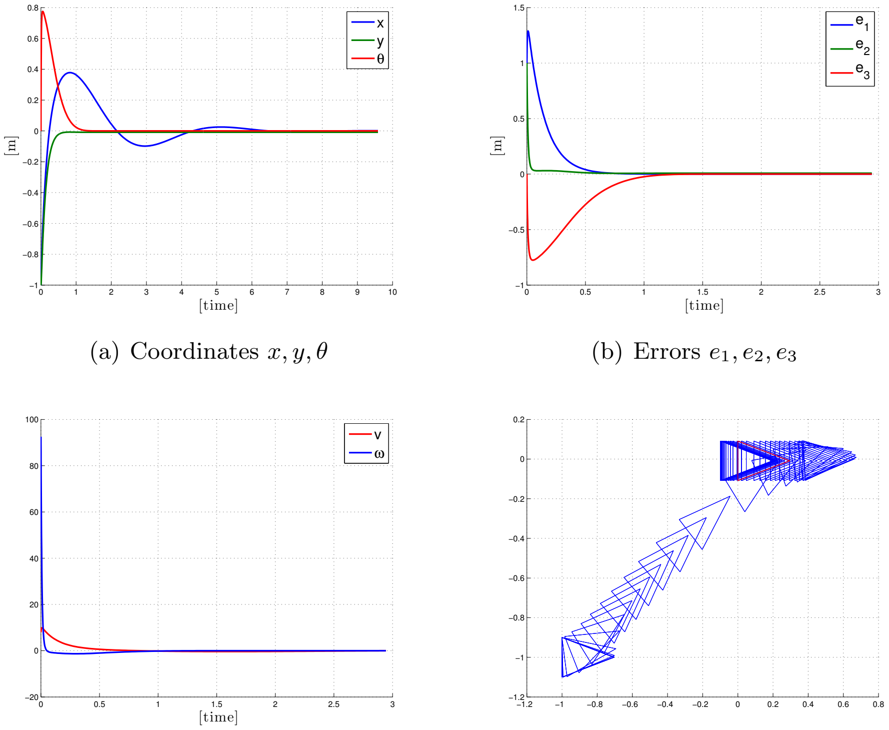
(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.12: Unicycle behavior. Desired linear velocity designed with heating function. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3, k_4, k_5) = (10, 50, 0.1, 0.8, 70)$.

Notice that in this case the convergence is achieved faster than in the previous attempts, even if the oscillating amplitude is slightly higher. The values of the controller inputs are initially high, and then they vanish remaining bounded. The vehicle performs more natural maneuvers for parking, even if it still requires some settling steps.
Finally, we remark that the performances are also affected by the parameters we use to design the control laws. Indeed, modifying or adapting k_1, k_2, k_3, k_4, k_5 in the case exploiting the heating function and $k_1, k_2, k_3, \bar{k}_d, \bar{k}_N^2$

in the other one, it is possible to achieve different performances.



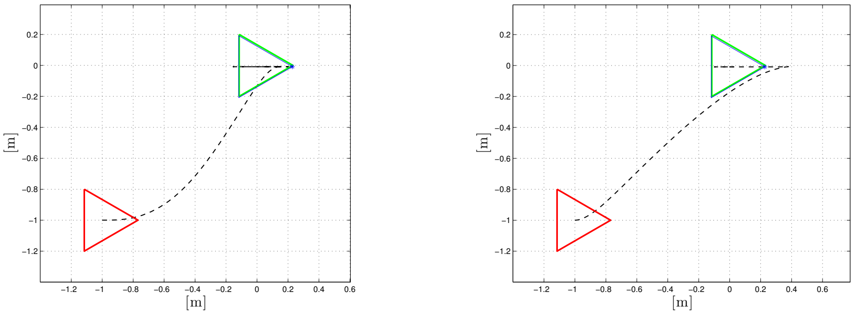
(a) Coordinates x, y, θ

(b) Errors e_1, e_2, e_3

(c) Input velocities v, ω

(d) Unicycle behavior

Figure 4.13: Unicycle behavior. Desired linear velocity designed without heating function. Settings: $(x_0, y_0, \theta_0) = (-1, -1, 0)$ (m,m,rad). $(k_1, k_2, k_3, \bar{k}_d, \bar{k}_N^2) = (7, 15, 0.4, 0.8\rho_0, 1.6\rho_0)$.



(a) Control designed with heating function.

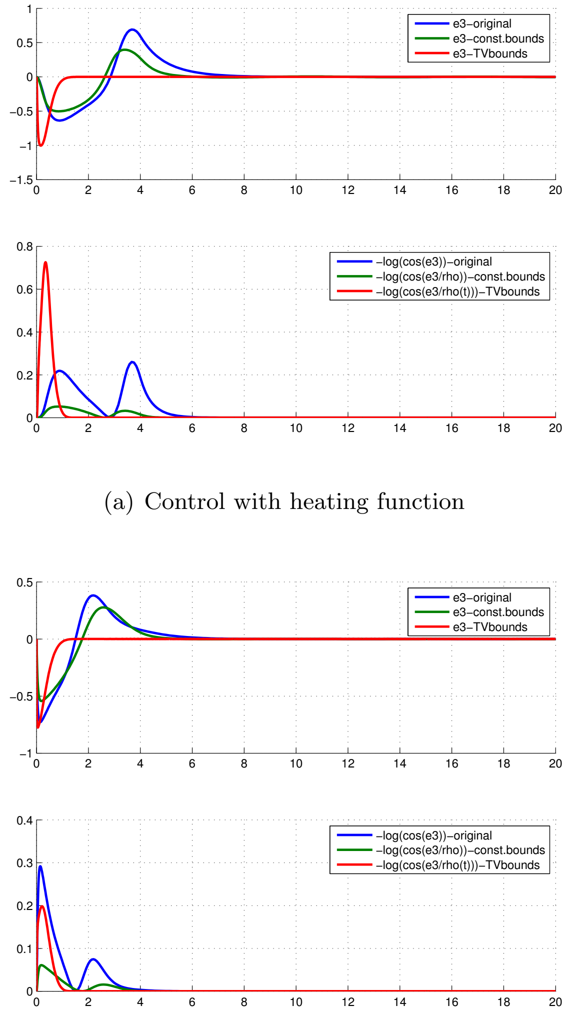
(b) Control designed without heating function.

Figure 4.14: Unicycle trajectories.

4.4.3 Performances of the designed time-varying controllers

In this section we report some figures depicting performances and in particular the transient of the error component e_3 when adopting the three designed time-varying controllers that guarantee prescribed bounds on e_3.

It is not possible to directly make a fair comparison between the aforementioned controllers, since their reliance on parameters (e.g. $k_1, k_2, ...$ etc.) is critical. Hence, we compare the performances of the proposed controllers in the event that they achieve fair convergence. This means that the following figures refer to controllers which have been properly tuned.



(a) Control with heating function

(b) Control without heating function

Figure 4.15: Comparison of the three designed control law.

Figure 4.15 plots the bounded error component e_3 evolution in the time: notice in particular that the convergence achieved by mean of time-varying bounds is much faster than the other cases. In figure 4.15 the bounding term "$-\ln\cos(e_3)$" (or "$-\ln\cos(\hat{e}_3)$") with respect to the time are also reported: we are confirmed that this part of Lyapunov function converges to zero and also has a fast dynamics, so that the bounded coordinate can quickly reach convergence.
Figure 4.15.a refers to the control adopting the heating function, and 4.15.b the other one. One can notice that we achieve better results in the second case, both in term of speed of convergence and in term of oscillating amplitude. We recall that the performances are affected from the choice of the parameters.

A measure of the results obtained by binding e_3 by mean of a time-varying function $\rho(t)$ is reported in Figure 4.16. This picture plots the bounded error component e_3 evolution in the time together with the bounds defined by the modulating function $\rho(t)$, pointing out that this bounds are fully satisfied.

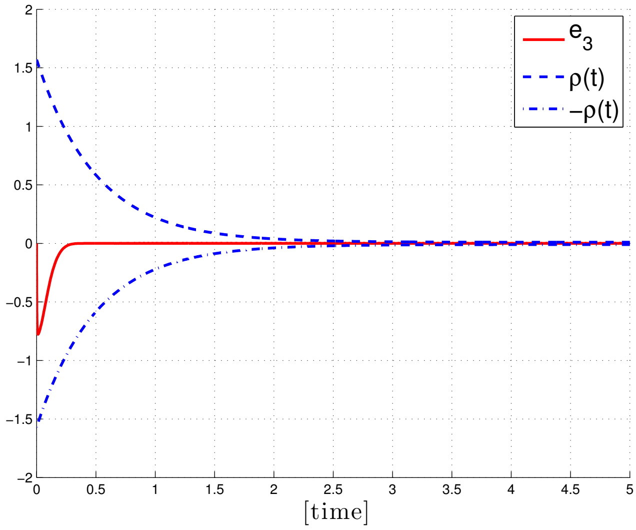


Figure 4.16: Time-varying bounds case: the error e_3 stays within the specified range of value, defined by the $\rho(t)$ function.

