Control of Wheeled Mobile Robots: An Experimental Overview

Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma "La Sapienza"
`http://labrob.ing.uniroma1.it`

The subject of this chapter is the motion control problem of wheeled mobile robots (WMRs). With reference to the unicycle kinematics, we review and compare several control strategies for trajectory tracking and posture stabilization in an environment free of obstacles. Experiments are reported for SuperMARIO, a two-wheel differentially-driven mobile robot. From the comparison of the obtained results, guidelines are provided for WMR end-users.

1 Introduction

Wheeled mobile robots (WMRs) are increasingly present in industrial and service robotics, particularly when flexible motion capabilities are required on reasonably smooth grounds and surfaces [29]. Several mobility configurations (wheel number and type, their location and actuation, single- or multi-body vehicle structure) can be found in the applications, e.g, see [18]. 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. A detailed reference on the analytical study of the kinematics of WMRs is [2].

Beyond the relevance in applications, the problem of autonomous motion planning and control of WMRs has attracted the interest of researchers in view of its 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) [24].

In the absence of workspace obstacles, the basic motion tasks assigned to a WMR may be reduced to moving between two robot postures and following a given trajectory. From a control viewpoint, the peculiar nature of nonholonomic kinematics makes the second problem easier than the first; in fact, it is known [8] that feedback stabilization at a given posture cannot be achieved via smooth time-invariant control. This indicates that the problem is truly nonlinear; linear control is ineffective, even locally, and innovative design techniques are needed.

After a preliminary attempt at designing local controllers, the trajectory tracking problem was globally solved in [28] by using a nonlinear feedback

action, and independently in [12] and [11] through the use of dynamic feed-back linearisation. A recursive technique for trajectory tracking of nonholonomic systems in chained form can also be derived from the backstepping paradigm [17]. As for posture stabilization, both discontinuous and/or time-varying feedback controllers have been proposed. Smooth time-varying stabilization was pioneered by Samson [26], [27], while discontinuous (often, time-varying) control was used in various forms, e.g. see [1], [10], [21], [22], [32]. A recent addition to this class was presented in [14], where dynamic feedback linearisation has been extended to the posture stabilization problem.

While comparative simulations of several of the above methods are given in [9] for a unicycle and in [13] for a car-like vehicle, there is no extensive experimental testing on a single benchmark vehicle. The objective of this chapter is therefore to evaluate and compare the practical design and performance of control methods for trajectory tracking and posture stabilization, highlighting potential implementation problems related to kinematic or dynamic nonidealities, e.g. wheel slippage, discretisation and quantization of signals, friction and backlash, actuator saturation and dynamics.

All control designs are directly presented for the case of unicycle kinematics, the most common among WMRs, and experimentally tested on the laboratory prototype SuperMARIO. Nonetheless, most of the methods selected for comparison can be generalised to vehicles with more complex kinematics.

1.1 Organization of Contents

In Section 2 we classify the basic motion control tasks for WMRs. The modeling and main control properties are summarised in Section 3. In Section 4, the experimental setup used in our tests is described in detail.

Trajectory tracking controllers are presented in Section 5. After discussing the role of nominal feedforward commands (Section 5.1), three feedback laws are illustrated. They are based respectively on tangent linearisation along the reference trajectory and linear control design (Section 5.2), on a nonlinear Lyapunov-based control technique (Section 5.3), and on the use of dynamic feedback linearisation (Section 5.4). Comparative experiments on exact and asymptotic trajectory tracking are conducted in Section 5.5, using an eight-shaped desired trajectory.

The posture stabilization problem to the origin of the configuration space is considered in Section 6. Four conceptually different feedback methods are presented, using time-varying smooth (Section 6.1) or nonsmooth (Section 6.2) control laws, a discontinuous controller based on polar coordinates transformation (Section 6.3), and a stabilizing law based on dynamic feedback linearisation (Section 6.4). Results on forward and parallel parking experiments are reported.

Finally, in Section 7 the obtained results are summarised and compared in terms of performance, ease of control parameters tuning, sensitivity to non-idealities, and generalisability to other WMRs. In this way, some guidelines

are proposed to end-users interested in implementing control laws for WRMs. Open problems for further research are pointed out.