Chapter 5

ROS Simulations

This section presents the simulations developed in ROS, *Robot Operating System*. A brief introduction to the ROS framework is here reported. For more details the reader is referred to the wiki available in Internet [15].
ROS is a powerful and current tool to simulate robot behavior.

5.1 Brief introduction to ROS

ROS is an open-source, meta-operating system for robots. It provides the services of an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.
ROS supports code reuse in robotics research and development. It is a distributed framework of processes (called Nodes) that can be grouped into Packages and Stacks, which can be easily shared and distributed. Also it is language neutral and supports various modern programming languages as C++, Python, Lisp.
There exists several releases of ROS: Electric, Fuerte, Groovy, Hydro.
ROS currently only runs on Unix-based platforms. Software for ROS is tested on Ubuntu and Mac OS X systems. A list of robots that can be used with ROS software can be found at `http://www.ros.org/wiki/Robots`.
The peer-to-peer network of ROS processes is the Computation Graph. The basic Computation Graph concepts of ROS are Nodes, Master, Parameter Server, messages, services, topics, and bags. The ROS Master stores topics and services registration information for ROS nodes. Nodes connect to each other directly while the Master only provides lookup information. Nodes communicate with the Master to report their information and to receive information about other registered nodes and make connections. The Master will also make callbacks to these nodes when informations change, allowing

dynamic connections between nodes.
Topics are named buses over which nodes exchange messages, and they are intended for unidirectional, streaming communication. Topics have anonymous publish/subscribe semantics, which decouples the production of information from its consumption. There can be multiple publishers and subscribers to a topic. A schematic representation is depicted in Figure 5.1.

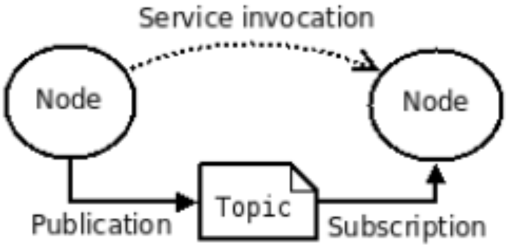


Figure 5.1: Nodes connection: communication structure. Figure source: `//http://www.ros.org/wiki/ROS/Concepts`

TurtleSim

TurtleSim is a ROS simulator for teaching. It is a simple tool, but effective for the purpose of testing and simulating the designed controllers for the unicycle-like robot.
ROS structures are used to implement the controllers. Turtlesim node permits to visualize the moving vehicle, which in this case is a turtle. With a view to future implementations, it is possible to reuse the code implementing the controller (control Node) and apply it on different models of unicycle-like robot: as an example Pioneer.

5.2 Implementation

We perform simulations for the controllers designed by mean of polar coordinates. We use ROS Fuerte release and C++ language.
A package `turtle_unicycle` has been created. It contains a single Node called `/controller_node`. There are two classes: `UnicycleVelocity` and `UnicycleVelocityControllerNode`. The first one is responsible for calculating the unicycle velocity while implementing the designed control laws and the error transformations for prescribed performance constraints. The second one implements all the necessary ROS components. For code documentation see Appendix E.
The `main()` function creates and initializes the Control Node. During the

simulations, the Control Node receives as input the unicycle position, that is the Cartesian coordinates x, y, θ, and calculates the unicycle velocities exploiting UnicycleVelocity class methods. The velocities are then used to visualize the turtle movements. Figure 5.2 depicts what explained. In particular note the three topics related to `/turtlesim_node`: `/command_velocity`, `/pose`, `/color_sensor`. The position of the turtle is used to calculate the velocities, which are given in input to the turtle node to generate its movement.

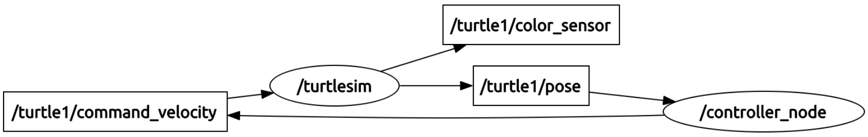
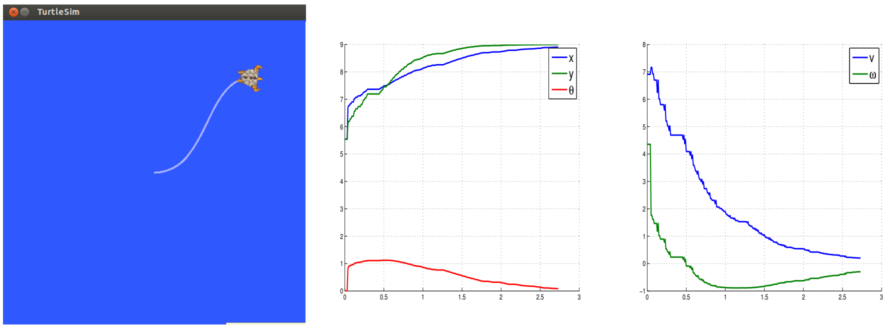


Figure 5.2: Graph obtained from `ROS rqt_graph`: it shows topics flowing in and out of the `/controller_node` and the `/turtlesim_node`.

Simulation 1.

In the first simulation, we implement the standard control law with polar coordinates, namely (3.3). The initial position of the turtle is $(x, y, \theta) = (5.544445, 5.544445, 0)$ while the goal position is set to $(x, y, \theta) = (9, 9, 0)$. Figure 5.3 shows the turtle behavior and the evolution in the time of the three coordinates x, y, θ.



(a) Turtle movement

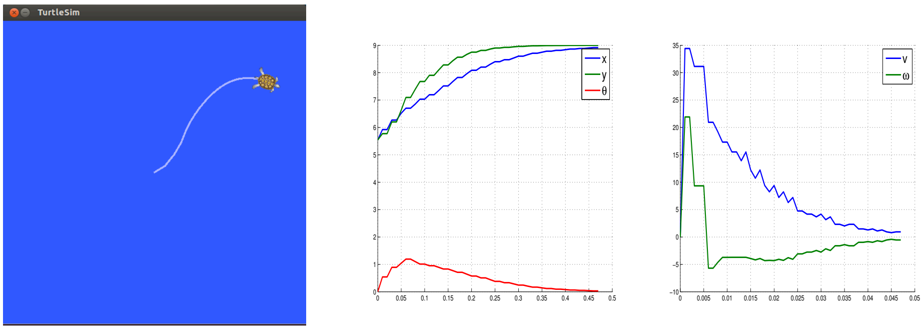
(b) Coordinates x, y, θ

(c) Inputs

Figure 5.3: Turtle behavior under the original control law (3.3) and time evolution of the turtle coordinates.

Simulation 2.

In this simulation, we implement the control law (3.5), that is the controller that guarantees prescribed performance on the radial coordinate. The turtle departs from the position $(x, y, \theta) = (5.544445, 5.544445, 0)$. The goal position is set to $(x, y, \theta) = (9, 9, 0)$. The movement performed by the turtle in its window fit with that one obtained with Matlab simulations, and it is reported in Figure 5.4.a. Figure 5.4.b shows the evolution in the time of the three coordinates x, y, θ.

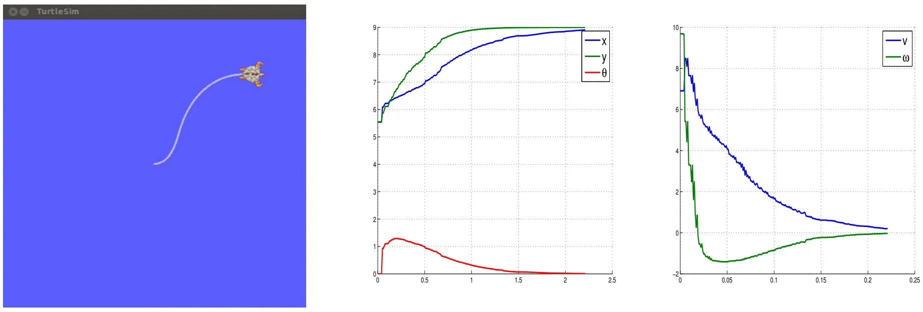


(a) Turtle movement (b) Coordinates x, y, θ (c) Inputs

Figure 5.4: Turtle behavior under the control law (3.5) and time evolution of the turtle coordinates.

Simulation 3.

Figure 5.5 shows the turtle behavior under the action of the control law (3.13), that is the controller that guarantees prescribed bounds on the orientation of the unicycle. The initial and goal positions are the same as in the previous simulations.

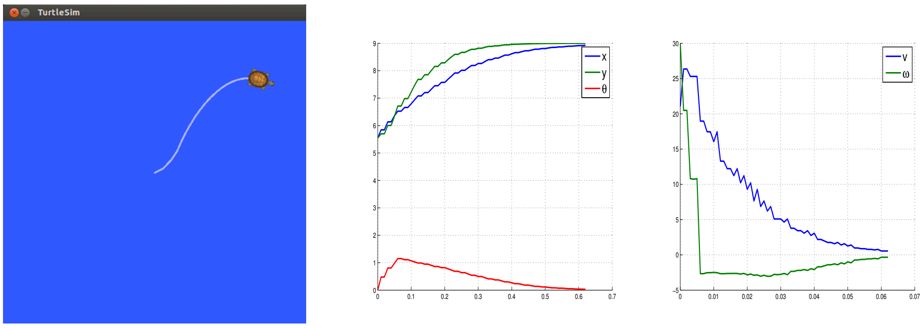


(a) Turtle movement (b) Coordinates x, y, θ (c) Inputs

Figure 5.5: Turtle behavior under the control law (3.13) and time evolution of the turtle coordinates.

Simulation 4.

Eventually, this simulation implements the controller that permits to bind both radial and angle coordinate. Figure 5.6 shows the turtle behavior under the action of the control law (3.16). The initial and goal positions are the same as in the previous simulations. Note that in this case the convergence to the desired posture is much faster than in the previous simulations.



(a) Turtle movement

(b) Coordinates x, y, θ

(c) Inputs

Figure 5.6: Turtle behavior under the control law (3.16) and time evolution of the turtle coordinates.

Chapter 6

Conclusion

This thesis addresses the regulation problem for mobile robots of the type of the unicycle. Different controllers have been designed, in order to guarantee prescribed performances. The main results and proofs of convergence have been obtained by mean of the Lyapunov analysis.
WMRs are nonholonomic system, thus the regulation problem cannot be solved via a smooth, time-invariant state feedback law due to the implications of Brockett's condition.

The unicycle model has been introduced and two main approaches to its regulation have been examined: control with polar coordinates and time-varying control.
The prescribed performance control concept has also been analyzed and employed to design controllers that solve the regulation problem while guaranteeing prescribed bounds.
A first example of solution combining the regulation aim and the prescribed performance control concepts has been given by the Dynamic Feedback Linearization.

The best performing presented control law has been designed by mean of polar coordinates. Prescribed performances have been imposed in order to bind both the position and the orientation of the vehicle. The controller based on polar coordinates transformation performs very well. The resulting vehicle path is very natural and convergence is quite fast. This is one of the main result of this study. The convergence to the desired position is achieved with natural maneuvers, while the polar coordinates r and γ are guaranteed to respect the predefined limits. This result also points out a relevant difference between the approach followed to bind the distance r and the one used to bind the orientation (through the angle γ). While in the first case we begin with the error transformation and exploit the transformed error in the potential to define the Lyapunov function, in the latter we bind the coordinate starting from the definition of a different potential depending from the cosine of the coordinate to bind.

Then, time-varying control laws has been introduced. Prescribed performances have been imposed to bind the vehicle orientation. The desired driving velocity is defined by mean of oscillating functions: in the original case a heating function designed as a modulated sine function is exploited; a different solution is give defining the driving velocity with a dumped oscillator dynamics. Controllers have been tested both with and without heating function. Different types of Lyapunov functions have been used. A control law as been designed exploiting a Lyapunov function recalling the pendulum candidate Lyapunov function. Also, a Lyapunov function including a potential term depending from the cosine of the component to bind e_3 has been employed to impose prescribed performance on the vehicle orientation. Time-varying controllers, both with and without heating function, exhibit a rather slow final convergence to the goal. The dependence of the convergence rate on the available gains is critical. The oscillatory behavior of the vehicle is an intrinsic characteristic.

The last Section presented simulations implemented in *ROS* environment. The controllers designed by mean of polar coordinates have been tested, exploiting the Turtlesim simulator.

Future work

Future directions of work should consider also different approaches to the regulation problem for the unicycle. For example discontinuous or non-smooth time-varying controllers can be analyzed in order to introduce prescribed performance.
The prescribed performance control concept can be applied for the control problem of other type of nonholonomic systems, such as the car-like mobile robot or more complex structures.
Prescribed performance can be imposed also in the trajectory tracking problem, and in the robust control approach.
Time-varying controllers can be tested in ROS environment and simulations can be also implemented employing real robot models, as for example Pioneer.
Also, the proposed control laws could be implemented on a real mobile unicycle-like robots.

Bibliography

[1] Roger W Brockett et al. *Asymptotic stability and feedback stabilization.* Defense Technical Information Center, 1983.

[2] Alessandro Astolfi. Discontinuous control of nonholonomic systems. *Systems & Control Letters*, 27(1):37–45, 1996.

[3] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli. Wmr control via dynamic feedback linearization: design, implementation, and experimental validation. *Control Systems Technology, IEEE Transactions on*, 10(6):835–852, 2002.

[4] Herbert G Tanner and Kostas J Kyriakopoulos. Discontinuous backstepping for stabilization of nonholonomic mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3948–3953. IEEE, 2002.

[5] Herbert G Tanner, Savvas Loizou, and Kostas J Kyriakopoulos. Nonholonomic stabilization with collision avoidance for mobile robots. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1220–1225. IEEE, 2001.

[6] Claude Samson. Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *Automatic Control, IEEE Transactions on*, 40(1):64–77, 1995.

[7] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators.* Springer Verlag, 2000.

[8] Ilya Kolmanovsky and N Harris McClamroch. Developments in nonholonomic control problems. *Control Systems, IEEE*, 15(6):20–36, 1995.

[9] Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. In *Ramsete*, pages 181–226. Springer, 2001.

[10] Y. Karayiannidis and Z. Doulgeri. Model-free robot joint position regulation and tracking with prescribed performance guarantees. *Robotics and Autonomous Systems*, 60(2):214–226, 2012.

[11] Zoe Doulgeri and Yiannis Karayiannidis. Pid type robot joint position regulation with prescribed performance guaranties. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4137–4142. IEEE, 2010.

[12] Ch Bechlioulis, Zoe Doulgeri, and G Rovithakis. Prescribed performance adaptive control for robot force/position tracking. In *Control Applications,(CCA) & Intelligent Control,(ISIC), 2009 IEEE*, pages 920–925. IEEE, 2009.

[13] Zoe Doulgeri, Yiannis Karayiannidis, and Olga Zoidi. Prescribed performance control for robot joint trajectory tracking under parametric and model uncertainties. In *Control and Automation, 2009. MED'09. 17th Mediterranean Conference on*, pages 1313–1318. IEEE, 2009.

[14] Charalampos P Bechlioulis and George A Rovithakis. Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance. *Automatic Control, IEEE Transactions on*, 53(9):2090–2099, 2008.

[15] URL `//http://www.ros.org/wiki/`.

[16] Daniel Liberzon. *Switching in Systems and Control.* Springer Birkhauser, Boston, 2003.

[17] Herbert G Tanner and Kostas J Kyriakopoulos. Nonholonomic motion planning for mobile manipulators. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1233–1238. IEEE, 2000.

[18] Anthony M Bloch, Mahmut Reyhanoglu, and N Harris McClamroch. Control and stabilization of nonholonomic dynamic systems. *Automatic Control, IEEE Transactions on*, 37(11):1746–1757, 1992.

[19] Alessandro De Luca and Giuseppe Oriolo. Modeling and control of nonholonomic mechanical systems. *Kinematics and Dynamics of Multi-Body Systems, CISM Courses and Lectures*, 360:277–342, 1995.

[20] Stern R.J. Brockett's stabilization condition under state constraints. *Systems and Control Letters*, 47(4):335–341, 2002. URL `http://www.ingentaconnect.com/content/els/01676911/2002/00000047/00000004/art00227`.

[21] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control.* Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 1846286417, 9781846286414.

[22] G Campion, B d'Andrea Novel, and G Bastin. Controllability and state feedback stabilizability of non holonomic mechanical systems. In *Advanced robot control*, pages 106–124. Springer, 1991.

[23] C Canudas De Wit and OJ Sordalen. Exponential stabilization of mobile robots with nonholonomic constraints. *Automatic Control, IEEE Transactions on*, 37(11):1791–1797, 1992.

[24] C Canudas De Wit, H Khennouf, C Samson, and OJ Sordalen. Nonlinear control design for mobile robots. *Recent trends in mobile robots*, 11:121–156, 1993.

[25] Jean-Michel Coron. Global asymptotic stabilization for controllable systems without drift. *Mathematics of Control, Signals and Systems*, 5 (3):295–312, 1992.

[26] Yiannis Karayiannidis, Dimos V Dimarogonas, and Danica Kragic. Multi-agent average consensus control with prescribed performance guarantees. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 2219–2225. IEEE, 2012.