2 Basic Motion Tasks

The basic motion tasks that we consider for a WMR in an obstacle-free environment are (see Figure 2.1):

- *Point-to-point motion*: The robot must reach a desired goal configuration starting from a given initial configuration.
- *Trajectory following*: A reference point on the robot must follow a trajectory in the Cartesian space (i.e. a geometric path with an associated timing law) starting from a given initial configuration.

(a)

(b)
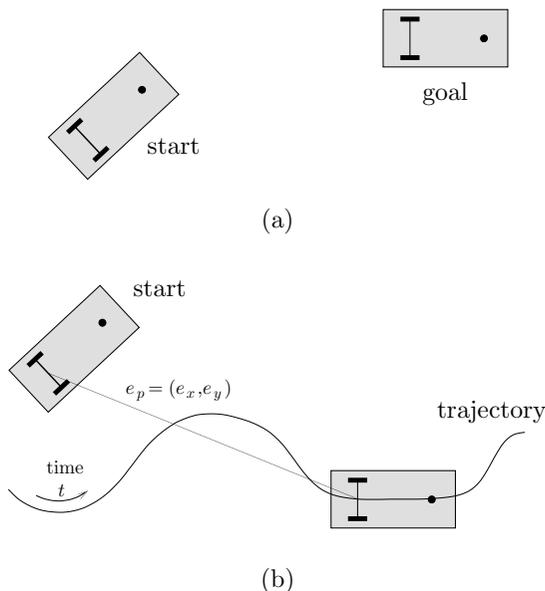
Figure 2.1. Basic motion tasks for a WMR: (a) point-to-point motion; (b) trajectory following

Execution of these tasks can be achieved using either feedforward commands, or feedback control, or a combination of the two. Indeed, feedback solutions exhibit an intrinsic degree of robustness. However, especially in the case of point-to-point motion, the design of feedback laws for nonholonomic systems has to face a serious structural obstruction, as we will show in Section 3; controllers that overcome such difficulty may lead to unsatisfactory transient

performance. The design of feedforward commands is instead strictly related to trajectory planning, whose solution should take into account the specific nonholonomic nature of the WMR kinematics.

When using a feedback strategy, the point-to-point motion task leads to a *state regulation* control problem for a point in the robot state space — *posture stabilization* is another frequently used term. Without loss of generality, the goal can be taken as the origin of the n-dimensional robot configuration space. As for trajectory following, in the presence of an initial error (i.e. an off-trajectory start for the vehicle) the *asymptotic tracking* control problem consists in the stabilization to zero of $e_p = (e_x, e_y)$, the two-dimensional Cartesian error with respect to the position of a moving reference robot (see Figure 2.1b).

Contrary to the usual situation, tracking is easier than regulation for a nonholonomic WMR. An intuitive explanation of this can be given in terms of a comparison between the number of controlled variables (outputs) and the number of control inputs. For the unicycle-like vehicle of Section 3, two input commands are available while three variables (x, y, and the orientation θ) are needed to determine its configuration. Thus, regulation of the WMR posture to a desired configuration implies zeroing three independent configuration errors. When tracking a trajectory, instead, the output e_p has the same dimension as the input and the control problem is square.

3 Modelling and Control Properties

Let $q \in \mathcal{Q}$ be the n-vector of generalised coordinates for a wheeled mobile robot. *Pfaffian* nonholonomic systems are characterised by the presence of $n - m$ non-integrable differential constraints on the generalised velocities of the form

$$A(q)\dot{q} = 0. \tag{3.1}$$

For a WMR, these arise from the rolling without slipping condition for the wheels. All feasible instantaneous motions can then be generated as

$$\dot{q} = G(q)w, \quad w \in I\!R^m, \tag{3.2}$$

where the columns g_i, $i = 1, \ldots, m$, of the $n \times m$ matrix $G(q)$ are chosen so as to span the null space of matrix $A(q)$. Different choices are possible for G, according to the physical interpretation that can be given to the 'weights' $w_1, \ldots, w_m$. Equation (3.2), which is called the (first-order) *kinematic model* of the system, represents a driftless nonlinear system.

The simplest model of a nonholonomic WMR is that of the *unicycle*, which corresponds to a single upright wheel rolling on the plane (top view in Figure 3.1). The generalised coordinates are $q = (x, y, \theta) \in \mathcal{Q} = I\!R^2 \times SO^1$ ($n = 3$). The constraint that the wheel cannot slip in the lateral direction is given in the form (3.1) as
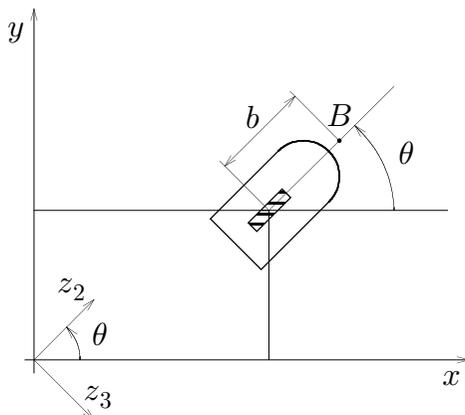
Figure 3.1. Relevant variables for the unicycle (top view)

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0.$$

A kinematic model is thus

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = g_1(q)v + g_2(q)\omega = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \qquad (3.3)$$

where v and ω (respectively, the linear velocity of the wheel and its angular velocity around the vertical axis) are assumed as available control inputs ($m = 2$). As we will show in Section 4, this model is equivalent to that of SuperMARIO.

System (3.3) displays a number of structural control properties, most of which actually hold more in general for (3.2).

3.1 Controllability at a Point

The tangent linearisation of (3.3) at any point q_e is the linear system

$$\dot{\tilde{q}} = \begin{bmatrix} \cos\theta_e \\ \sin\theta_e \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \qquad \tilde{q} = q - q_e,$$

that is clearly not controllable. This implies that a linear controller will never achieve posture stabilization, not even in a local sense. In order to study the controllability of the unicycle, we need therefore to use tools from nonlinear control theory [16]. It is easy to check that the *accessibility rank condition* is satisfied globally (at any q_e), since

$$\text{rank} \begin{bmatrix} g_1 & g_2 & [g_1, g_2] \end{bmatrix} = 3 = n, \qquad (3.4)$$

being the Lie bracket $[g_1, g_2]$ of the two input vector fields

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2 = \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix}.$$

Since the system is driftless, condition (3.4) implies its controllability.

Controllability can also be shown constructively, i.e. by providing an explicit sequence of maneuvers bringing the robot from any start configuration (x_s, y_s, θ_s) to any desired goal configuration (x_g, y_g, θ_g). Since the unicycle can rotate on itself, this task is simply achieved by an initial rotation on (x_s, y_s) until the unicycle is oriented toward (x_g, y_g), followed by a translation to the goal position, and by a final rotation on (x_g, y_g) so as to align θ with θ_g.

As for the stabilisability of system (3.3) to a point, the failure of the previous linear analysis indicates that exponential stability cannot be achieved by smooth feedback [31]. Things turn out to be even worse: if smooth (in fact, even continuous) time-invariant feedback laws are used, Lyapunov stability is out of reach. This negative result is established on the basis of a necessary condition due to Brockett [6]: smooth stabilisability of a driftless *regular* system (i.e. such that the input vector fields are well defined and linearly independent at q_e) requires a number of inputs equal to the number of states.

The above obstruction has a deep impact on the control design. In fact, to obtain a posture stabilizing controller it is either necessary to give up the continuity requirement and/or to resort to time-varying control laws. In Section 6 we shall pursue both approaches.

3.2 Controllability About a Trajectory

Given a desired Cartesian motion for the unicycle, it may be convenient to generate a corresponding *state* trajectory $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$. In order to be feasible, the latter must satisfy the nonholonomic constraint on the vehicle motion or, equivalently, be consistent with (3.3). The generation of $q_d(t)$ and of the corresponding reference velocity inputs $v_d(t)$ and $\omega_d(t)$ will be addressed in Section 5.