Appendix A

Two approaches to impose PP bounds

Polar coordinate control case

Being $e = \gamma$,

$$\dot{e} = \dot{\gamma} = \frac{\sin\gamma}{r}v - \omega;$$

thus, in the first derivative of the Lyapunov function there are terms of the form

$$\varepsilon\big(\frac{\sin\gamma}{r}v - \omega\big) \qquad \text{and} \qquad \tan\hat{e}\big(\frac{\sin\gamma}{r}v - \omega\big).$$

Given the polar description of the unicycle and its dynamics, the driving velocity is designed as

$$v = k_1 r\cos\gamma.$$

This input allows to have a square term of r multiplied for a $\cos^2\gamma$ in the first derivative of the Lyapunov function. This is needed for the convergence proof.
Given this driving velocity and the δ dynamics, in the first derivative of the Lyapunov function appears a term of the form

$$\delta\frac{\sin\gamma}{r}v = \delta\frac{\sin\gamma}{r}[k_1 r\cos\gamma] = k_1\delta\sin\gamma\cos\gamma \tag{A.1}$$

which is the derivative of the quadratic term $\frac{1}{2}\delta^2$ in the Lyapunov function. The steering velocity is designed in order to cancel out the term in (A.1), thus ω will depend on a term of the form

$$\tilde{\omega}_1 = \frac{1}{\varepsilon}k_1\delta\sin\gamma\cos\gamma \qquad \text{and} \qquad \tilde{\omega}_2 = \frac{1}{\tan\hat{e}}k_1\delta\sin\gamma\cos\gamma \tag{A.2}$$

in the first and second approach respectively.
Observe that in both the expression in (A.2) the terms δ/ε and $\delta/\tan\hat{\gamma}$ are

unbounded, since both ε and $\hat{e}$ vanish faster than δ, due to the prescribed performance bounds.
The previous equations can be rewritten as

$$\tilde{\omega}_1 = k_1\delta\frac{\frac{\sin\gamma\cos\gamma}{\gamma}}{\frac{\varepsilon}{\gamma}} \qquad \text{and} \qquad \tilde{\omega}_2 = k_1\delta\frac{\frac{\sin\gamma\cos\gamma}{\gamma}}{\frac{\tan\hat{\gamma}}{\gamma}} \tag{A.3}$$

which are well defined terms.
The convergence proof for the unicycle system is based on Barbalat lemma; in particular we are interested to prove that $\dot{\gamma}$ is uniformly continuous. To do that we check the boundedness of the second derivative $\ddot{\gamma}$.
In the first case, the lack of convergence is related to the fact that $\dot{\tilde{\omega}}_1$ is not bounded and thus $\ddot{\gamma}$ is not uniformly continuous, while in the second case $\dot{\tilde{\omega}}_2$ is proved to be bounded and it is possible to conclude the convergence proof. In details, $\dot{\tilde{\omega}}_1$ is given by

$$\dot{\tilde{\omega}}_1 = k_1\Big[\dot{\delta}\frac{s_\gamma c_\gamma/\gamma}{\varepsilon/\gamma} + \delta\Big(\frac{d}{dt}\Big(\frac{s_\gamma c_\gamma}{\gamma}\Big)\Big)\frac{\gamma}{\varepsilon} + \delta\frac{s_\gamma c_\gamma}{\gamma}\frac{d}{dt}\Big(\frac{\gamma}{\varepsilon}\Big)\Big] \tag{A.4}$$

where s_γ, c_γ indicate $\sin\gamma, \cos\gamma$. The only unbounded term in (A.4) is

$$\frac{d}{dt}\frac{\gamma}{\varepsilon} = \Big(\frac{\sin\gamma}{r}v - \omega\Big)\underbrace{\Big(\frac{1}{\varepsilon} - \frac{J\gamma}{\varepsilon^2}\Big)}_{unbounded} - \alpha J\Big(\frac{\gamma}{\varepsilon}\Big)^2. \tag{A.5}$$

In the second case, $\dot{\tilde{\omega}}_2$ is given by

$$\dot{\tilde{\omega}}_1 = k_1\Big[\dot{\delta}\frac{s_\gamma c_\gamma/\gamma}{\tan\hat{\gamma}/\gamma} + \delta\Big(\frac{d}{dt}\Big(\frac{s_\gamma c_\gamma}{\gamma}\Big)\Big)\frac{\gamma}{\tan\hat{\gamma}} + \delta\frac{s_\gamma c_\gamma}{\gamma}\frac{d}{dt}\Big(\frac{\gamma}{\tan\hat{\gamma}}\Big)\Big] \tag{A.6}$$

All the terms are well defined and bounded. In particular now

$$\begin{aligned} \frac{d}{dt}\frac{\gamma}{\tan\hat{\gamma}} &= \Big(\frac{\sin\gamma}{r}v - \omega\Big)\Big(\frac{1}{\tan\hat{\gamma}} - \frac{\gamma}{\rho}\frac{1+\tan^2\hat{\gamma}}{\tan^2\hat{\gamma}}\Big) - \frac{\alpha}{\rho}\gamma^2 \\ &= \Big(\frac{\sin\gamma}{r}v - \omega\Big)\Big(-\hat{\gamma} + \frac{\tan\hat{\gamma} - \hat{\gamma}}{\tan^2\hat{\gamma}}\Big) - \frac{\alpha}{\rho}\gamma^2 \end{aligned} \tag{A.7}$$

where the term $\frac{\tan\hat{\gamma}-\hat{\gamma}}{\tan^2\hat{\gamma}}$ is bounded as long as $\hat{\gamma}$ is not null, and as $\hat{\gamma} \to 0$ the limit gives

$$\lim_{\hat{\gamma}\to 0}\frac{\tan\hat{\gamma} - \hat{\gamma}}{\tan^2\hat{\gamma}} = 0. \tag{A.8}$$

Being $\dot{\tilde{\omega}}_2$ bounded, then also the derivative of the steering velocity is bounded and this fact allows to conclude the convergence proof.

We remark that this issue is not arising in the case we bind the radial coordinate r. This is still related to the unicycle dynamics definition with polar coordinates. The original Lyapunov function can be modified just substituting e with $\varepsilon = T(r/\rho)$, that is adopting a potential of the form (2.11). In this case the driving velocity v acts directly on ε, so that we have a square term ε^2 needed for convergence proof. Then the steering velocity can easily compensate the term deriving from $\frac{1}{2}\delta^2$ and no unbounded terms arise: the term ε/e appears, but this term is proved to be bounded.

Time-varying control case

In the case of time-varying control, the reasoning is analogous: a potential defined as in (2.12) allows to conclude the convergence proof according to Barbalat lemma.

In the case of time-varying control we define the error through a rotation of the Cartesian error:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}.$$

The orientation of the unicycle is directly linked with the third error component e_3, which is now the coordinate to bind. After some simplifications, the error dynamics can be written as

$$\begin{aligned} \dot{e}_1 &= u_1 - e_2 u_2 \\ \dot{e}_2 &= \sin e_3 v_d + e_1 u_2 \\ \dot{e}_3 &= u_2 \end{aligned} \tag{A.9}$$

The input control u_1 is designed in order to obtain a square term e_1^2, needed for the convergence proof. This input is defined as $u_1 = -k_1 e_1$ and does not compensate the sinusoidal term related to e_2.

Considering the dynamics of e_2, in the first derivative of the Lyapunov function appears a term of the form

$$e_2 \sin e_3 v_d \tag{A.10}$$

The second input velocity u_2 is designed in order to cancel out the term in (A.10), thus u_2 will depend on a term of the form

$$\tilde{u}_2 = \frac{1}{\varepsilon} e_2 \sin e_3 v_d \qquad \text{and} \qquad \tilde{u}_2 = \frac{1}{\tan\hat{e}} e_2 \sin e_3 v_d \tag{A.11}$$

in the first and second approach respectively.

Observe that in both the expression in (A.11) the terms e_2/ε and $e_2/\tan\hat{e}$

are unbounded, since both ε and $\hat{e}$ vanish faster than e_2, due to the prescribed performance bounds.

The first approach, based on defining the transformed error ε and employing a potential of the form (2.11), yields also in this case to unbounded terms that cause the lack of convergence. On the other hand, the second approach is successful, since we can prove that the first derivative of $\tilde{u}_2$ is bounded and $\ddot{e}_3$ as well. This allows to conclude the convergence proof.

Appendix B

DFL: details and convergence proof.

The first derivative of the Lyapunov function (2.20) is calculated as follow:

$$\begin{aligned}
\dot{V} &= \big(\dot{e} + \alpha e\big)^T \big(\ddot{e} + \dot{\alpha} e + \alpha \dot{e}\big) + \varepsilon^T K_\varepsilon \dot{\varepsilon} \\
&= \big(\dot{e} + \alpha e\big)^T \big(u + \dot{\alpha} e + \alpha \dot{e}\big) + \varepsilon^T K_\varepsilon J_T \big(\dot{e} + \alpha e\big) \\
&= \big(\dot{e} + \alpha e\big)^T \big(-K_v \big(\dot{e} + \alpha e\big) - K_\varepsilon J_T \varepsilon - \cancel{\dot{\alpha} e} - \cancel{\alpha \dot{e}} + \cancel{\dot{\alpha} e} + \cancel{\alpha \dot{e}}\big) + \varepsilon^T K_\varepsilon J_T \big(\dot{e} + \alpha e\big) \\
&= -\dot{e}^T K_v \dot{e} - \cancel{\dot{e}^T K_v \alpha e} - \cancel{\dot{e}^T K_\varepsilon J_T \varepsilon} - \cancel{e^T \alpha^T K_v \dot{e}} - e^T \alpha^T K_v \alpha e + \\
&\quad - \cancel{e^T \alpha^T K_\varepsilon J_T \varepsilon} + \cancel{\varepsilon^T K_\varepsilon J_T \dot{e}} + \cancel{\varepsilon^T K_\varepsilon J_T \alpha e} \\
&= -\dot{e}^T K_v \dot{e} - e^T \alpha(t)^T K_v \alpha(t) e
\end{aligned} \tag{B.1}$$

where the input control law (2.19) has been substituted.
Being $\dot{V} \leq 0$, the state is bounded in norm, $\dot{V}$ is uniformly continuous, and V tends to a limit value. Exploiting Barbalat lemma, it is possible to conclude that $\dot{V}$ tends to zero and thus $\dot{e}$ and αe tend to zero as well. Hence ε tends to a finite value $\bar{\varepsilon}$ since

$$\dot{\varepsilon} = J_T \big(\dot{e} + \alpha e\big) \to 0 \quad \Rightarrow \quad \varepsilon \to \bar{\varepsilon}.$$

The control input is bounded, since all its components are bounded. Also, $\ddot{e} = u$ tends to the finite limit $-K_\varepsilon J_T \bar{\varepsilon}$ and it is uniformly continuous, as $\dot{u}$ is bounded. Hence, the finite limit $-K_\varepsilon J_T \bar{\varepsilon}$ must be zero, according to Barbalat Lemma and thus also $\varepsilon \to 0$. □

Appendix C

Control with Polar Coordinates: details and proofs.