Defining the state tracking error as $\tilde{q} = q - q_d$ and the input variations as $\tilde{v} = v - v_d$ and $\tilde{\omega} = \omega - \omega_d$, the tangent linearisation of system (3.3) about the reference trajectory is

$$\dot{\tilde{q}} = \begin{bmatrix} 0 & 0 & -v_d \sin\theta_d \\ 0 & 0 & v_d \cos\theta_d \\ 0 & 0 & 0 \end{bmatrix} \tilde{q} + \begin{bmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} = A(t)\tilde{q} + B(t) \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix}.$$

$$(3.5)$$

Since the linearised system is *time-varying*, a necessary and sufficient controllability condition is that the controllability Gramian is nonsingular. However, a simpler analysis can be conducted by defining the state tracking error through a rotation matrix as

$$\tilde{q}_R = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 \\ -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}.$$

Using (3.5), we obtain

$$\dot{\tilde{q}}_R = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_R + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix}. \qquad (3.6)$$

When v_d and ω_d are constant, the above linear system becomes time-invariant and controllable, since matrix

$$\mathcal{C} = [\, B \quad AB \quad A^2B \,] = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_d^2 & v_d\omega_d \\ 0 & 0 & -\omega_d & v_d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

has rank 3 provided that either v_d or ω_d are nonzero. Therefore, we conclude that the kinematic system (3.3) can be locally stabilised by linear feedback about trajectories which consist of linear or circular paths, executed with constant velocity.

In Section 5 we shall see that it is possible to use linear design techniques in order to obtain local stabilization for arbitrary feasible trajectories, provided they do not come to a stop.

3.3 Feedback Linearisability

Based on the previous discussion, it is easy to see that the driftless nonholonomic system (3.2) cannot be transformed into a linear controllable one using *static* state feedback. In particular, for the unicycle (3.3) the controllability condition (3.4) implies that the distribution generated by vector fields g_1 and g_2 is not involutive, thus violating the necessary condition for full state feedback linearisability [16].

However, when matrix $G(q)$ in (3.2) has full column rank, m equations can always be transformed via feedback into simple integrators (input-output linearisation and decoupling). The choice of the linearising outputs is not unique and can be accommodated for special purposes. An interesting example is the following. Define the two outputs as

$$\begin{aligned} y_1 &= x + b\cos\theta \\ y_2 &= y + b\sin\theta, \end{aligned}$$

with $b \neq 0$, i.e. the Cartesian coordinates of a point B displaced at a distance b along the main axis of the unicycle (see Figure 3.1).

Using the globally defined state feedback

$$
\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta/b & \cos\theta/b \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},
$$

the unicycle is equivalent to

$$
\begin{aligned}
\dot{y}_1 &= u_1 \\
\dot{y}_2 &= u_2 \\
\dot{\theta} &= \frac{u_2\cos\theta - u_1\sin\theta}{b}.
\end{aligned}
$$

As a consequence, a linear feedback controller for $u = (u_1, u_2)$ will make the point B track *any* reference trajectory, even with discontinuous tangent to the path (e.g. a square without stopping at corners). Moreover, it is easy to show that the internal state evolution $\theta(t)$ is bounded. This approach, however, will not be pursued in this chapter because of its limited interest for more general kinematics.

For exact linearisation purposes, one may also resort to the more general class of *dynamic* state feedback. In this case, the conditions for full state linearisation are less stringent and turn out to be satisfied for a large class of nonholonomic WMRs. However, there is a potential control singularity that has to be considered carefully. The use of dynamic feedback linearisation will be illustrated later both for asymptotic trajectory tracking (Section 5) and for posture stabilization (Section 6).

3.4 Chained Forms

The existence of canonical forms for kinematic models of nonholonomic robots allows a general and systematic development of both open-loop and closed-loop control strategies. The most useful canonical structure is the *chained form*, which in the case of two-input systems is

$$
\begin{aligned}
\dot{z}_1 &= u_1 \\
\dot{z}_2 &= u_2 \\
\dot{z}_3 &= z_2 u_1 \\
&\ \ \vdots \\
\dot{z}_n &= z_{n-1} u_1.
\end{aligned}
\tag{3.7}
$$

It has been shown that a two-input driftless nonholonomic system with up to $n = 4$ generalised coordinates can always be transformed in chained form by static feedback transformation [23]. As a matter of fact, most (but not all) WMRs can be transformed in chained form.

For the kinematic model (3.3) of the unicycle, we introduce the following globally defined coordinate transformation

$$\begin{aligned} z_1 &= \theta \\ z_2 &= x\cos\theta + y\sin\theta \\ z_3 &= x\sin\theta - y\cos\theta \end{aligned}$$

and static state feedback

$$\begin{aligned} v &= u_2 + z_3 u_1 \\ \omega &= u_1, \end{aligned} \tag{3.8}$$

obtaining

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1. \end{aligned} \tag{3.9}$$

Note that (z_2, z_3) is the position of the unicycle in a rotating left-handed frame having the z_2 axis aligned with the vehicle orientation (see Figure 3.1). Equation (3.9) is another example of static input-output linearisation, with z_1 and z_2 as linearising outputs. We note also that the transformation in chained form is not unique (see, e.g. [9]).

4 Target Vehicle: SuperMARIO

The experimental comparison of the control methods to be reviewed in this chapter has been performed on the mobile robot SuperMARIO, built in the Robotics Laboratory of our Department (Figure 4.1).

4.1 Physical Description

SuperMARIO is a two-wheel differentially-driven vehicle, a mobility configuration found in many wheeled mobile robots. The two wheels have radius $r = 9.93$ cm and are mounted on the same axle of length $d = 29$ cm. The wheel radius includes also the o-ring used to prevent slippage; the rubber is stiff enough that point contact with the ground can be assumed. A small passive off-centered wheel is used as a caster, mounted in the front of the vehicle at a distance of 29 cm from the rear axle. The aluminum chassis of the robot measures $46 \times 32 \times 30.5$ cm (l/w/h) and contains two motors, transmission elements, electronics, and four 12 V batteries. The total weight of the robot (including batteries) is about 20 kg and its center of mass is located slightly in front of the rear axle. This design choice limits the disturbance on robot motion induced by sudden reorientation of the caster. Each wheel is independently driven by a DC servomotor (by MCA) supplied at 24 V with a peak torque of 0.56 Nm. Each motor is equipped with an incremental encoder counting $n_e = 200$ pulses/turn and a gearbox with reduction ratio $n_r = 20$. On-board electronics multiplies by a factor $m = 4$ the number of pulses/turn of the encoders, representing the angular increments with 16 bits.
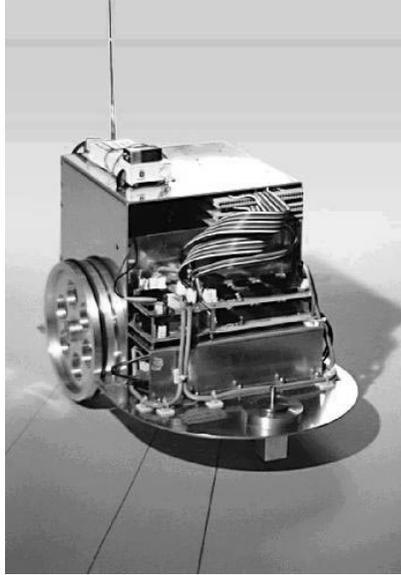
Figure 4.1. The wheeled mobile robot SuperMARIO

SuperMARIO is a low-cost prototype and presents therefore the typical nonidealities of electromechanical systems, namely friction, gear backlash, wheel slippage, actuator deadzone and saturation. These limitations clearly affect control performance. In addition, all controllers have been designed on the basis of a purely kinematic model. However, due to robot and actuator dynamics (masses and rotational inertias), velocity commands with discontinuous profile will not be exactly realised by the vehicle.

4.2 Control System Architecture

SuperMARIO has a two-level control architecture (see Figure 4.2). High-level control algorithms (including reference motion generation) are written in C^{++} and run with a sampling time of $T_s = 50$ ms on a remote server (a 300 MHz Pentium II), which also provides a user interface with real-time visualization as well as a simulation environment. The PC communicates through a radio modem with serial communication boards on the robot. The maximum speed of the radio link is 4800 bit/s. Wheel angular velocity commands ω_L and ω_R are sent to the robot and encoder measures $\Delta\phi_L$ and $\Delta\phi_R$ are received for odometric computations.

The low-level control layer is in charge of the execution of the high-level velocity commands. For each wheel, an 8-bit ST6265 microcontroller implements a digital PID with a cycle time of $T_c = 5$ ms. Two power amplifiers drive the motors with a 51 KHz PWM voltage.
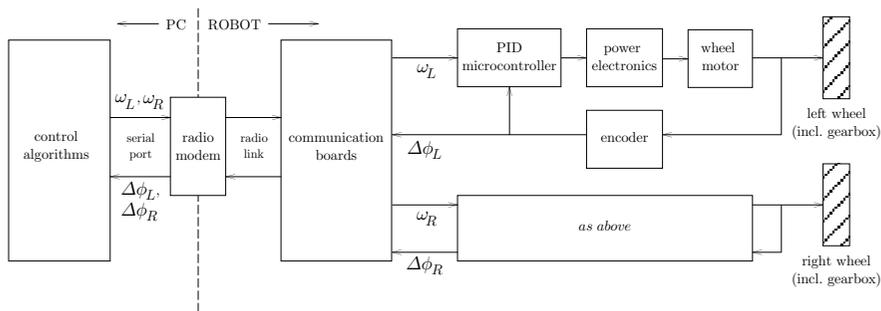
Figure 4.2. Control architecture of SuperMARIO