C.1 Details for section *Prescribed performance on the distance vector*

Substituting (3.5) in (3.4) we obtain

$$
\begin{aligned}
\dot{V} &= -k_1\varepsilon^2 J_T \cos^2\gamma - \varepsilon J_T \alpha(t) e\big(k_3\cos^2\gamma - 1\big) + \\
&\quad + \Big(k_1\frac{\varepsilon}{e} + k_3\alpha(t)\Big)\sin\gamma\cos\gamma(\gamma + k_3\delta) - k_2\gamma^2 + \\
&\quad - \Big(k_1\frac{\varepsilon}{e} + k_3\alpha(t)\Big)\sin\gamma\cos\gamma(\gamma + k_3\delta) - k_3\varepsilon J_T\alpha(t)e\sin^2\gamma \\
&= -k_1\varepsilon^2 J_T\cos^2\gamma - k_2\gamma^2 - \varepsilon J\alpha(t)e\big(k_3(\sin^2\gamma + \cos^2\gamma) - 1\big) \\
&= -k_1\varepsilon^2 J_T\cos^2\gamma - k_2\gamma^2 - \varepsilon J\alpha(t)e(k_3 - 1)
\end{aligned}
$$

The following inequality holds (see [10]):

$$
\varepsilon J e \geq \mu\varepsilon^2 \tag{C.1}
$$

with μ a positive constant. Hence, provided that $k_3 \geq 1$ and taking $\mu > 0$, then

$$
\begin{aligned}
\dot{V} &= -k_1\varepsilon^2 J_T\cos^2\gamma - k_2\gamma^2 - \varepsilon J_T\alpha(t)e(k_3 - 1) \\
&\leq -k_1\varepsilon^2 J_T\cos^2\gamma - k_2\gamma^2 - \mu\varepsilon^2\alpha(t)(k_3 - 1) \\
&\leq -k_1\varepsilon^2 J_T\cos^2\gamma - k_2\gamma^2 \leq 0.
\end{aligned} \tag{C.2}
$$

Proof of Proposition 3.1

Since (3.6) holds, the state is bounded in norm, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to a limit value. By Barbalat lemma, $\dot{V}(t)$ tends to zero and thus also ε and γ do: $q(t) \to \left\{q : \dot{V}(x) = 0\right\} = \{\varepsilon = 0,\ \gamma = 0,\ \forall\delta\}$.

Analyzing the closed-loop system, we note that $\dot{r}$ and $\dot{\delta}$ converge to zero.

In fact,

$$\dot{r} = -v\cos\gamma = -k_1\varepsilon\cos^2\gamma - k_3\alpha(t)e\cos^2\gamma \to 0$$

as $\varepsilon \to 0$ (and consequently also $e \to 0$), and also

$$\dot{r} = -v\cos\gamma = -k_1\varepsilon\cos^2\gamma - k_3\alpha(t)e \to 0$$

as $\varepsilon \to 0$ (and consequently also $e \to 0$), and also

$$\dot{\delta} = \frac{\sin\gamma}{r}v = k_1\frac{\varepsilon}{e}\sin\gamma\cos\gamma + k_3\alpha(t)\sin\gamma\cos\gamma \to 0$$

by the fact that, as $\gamma \to 0$ then $\sin\gamma \to 0$, $\cos\gamma \to 1$.

Also, δ converges to some finite limit $\bar{\delta}$.

Then, substituting the expression of v and ω in the evolution equation of γ we have

$$\begin{aligned}\dot{\gamma} &= \frac{\sin\gamma}{r}v - \omega \\ &= k_1\frac{\varepsilon}{e}\sin\gamma\cos\gamma + k_3\alpha(t)\sin\gamma\cos\gamma + \\ &\quad - k_2\gamma - \left(k_1\frac{\varepsilon}{e} + k_3\alpha(t)\right)\frac{\sin\gamma\cos\gamma}{\gamma}(\gamma + k_3\delta) + \\ &\quad - k_3\varepsilon J\alpha(t)e\frac{\sin^2\gamma}{\gamma}\end{aligned}$$

As $\delta \to \bar{\delta}$, $\dot{\gamma}$ tends to the finite limit

$$\dot{\gamma} \to -k_1k_3\frac{\varepsilon}{e}\bar{\delta} - k_3^2\alpha(t)\bar{\delta}$$

provided that $T(\cdot) = T_b(\cdot)$ or also $T(\cdot) = T_a(\cdot)$ but choosing $M = 0$ (see section 2.2). Since $\alpha(t) \to 0$ and choosing $T(\cdot) = T_b(\cdot)$

$$\dot{\gamma} \to -k_1k_3\frac{M+1}{M\rho}\bar{\delta}$$

Also, $\dot{\gamma}$ is uniformly continuous since $\ddot{\gamma}$ is bounded.
Taking the second derivative of γ we obtain:

$$\ddot{\gamma} = -\frac{\dot{r}}{r^2}\sin\gamma v + \frac{\cos\gamma\dot{\gamma}}{r}v + \frac{\sin\gamma}{r}\dot{v} - \dot{\omega}.$$

Both v and ω are bounded; hence $\dot{r} = -v\cos\gamma$ is still bounded (and thus also $\dot{\varepsilon}$). If $r \neq 0$, also $\dot{\gamma} = \frac{\sin\gamma}{r}v - \omega$ is bounded. We can observe that $r = 0$ is a singularity which intrinsically exists from the definition of the polar coordinates system.
Thus, also

$$\dot{v} = k_1\varepsilon J(\dot{r} + \alpha e)\cos\gamma - k_1\varepsilon\sin\gamma\dot{\gamma} + k_3\dot{\alpha}e\cos\gamma + k_3\alpha\dot{e}\cos\gamma - k_3\alpha e\sin\gamma\dot{\gamma}$$

is bounded. The last term to be checked is then $\dot{\omega}$. Writing extensively the expression of $\dot{\omega}$ we have:

$$\begin{aligned}\dot{\omega} = k_2\dot{\gamma} &+ \Big(k_1\frac{\dot{\varepsilon}e - \varepsilon\dot{e}}{e^2} + k_3\dot{\alpha}\Big)\frac{\sin\gamma\cos\gamma}{\gamma}(\gamma + k_3\delta) + \\ &+ \Big(k_1\frac{\varepsilon}{e} + k_3\alpha\Big)\Big(-\frac{1}{\gamma} + \frac{2\cos^2\gamma}{\gamma} - \frac{\sin\gamma\cos\gamma}{\gamma^2}\Big)\dot{\gamma}(\gamma + k_3\delta) + \\ &+ \Big(k_1\frac{\varepsilon}{e} + k_3\alpha\Big)\frac{\sin\gamma\cos\gamma}{\gamma}(\dot{\gamma} + k_3\dot{\delta}) + k_3\dot{\varepsilon}J\alpha e\frac{\sin^2\gamma}{\gamma} + \\ &+ k_3\varepsilon J\dot{\alpha}e\frac{\sin^2\gamma}{\gamma} + k_3\varepsilon J\alpha\dot{e}\frac{\sin^2\gamma}{\gamma} + k_3\varepsilon J\alpha e\frac{2\sin\gamma\cos\gamma - \sin^2\gamma}{\gamma^2}\dot{\gamma}.\end{aligned} \tag{C.3}$$

All the terms are bounded or vanish as γ goes to zero. In particular, $\Big(-\frac{1}{\gamma} + \frac{2\cos^2\gamma}{\gamma} - \frac{\sin\gamma\cos\gamma}{\gamma^2}\Big)$ and $\frac{2\sin\gamma\cos\gamma - \sin^2\gamma}{\gamma^2}$ remain bounded as long as γ stays away from zero and tend to zero and 1 respectively as $\gamma \to 0$.
Eventually, $\ddot{\gamma}$ is bounded.

According to the theoretical analysis carried out so far, since γ has a finite limit as $t \to \infty$ and $\dot{\gamma}$ is uniformly continuous, then $\dot{\gamma}$ tends to zero. Being $E \neq 0$, it follows that $\bar{\delta}$ must be zero, so that δ converges to zero.