Custom interpolation algorithms have been developed on the PC so as to reduce the effect of quantization errors and communication delays in the reconstruction of the robot posture from the odometric information provided by the encoders. For all control schemes, an additional filtering of high-level velocity commands is included to account for vehicle and actuator dynamics. Simple first-order linear filters smooth possible discontinuities in the velocity profiles.

4.3 Kinematics

The kinematic model of SuperMARIO is given by (3.3), i.e. is equivalent to that of a unicycle. However, the actual commands are the angular velocities ω_R and ω_L of the right and left wheel, respectively, rather than the driving and steering velocities v and ω. There is, however, a one-to-one mapping between these velocities:

$$v = \frac{r\,(\omega_R + \omega_L)}{2} \qquad \omega = \frac{r\,(\omega_R - \omega_L)}{d}. \tag{4.1}$$

A calibration procedure has also been developed to estimate the actual wheel radii and axle length.

The reconstruction of the current robot configuration is based on in-cremental encoder data (odometry). Let $\Delta\phi_R$ and $\Delta\phi_L$ be the angular wheel displacements measured during the sampling time T_s by the encoders. From (4.1), we obtain the linear and angular displacements of the robot as

$$\Delta s = \frac{r}{2}\,(\Delta\phi_R + \Delta\phi_L) \qquad \Delta\theta = \frac{r}{d}\,(\Delta\phi_R - \Delta\phi_L).$$

The estimate of the posture at time $t_k = kT_s$ is computed as

$$\hat{q}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix} = \hat{q}_{k-1} + \begin{bmatrix} \cos\bar{\theta}_k & 0 \\ \sin\bar{\theta}_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix},$$

where[1]

$$\bar{\theta}_k = \hat{\theta}_{k-1} + \frac{\Delta\theta}{2}.$$

Robot localization using the above odometric prediction (commonly referred to as *dead reckoning*) is accurate enough in the absence of wheel slippage and backlash. These effects are however largely reduced when the velocity is kept reasonably small and the number of backup maneuvers is limited.

4.4 Control Constraints

Because of the bounded velocity capability of the motors, each wheel can achieve a maximum angular velocity Ω. Through (4.1), we obtain bounds on the driving and steering velocities:

$$|v| \leq \Omega r, \qquad |\omega| \leq \frac{2\Omega r}{d}.$$

There is, however, a more stringent constraint due to the limited resolution of the digital low-level control layer. In fact, the linear displacement resolution Δs_{min} of the robot can be computed from the previous data as

$$\Delta s_{min} = \frac{2\pi r}{m\, n_e\, n_r} = \frac{19.86\,\pi}{4 \cdot 200 \cdot 20} \simeq 0.0039 \text{ cm}.$$

This value corresponds to the least significant bit of the encoder, so that the average quantization error will be less than 2 hundreds of mm. In view of the 8-bit resolution of the on-board velocity microcontroller and of the PWM circuit (having $f_m = 1/T_c = 200$ Hz as minimum pulse frequency), the actual linear velocity command has the following threshold and saturation levels

$$v_m = f_m \Delta s_{min} \simeq 0.78 \text{ cm/s} \qquad v_M = v_m \cdot (2^8 - 1) = 198.9 \text{ cm/s}.$$

To prevent as much as possible wheel slippage, in our control software we have imposed the following even more conservative bounds on high-level velocity commands:

$$|v| \leq v_{max} = 0.3 \text{ m/s} \qquad |\omega| \leq \omega_{max} = 0.5 \text{ rad/s}.$$

With these saturations, it is necessary to perform a suitable velocity scaling so as to preserve the curvature radius corresponding to the nominal velocities v and ω. The actual commands v_c and ω_c are then computed by defining

$$\sigma = \max\left\{ \frac{|v|}{v_{max}}, \frac{|\omega|}{\omega_{max}}, 1 \right\},$$

and letting $v_c = v$ and $\omega_c = \omega$ if $\sigma = 1$, while

$$\text{if} \quad \sigma = |v|/v_{max} \quad \text{then} \quad v_c = \text{sign}(v)\, v_{max}, \ \ \omega_c = \omega/\sigma,$$

$$\text{else} \qquad\qquad\qquad\qquad v_c = v/\sigma, \ \ \omega_c = \text{sign}(\omega)\, \omega_{max}.$$

---

[1] Use of the average value $\bar{\theta}_k$ of the robot orientation is equivalent to the numerical integration of (3.3) via a 2nd order Runge-Kutta method.

5 Trajectory Tracking

The solution of the asymptotic tracking problem requires the combination of a nominal feedforward command with a feedback action on the error. In the control schemes to be presented, this error will be defined with respect to either the reference output trajectory (output error) or an associated reference state trajectory (state error).

5.1 Feedforward Command Generation

Assume that the representative point (x, y) of the unicycle must follow the Cartesian trajectory $(x_d(t), y_d(t))$, with $t \in [0, T]$ (possibly, $T \to \infty$). From the kinematic model (3.3) one has

$$\theta = \text{ATAN2}\,(\dot{y}, \dot{x}) + k\pi \qquad k = 0, 1, \tag{5.1}$$

where ATAN2 is the four-quadrant inverse tangent function (undefined only if both arguments are zero). Therefore, the nominal feedforward commands are

$$v_d(t) = \pm\sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \tag{5.2}$$

$$\omega_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}, \tag{5.3}$$

having differentiated (5.1) w.r.t. time in order to compute ω_d. The chosen sign for $v_d(t)$ will determine forward or backward motion of the vehicle. We note that, in order to be exactly reproducible using $v_d(t)$ and $\omega_d(t)$, the desired Cartesian motion $(x_d(t), y_d(t))$ should be twice differentiable in $[0, T]$.

A remarkable property of the unicycle is that, given an initial posture and a *consistent* desired output trajectory $(x_d(t), y_d(t))$ together with its derivatives, there is a unique associated state trajectory $q_d(t) = (x_d(t), y_d(t), \theta_d(t))$ which can be computed in a purely algebraic way[2], since

$$\theta_d(t) = \text{ATAN2}\,(\dot{y}_d(t), \dot{x}_d(t)) + k\pi \qquad k = 0, 1, \tag{5.4}$$

where the value of k is chosen so that $\theta_d(0) = \theta(0)$. If $k = 1$, a backward motion will result. Therefore, the nominal orientation $\theta_d(t)$ may be computed off-line and used for defining a state trajectory error.

Note the following facts.

- When the desired linear velocity $v_d(t)$ is zero for some $\bar{t}$, neither the nominal angular velocity nor the nominal orientation are defined from (5.3) and (5.4), respectively. This may occur at the initial instant, if a smooth start is specified, or at a cusp along the geometric path underlying the

---

[2] This is related to the fact that (x, y) is a *flat* output for the unicycle [15] or, equivalently, a linearising output under dynamic feedback (see Section 5.4).

Cartesian trajectory $(x_d(t), y_d(t))$. For the first case, one can use (if available) higher-order differential information about $(x_d(t), y_d(t))$ at $t = 0$ in order to determine the consistent initial orientation and the initial angular velocity command. For the second case, a continuous motion is guaranteed by keeping the same orientation attained at $\bar{t}^-$; by using de L'Hôpital analysis in (5.3), one can also compute the value of $\omega_d(\bar{t})$.

– More in general, the reference trajectory may be specified by separating the geometric aspects of the path (parameterised by a scalar s) from the timing law $s = s(t)$ used for path execution. The driftless nature of the kinematic model of a WMR allows to overcome in this way the above 'zero velocity' problem. For the unicycle, we can rewrite purely geometric relationships as

$$
\begin{bmatrix} dx/ds \\ dy/ds \\ d\theta/ds \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v' + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega',
$$

where the time commands are recovered as

$$
v(t) = v'(s)\dot{s}(t) \qquad \omega(t) = \omega'(s)\dot{s}(t).
$$

Zero-velocity points with well-defined geometric tangent (e.g. cusps) are then obtained for $\dot{s}(\bar{t}) = 0$. The feedforward pseudo-velocity commands $v'_d(s)$ and $\omega'_d(s)$ are computed by replacing in (5.2) and (5.3) time derivatives with space derivatives.