From the previous analysis, ε is proved bounded, thus the transformation $T(\hat{e})$ is bounded. Hence, it is possible to conclude that r respects the predefined limits. □

C.2 Details for bounds on the angles

Bounds on γ

If we bind γ and put prescribed performance on it, exploiting the transformation

$$e = \gamma \quad \mapsto \quad \varepsilon = T\Big(\frac{e}{\rho}\Big)$$

and consider the Lyapunov function

$$V = \frac{1}{2}\big(r^2 + \varepsilon^2 + \delta^2\big), \tag{C.4}$$

differentiating (C.4) wrt to time and substituting the dynamics of the model written with polar coordinates (3.2) we have

$$\dot{V} = r(-v\cos\gamma) + \varepsilon J\Big(\frac{\sin\gamma}{r}v - \omega + \alpha\gamma\Big) + \delta\frac{\sin\gamma}{r}v. \tag{C.5}$$

We can define the following control law

$$\begin{aligned} v &= k_1 r\cos\gamma \\ \omega &= k_2\varepsilon J + \alpha\gamma + \frac{\sin\gamma}{r}v + \tilde{\omega} \end{aligned} \tag{C.6}$$

with

$$\tilde{\omega} = \frac{k_1}{J}\delta\frac{\frac{\sin\gamma\cos\gamma}{\gamma}}{\frac{\varepsilon}{\gamma}}. \tag{C.7}$$

We notice that (C.7) is well defined and the inputs are bounded.
With this control law we have

$$\dot{V} = -k_1 r^2\cos^2\gamma - k_2\varepsilon^2 J^2 \leq 0. \tag{C.8}$$

However, when we try to calculate the second derivative of γ we find

$$\begin{aligned} \ddot{\gamma} &= \frac{d}{dt}\Big(\frac{\sin\gamma}{r}v - \omega\Big) \\ &= \frac{d}{dt}\Big(\frac{\sin\gamma}{r}\Big)v + \Big(\frac{\sin\gamma}{r}\Big)\dot{v} - \dot{\omega}. \end{aligned} \tag{C.9}$$

All the terms are well defined and bounded, except for $\dot{\omega}$. In particular, the critical term is $\dot{\tilde{\omega}}$:

$$\dot{\tilde{\omega}} = \frac{k_1}{J}\Big[\dot{\delta}\frac{s_\gamma c_\gamma/\gamma}{\varepsilon/\gamma} + \delta\Big(\frac{d}{dt}\Big(\frac{s_\gamma c_\gamma}{\gamma}\Big)\Big)\frac{\gamma}{\varepsilon} + \delta\frac{s_\gamma c_\gamma}{\gamma}\frac{d}{dt}\Big(\frac{\gamma}{\varepsilon}\Big)\Big] \tag{C.10}$$

where s_γ, c_γ indicate $\sin\gamma, \cos\gamma$. The only unbounded term in (C.10) is

$$\frac{d}{dt}\frac{\gamma}{\varepsilon} = \Big(\frac{\sin\gamma}{r}v - \omega\Big)\underbrace{\Big(\frac{1}{\varepsilon} - \frac{J\gamma}{\varepsilon^2}\Big)}_{unbounded} - \alpha J\Big(\frac{\gamma}{\varepsilon}\Big)^2. \tag{C.11}$$

The analysis carried on so far points out that with the control law (C.6) δ does not converge to zero but it settles on another value.

It is possible to reach practical convergence, since δ multiplies $\sin\gamma$. Hence, the only quadratic term that we can have for γ will always be multiplied by (at least) a square sine. Namely, we are referring to a controller of the form

$$\begin{aligned} v &= k_1 r\cos\gamma - k_3\delta r\sin\gamma \\ \omega &= k_2\varepsilon J + \alpha\gamma + \frac{\sin\gamma}{r}v + \tilde{\omega} \end{aligned} \tag{C.12}$$

with

$$\tilde{\omega} = (k_1 - k_3 r^2)\frac{s_\gamma c_\gamma}{\varepsilon J}\delta$$

With this control law we have

$$\dot{V} = -k_1 r^2 \cos^2\gamma - k_2 \varepsilon^2 J^2 - k_3 \delta^2 \sin^2\gamma \leq 0 \tag{C.13}$$

An alternative to this solution could be to define the linear velocity input with a component of the form

$$\tilde{v} = k_\delta \frac{r}{\sin\gamma}\delta. \tag{C.14}$$

This would allow to have a square δ term in $\dot{V}$, but we have to guarantee that γ always remains away from zero (that is actually our initial goal).

Bounds on δ

From an analytical point of view, if we put PP bounds on δ exploiting the transformation

$$e = \delta \quad \mapsto \quad \varepsilon = T\Big(\frac{e}{\rho}\Big)$$

and consider the Lyapunov function

$$V = \frac{1}{2}\big(r^2 + \gamma^2 + \varepsilon^2\big), \tag{C.15}$$

differentiating (C.15) wrt to time and substituting the dynamics of the model written with polar coordinates (3.2) we have

$$\dot{V} = r(-v\cos\gamma) + \gamma\Big(\frac{\sin\gamma}{r}v - \omega\Big) + \varepsilon J\Big(\frac{\sin\gamma}{r}v + \alpha\delta\Big). \tag{C.16}$$

Notice that we have a term $\varepsilon J \alpha \delta$ that we can not cancel out: in order to cancel out this term we would define the following control law

$$\begin{aligned} v &= k_3 r \varepsilon J \sin\gamma \\ \omega &= k_2\gamma + \varepsilon J \alpha \frac{\delta}{\gamma} + k_3 r^2 \frac{s_\gamma c_\gamma}{\gamma}\varepsilon J - k_3 r \varepsilon J \sin^2\gamma, \end{aligned} \tag{C.17}$$

but in this way we would have a critical term $\frac{\delta}{\gamma}$.
Substituting (C.17) in (C.16) we have

$$\dot{V} = -k_2\gamma^2 - k_3 \varepsilon^2 J^2 \sin^2\gamma \leq 0. \tag{C.18}$$

However, this result does not guarantee the convergence of r and allow to conclude just for a practical convergence of ε, by tuning k_2, k_3.

If we define a control law as

$$\begin{aligned} v &= k_3 r \varepsilon J \sin\gamma \\ \omega &= k_2\gamma + k_3 r^2 \frac{s_\gamma c_\gamma}{\gamma}\varepsilon J - k_3 r \varepsilon J \sin^2\gamma, \end{aligned} \tag{C.19}$$

substituting (C.19) in (C.16) we have

$$\dot{V} = -k_2\gamma^2 - k_3\varepsilon^2 J^2 \sin^2\gamma + \varepsilon J \alpha \delta. \tag{C.20}$$

Recalling that the following inequality holds:

$$\varepsilon J e \geq \mu \varepsilon^2$$

with μ a positive constant, it is easy to see that (C.20) can just be

$$\dot{V} \leq \bar{V},$$

with $\bar{V} = \varepsilon J \alpha \delta \geq 0$.

C.3 Details and proof for section *Bounds on the angle γ through a different Lyapunov function*

Differentiating (3.8) wrt to time we obtain

$$\begin{aligned} \dot{V} &= r(-v\cos\gamma) + \tan\gamma\Big(\frac{\sin\gamma}{r}v - \omega\Big) + k_3\delta\frac{\sin\gamma}{r}v \\ &= r(-v\cos\gamma) + \tan\gamma(-\omega) + \Big(k_3\delta + \tan\gamma\Big)\frac{\sin\gamma}{r}v. \end{aligned} \tag{C.21}$$

Substituting the control inputs defined by eq. 3.9, one can easily obtain the expression 3.10.

Proof of Convergence: [*Sketch*] Equation (3.10) implies that the state is bounded in norm, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to a limit value. By Barbalat lemma, $\dot{V}(t)$ tends to zero and thus also r and γ do. Analyzing the closed-loop system, we note that $\dot{r}$ and $\dot{\delta}$ converge to zero and δ converges to a finite limit; $\dot{\gamma}$ tends to a finite limit also and it is uniformly continuous since $\ddot{\gamma}$ is bounded. Hence we can conclude that all the coordinates converge to zero.

Also, from this analysis, we know that $-\ln(\cos(\hat{\gamma}))$ is bounded. Hence, γ is guaranteed to respect the predefined bounds. □

C.4 Details for section *Time-varying bounds on* γ

The first derivative of $\hat{\gamma}$ is

$$\dot{\hat{\gamma}} = \frac{\dot{\gamma}\rho(t) + \gamma\dot{\rho}(t)}{\rho^2(t)} = \frac{\dot{\gamma}}{\rho(t)} - \frac{\gamma}{\rho(t)}\frac{\dot{\rho}(t)}{\rho(t)} = \frac{1}{\rho(t)}\dot{\gamma} + \alpha(t)\hat{\gamma}$$

So, differentiating V wrt to time we obtain

$$\begin{aligned}\dot{V} &= r(-v\cos\gamma) + \tan\hat{\gamma}\Big(\frac{1}{\rho(t)}\dot{\gamma} + \alpha(t)\hat{\gamma}\Big) + k_3\delta\frac{\sin\gamma}{r}v \\ &= r(-v\cos\gamma) + \tan\hat{\gamma}\Big(\frac{1}{\rho(t)}\big(\frac{\sin\gamma}{r}v - \omega\big) + \alpha(t)\hat{\gamma}\Big) + k_3\delta\frac{\sin\gamma}{r}v \qquad \text{(C.22)} \\ &= r(-v\cos\gamma) + \tan\hat{\gamma}\Big(-\frac{1}{\rho(t)}\omega + \alpha(t)\hat{\gamma}\Big) + \Big(k_3\delta + \frac{\tan\hat{\gamma}}{\rho(t)}\Big)\frac{\sin\gamma}{r}v.\end{aligned}$$

Substituting the controllers defined by eq. (3.13) one can easily obtain

$$\dot{V} = -k_1 r^2\cos^2\gamma - \frac{k_2}{\rho(t)}\tan^2\hat{\gamma} \leq 0.$$

Notice that the control inputs are well defined and bounded. In particular, the term $\frac{\sin\gamma}{\sin\hat{\gamma}}$, which appears in the steering velocity input, is bounded as long as $\hat{\gamma}$ stays away from zero, and as $\hat{\gamma} \to 0$ we have

$$\lim_{\hat{\gamma}\to 0}\frac{\sin\gamma}{\sin\hat{\gamma}} = \lim_{\gamma\to 0}\frac{\sin\gamma}{\sin\big(\frac{\gamma}{\rho(t)}\big)} = \rho(t).$$

C.5 Details and proof for section *Bounds on both radial and angle coordinate*

The control inputs (3.16) are well defined and bounded.

In details:

- $\frac{\varepsilon_r}{r}$ is bounded as long as r stays away from zero and converge to the finite value $E_r = \frac{1+M}{\rho_r M}$ as $r \to 0$ (see Section 2);
- $\frac{\sin\gamma}{\sin\hat{\gamma}}$ is bounded as long as $\hat{\gamma}$ stays away from zero and converge to the finite value $\rho_\gamma(t)$ as γ (or in the same way $\hat{\gamma}$) tends to zero (see Section 3.3.3).

Substituting the designed controller in $\dot{V}$ we have

$$\dot{V} = -k_1\varepsilon^2 J_r^2\cos^2\gamma - \frac{k_2}{\rho_\gamma}\tan^2\hat{\gamma} \leq 0. \qquad \text{(C.23)}$$

The proof for the convergence of the coordinates can be carried out exploiting LaSalle theorem and Barbalat Lemma.
[*Sketch*] Equation (C.23) implies that the state is bounded in norm, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to a limit value. By Barbalat lemma, $\dot{V}(t)$ tends to zero and thus also ε and $\hat{\gamma}$ do (and hence r and γ also). Analyzing the closed-loop system, we note that $\dot{r}$ and $\dot{\delta}$ converge to zero and δ converges to a finite limit; $\dot{\gamma}$ tends to a finite limit also and it is uniformly continuous since $\ddot{\gamma}$ is bounded. To prove that $\ddot{\gamma}$ is bounded we notice that all its terms are bounded and in particular also $\dot{\omega}$. This term requires more attention: specifically we have that

$$\begin{aligned}
\frac{d}{dt}\frac{\sin\gamma}{\sin\left(\frac{\gamma}{\rho(t)}\right)} = \frac{d}{dt}\frac{\sin\gamma}{\sin\hat{\gamma}} &= \frac{\cos\gamma\sin\hat{\gamma}\dot{\gamma} - \sin\gamma\cos\hat{\gamma}\dot{\hat{\gamma}}}{\sin^2\hat{\gamma}} = \\
&= \frac{\cos\gamma}{\sin\hat{\gamma}}\dot{\gamma} - \frac{\sin\gamma}{\sin\hat{\gamma}}\frac{\cos\hat{\gamma}}{\sin\hat{\gamma}}\left[\frac{1}{\rho(t)}\dot{\gamma} + \alpha(t)\hat{\gamma}\right] = \\
&= \left[\frac{\cos\gamma}{\sin\hat{\gamma}} - \frac{1}{\rho(t)}\frac{\sin\gamma}{\sin\hat{\gamma}}\frac{\cos\hat{\gamma}}{\sin\hat{\gamma}}\right]\dot{\gamma} + \alpha\hat{\gamma}\frac{\sin\gamma}{\sin\hat{\gamma}}\frac{\cos\hat{\gamma}}{\sin\hat{\gamma}} = \\
&= \left[\frac{\cos\gamma}{\sin\hat{\gamma}} - \frac{1}{\rho(t)}\frac{\sin\gamma}{\sin\hat{\gamma}}\frac{\cos\hat{\gamma}}{\sin\hat{\gamma}}\right]\dot{\gamma} + \alpha\frac{\hat{\gamma}}{\sin\hat{\gamma}}\frac{\sin\gamma}{\sin\hat{\gamma}}\cos\hat{\gamma}
\end{aligned} \tag{C.24}$$

The second term is well defined and bounded, while term in the squared brackets

$$\frac{\cos\gamma}{\sin\hat{\gamma}} - \frac{1}{\rho(t)}\frac{\sin\gamma}{\sin\hat{\gamma}}\frac{\cos\hat{\gamma}}{\sin\hat{\gamma}}$$

is bounded as $\hat{\gamma} \neq 0$, and it tends to zero as $\hat{\gamma} \to 0$. Eventually, $\ddot{\gamma}$ is bounded, $\dot{\gamma}$ is uniformly continuous and tends to zero. Hence we can conclude that all the coordinates converge to zero. □

Appendix D

Time-varying control: details and proofs.

D.1 Details on the error definition

The error vector is defined as

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}$$

The expression for $\dot{e}$ is here obtained: first we can rewrite e as

$$e = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \tag{D.1}$$

where

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

and then we have

$$\begin{aligned} \dot{e} &= \begin{bmatrix} \dot{R} \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix} + R \frac{d}{dt} \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix} \\ \omega_d - \omega \end{bmatrix} \\ &= \begin{bmatrix} \dot{R}R^T e + R v_d \begin{bmatrix} \cos\theta_d \\ \sin\theta_d \end{bmatrix} - R \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \\ \omega_d - \omega \end{bmatrix} \\ &= \begin{bmatrix} S(\omega) e + \begin{bmatrix} \cos e_3 \\ \sin e_3 \end{bmatrix} v_d - \begin{bmatrix} 1 \\ 0 \end{bmatrix} v \\ \omega_d - \omega \end{bmatrix}. \end{aligned} \tag{D.2}$$

The last equality derives from the following relation:

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta_d \\ \sin\theta_d \end{bmatrix} = \begin{bmatrix} \cos(\theta-\theta_d) \\ \sin(\theta_d-\theta) \end{bmatrix} \tag{D.3}$$

and $S(\omega)$ is a skew-symmetric matrix:

$$S(\omega) = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}$$

Substituting the defined inputs:

$$\begin{aligned} v &= \cos e_3 v_d - u_1 \\ \omega &= \omega_d - u_2 \end{aligned}$$

we find

$$\begin{aligned} \dot{e} &= \begin{bmatrix} S(\omega_d)e - S(u_2)e + \begin{bmatrix} 0 \\ \sin e_3 \end{bmatrix} v_d + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} - \begin{bmatrix} 0 & u_2 & 0 \\ -u_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \end{aligned} \tag{D.4}$$

D.2 Details and proof for section *Time invariant bounds on the orientation*

The input u_2 is bounded and well defined. Indeed, it is bounded as long as $\hat{e}_3$ stays away from zero, and as $\hat{e}_3 \to 0$ we have

$$\lim_{\hat{e}_3 \to 0} \frac{\sin e_3}{\sin \hat{e}_3} = \lim_{e_3 \to 0} \frac{\sin e_3}{\sin\left(\frac{e_3}{\bar{\rho}}\right)} = \bar{\rho}.$$

In order to exploit Barbalat Lemma, we need e_3 to be uniformly continuous, so we have to check if its second derivative is bounded.
The full expression for the second derivative of e_3 is

$$\ddot{e}_3 = \dot{u}_2 = -\frac{k_2\bar{\rho}}{k_3}\left[\dot{v}_d e_2 \frac{s_{e_3}}{t_{\hat{e}_3}} + v_d \dot{e}_2 \frac{s_{e_3}}{t_{\hat{e}_3}} + v_d e_2 \underline{\left(\frac{d}{dt}\frac{s_{e_3}}{t_{\hat{e}_3}}\right)}\right] - \left(1 + t_{\hat{e}_3}^2\right)\dot{\hat{e}}_3$$

where $s_{e_3} = \sin e_3$, $t_{e_3} = \tan e_3$ and the underlined term can be written as

$$\frac{d}{dt}\frac{s_{e_3}}{t_{\hat{e}_3}} = \frac{d}{dt}\left(\frac{s_{e_3}}{s_{\hat{e}_3}}\right) c_{\hat{e}_3} - \frac{s_{e_3}}{s_{\hat{e}_3}} s_{\hat{e}_3} \dot{\hat{e}}_3$$

In particular, we have to carry about the term

$$\frac{d}{dt}\frac{\sin e_3}{\sin\left(\frac{e_3}{\bar{\rho}}\right)}$$

which appears in the first derivative of u_2 (i.e. in the second derivative of e_3): in detail we have

$$\begin{aligned}\frac{d}{dt}\frac{\sin e_3}{\sin\left(\frac{e_3}{\bar{\rho}}\right)} &= \frac{d}{dt}\frac{\sin e_3}{\sin \hat{e}_3} = \frac{\cos e_3 \sin \hat{e}_3 \dot{e}_3 - \sin e_3 \cos \hat{e}_3 \dot{\hat{e}}_3}{\sin^2 \hat{e}_3} = \\ &= \frac{\cos e_3 \sin \hat{e}_3 \dot{e}_3 - \sin e_3 \cos \hat{e}_3 \dot{e}_3/\bar{\rho}}{\sin^2 \hat{e}_3} = \\ &= \left[\frac{\cos e_3}{\sin \hat{e}_3} - \frac{1}{\bar{\rho}}\frac{\sin e_3}{\sin \hat{e}_3}\frac{\cos \hat{e}_3}{\sin \hat{e}_3}\right]\dot{e}_3.\end{aligned} \tag{D.5}$$

The term

$$\frac{\cos e_3}{\sin \hat{e}_3} - \frac{1}{\bar{\rho}}\frac{\sin e_3}{\sin \hat{e}_3}\frac{\cos \hat{e}_3}{\sin \hat{e}_3}$$

is bounded as $\hat{e}_3 \neq 0$, and it tends to zero as $\hat{e}_3 \to 0$. Also, $\dot{e}_3 = u_2$ is bounded, as previously discussed.
Hence we can conclude that all the coordinates and the error components converge to zero. □

D.3 Details for section *Time-varying bounds on the orientation*

The first derivative of the Lyapunov function (4.22) is

$$\begin{aligned}\dot{V} &= k_2 e_1\big(\omega_d e_2 + u_1 - e_2 u_2\big) + k_2 e_2\big(\omega_d e_1 \sin e_3 v_d + e_1 u_2\big) + k_3 \frac{\sin \hat{e}_3}{\cos \hat{e}_3}\left(\frac{u_2 \rho(t) - e_3 \dot{\rho}(t)}{\rho^2(t)}\right) \\ &= k_2 e_1\big(u_1\big) + k_2 e_2\big(\sin e_3 v_d\big) + k_3 \frac{\sin \hat{e}_3}{\cos \hat{e}_3}\left(\frac{u_2 + e_3 \alpha(t)}{\rho(t)}\right)\end{aligned} \tag{D.6}$$

where $\alpha(t) = -\frac{\dot{\rho}(t)}{\rho(t)} > 0$.
Substituting the designed controllers (4.23) in (4.24) and canceling out some terms we obtain (4.24). As in the previous case, the proof for the convergence of the error components and for the Cartesian coordinates can be carried on exploiting LaSalle theorem and Barbalat Lemma.
In order to exploit Barbalat Lemma, we would need e_3 to be uniformly continuous, so we have to check if its second derivative is bounded.
The full expression for the second derivative of e_3 differ from the previous

case because of the term which derive from the derivative of $\rho = \rho(t)$ (now time-dependent):

$$\ddot{e}_3 = \dot{u}_2 = -e_3\dot{\alpha}(t) - \frac{k_2\rho(t)}{k_3}\Big[\dot{v}_d e_2 \frac{s_{e_3}}{t_{\hat{e}_3}} + v_d \dot{e}_2 \frac{s_{e_3}}{t_{\hat{e}_3}} + \underline{v_d e_2 \Big(\frac{d}{dt}\frac{s_{e_3}}{t_{\hat{e}_3}}\Big)}\Big] - \big(1 + t^2_{\hat{e}_3}\big)\dot{e}_3 +$$
$$- \frac{k_2\dot{\rho}(t)}{k_3} v_d e_2 \frac{\cos\hat{e}_3}{\sin\hat{e}_3}\sin e_3 - \big(1 + \tan^2\hat{e}_3\big)\dot{\hat{e}}_3.$$

The derivative of $\hat{e}_3$ is

$$\dot{\hat{e}}_3 = \frac{\dot{e}_3\rho(t) + e_3\dot{\rho}(t)}{\rho^2(t)} = \frac{u_2}{\rho(t)} - \frac{e_3}{\rho(t)}\frac{\dot{\rho}(t)}{\rho(t)} = \frac{1}{\rho(t)}u_2 + \alpha(t)\hat{e}_3$$

and the underlined term, in this case, can be written as

$$\frac{d}{dt}\Big(\frac{\sin e_3}{\sin\hat{e}_3}\cos\hat{e}_3\Big)$$

while

$$\begin{aligned} \frac{d}{dt}\frac{\sin e_3}{\sin\big(\frac{e_3}{\rho(t)}\big)} &= \frac{d}{dt}\frac{\sin e_3}{\sin\hat{e}_3} = \frac{\cos e_3 \sin\hat{e}_3 \dot{e}_3 - \sin e_3 \cos\hat{e}_3 \dot{\hat{e}}_3}{\sin^2\hat{e}_3} = \\ &= \frac{\cos e_3}{\sin\hat{e}_3}u_2 - \frac{\sin e_3}{\sin\hat{e}_3}\frac{\cos\hat{e}_3}{\sin\hat{e}_3}\Big[\frac{1}{\rho(t)}u_2 + \alpha(t)\hat{e}_3\Big] = \\ &= \Big[\frac{\cos e_3}{\sin\hat{e}_3} - \frac{1}{\rho(t)}\frac{\sin e_3}{\sin\hat{e}_3}\frac{\cos\hat{e}_3}{\sin\hat{e}_3}\Big]u_2 + \alpha\hat{e}_3\frac{\sin e_3}{\sin\hat{e}_3}\frac{\cos\hat{e}_3}{\sin\hat{e}_3} \end{aligned} \tag{D.7}$$

The term

$$\frac{\cos e_3}{\sin\hat{e}_3} - \frac{1}{\rho(t)}\frac{\sin e_3}{\sin\hat{e}_3}\frac{\cos\hat{e}_3}{\sin\hat{e}_3}$$

is bounded as $\hat{e}_3 \neq 0$, and it tends to zero as $\hat{e}_3 \to 0$. Also, $\dot{e}_3 = u_2$ is bounded, as previously discussed. Also the term

$$\hat{e}_3\frac{\sin e_3}{\sin\hat{e}_3}\frac{\cos\hat{e}_3}{\sin\hat{e}_3} = \frac{\hat{e}_3}{\sin\hat{e}_3}\frac{\sin e_3}{\sin\hat{e}_3}\cos\hat{e}_3$$

is well defined and bounded.

Appendix E

ROS Simulations: Code.

This appendix reports a brief description of the main parts of the code that implements the control laws in ROS environment.

E.1 Class UnicycleVelocityControllerNode

The UnicycleVelocityControllerNode class implements the controller node.

```
#include <ros/ros.h>
#include <turtlesim/Pose.h>
#include <turtlesim/Velocity.h>
#include "unicycle_velocity.hh"
```

Public Member Functions

- `void getROSParameters()`
  gets all required parameters from the Parameter Server;
- `void topicCallbackUnicycleStates(const turtlesim::Pose::ConstPtr& msg)`: this is the call back, assuming that the node subscribes the unicycle states, that is the position;
- `bool calculateUnicycleVel()`
  calculates the unicycle velocities and updates the unicycle position;
- `void publishControlVel()`
  published the `turtlesim::Velocity`;
- `std::vector<double> get_unicycle_pos()`
  returns the unicycle position, that is the Cartesian coordinates x, y, θ;
- `std::vector<double> get_unicycle_vel()`
  returns the unicycle velocities v, ω.

Public Attributes

- `ros::NodeHandle n_`:
  this is the main access point to communications with the ROS system;

- `ros::Publisher pub`:
  topic to publish;

- `ros::Subscriber sub`:
  topic to subscribe;

- `ros::Time last_publish_time`:
  variable to stamp the last publish time;

- `UnicycleVelocity unicycleVel`:
  object instance of type `UnicycleVelocity`;

- `bool initialized_unicycleVel`:
  set to `true` when `unicycleVel` is initialized;

- `bool params_OK`:
  set to `true` if all required parameters are in the Parameter Server.

This documentation refers to the following files:

- include/unicycle_velocity_controller.hh
- src/unicycle_velocity_controller.cpp

E.2 Class UnicycleVelocity

The UnicycleVelocity class is responsible for calculating the unicycle velocity while implementing the designed control laws and the error transformations for prescribed performance constraints.

Public Member Functions

- `void init(vector<double> initPos, vector<double> goalPos, vector<double> controller_parameters)`
  initializes the initial and goal unicycle positions, calculates the corresponding polar coordinates and initialized the tuning parameters;

- `vector<double> calculateUnicycleVelocity(vector<double> unicycle_pose, double realTime)`
  calculates and returns the unicycle velocities implementing the control laws designed with polar coordinates. This function calculates also the ρ function, the error transformation for prescribed performance constraints. It also performs the conversion to the polar coordinates in order to define the input controllers v, ω;

- `double getRho()`
  returns the ρ function;

- `double getError()`
  return the error;

- `bool isGoalReached()`
  returns true if the goal position is achieved.

This documentation refers to the following files:

- include/unicycle_velocity.hh

- src/unicycle_velocity.cpp