5.2 Linear Control Design

The simplest trajectory tracking control design is based on tangent linearisation along the reference trajectory. Following [9], it is worth to reconsider the linearisation procedure of the unicycle around the trajectory. Define the state tracking error e as

$$
\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}. \tag{5.5}
$$

The difference between e and $\tilde{q}_R$ of Section 2 is in the rotation matrix, which is computed here at the current orientation, and in a change of sign in the right hand side. Using the following nonlinear transformation of velocity inputs

$$
\begin{aligned}
v &= v_d \cos e_3 - u_1 \\
\omega &= \omega_d - u_2,
\end{aligned} \tag{5.6}
$$

the error dynamics becomes

$$
\dot{e} = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{5.7}
$$

Linearising (5.7) around the reference trajectory, we obtain the same linear time-varying equations (3.6), now with state e and input (u_1, u_2).

Define the linear feedback law

$$
\begin{aligned}
u_1 &= -k_1\, e_1 \\
u_2 &= -k_2\, \mathrm{sign}(v_d(t))\, e_2 - k_3\, e_3.
\end{aligned}
\tag{5.8}
$$

A desired closed-loop polynomial

$$(\lambda + 2\zeta a)(\lambda^2 + 2\zeta a\lambda + a^2) \qquad \zeta, a > 0,$$

namely having *constant* eigenvalues (one negative real at $-2\zeta a$ and a complex pair with natural angular frequency $a > 0$ and damping coefficient $\zeta \in (0,1)$) can be obtained by choosing the gains in (5.8) as

$$
k_1 = k_3 = 2\zeta a \qquad k_2 = \frac{a^2 - \omega_d(t)^2}{|v_d(t)|}.
$$

However, k_2 will go to infinity (i.e. an infinite control effort would be required for the same transient performance) as $v_d \to 0$. Therefore, a convenient *gain scheduling* is achieved by letting $a = a(t) = \sqrt{\omega_d^2(t) + bv_d^2(t)}$ so that

$$
k_1 = k_3 = 2\zeta\sqrt{\omega_d^2(t) + bv_d^2(t)} \qquad k_2 = b\,|v_d(t)|,
\tag{5.9}
$$

where the factor $b > 0$ has been introduced as an additional degree of freedom. According to the controllability analysis in Section 2, these gains gracefully go to zero when local controllability around the (state) trajectory is lost because the latter stops.

In terms of the original control inputs, this design leads to the *nonlinear time-varying* controller

$$
\begin{aligned}
v &= v_d \cos(\theta_d - \theta) + k_1\left[\cos\theta(x_d - x) + \sin\theta(y_d - y)\right] \\
\omega &= \omega_d + k_2\, \mathrm{sign}(v_d)\left[\cos\theta(y_d - y) - \sin\theta(x_d - x)\right] + k_3(\theta_d - \theta).
\end{aligned}
\tag{5.10}
$$

It should be emphasised that, even if the closed-loop eigenvalues are constant and with negative real part, this control law does not guarantee the asymptotic stability of the state tracking error e, because the system is still time-varying. A complete Lyapunov-based stability analysis can be however carried out by including a simple nonlinear modification, as shown hereafter.

5.3 Nonlinear Control Design

Following [26], we present now a nonlinear design for trajectory tracking. Consider again (5.7). Define

$$
\begin{aligned}
u_1 &= -k_1(v_d(t), \omega_d(t))\, e_1 \\
u_2 &= -\bar{k}_2\, v_d(t)\frac{\sin e_3}{e_3}\, e_2 - k_3(v_d(t), \omega_d(t))\, e_3,
\end{aligned}
\tag{5.11}
$$

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

Theorem 5.1. *Assuming that v_d and ω_d are bounded with bounded derivatives, and that $v_d(t) \not\to 0$ or $\omega_d(t) \not\to 0$ when $t \to \infty$, the control law (5.11) globally asymptotically stabilises the origin $e = 0$.*

Proof. (Sketch) It is based on the use of the Lyapunov function

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

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

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

Thus, $\|e(t)\|$ is bounded, $\dot{V}(t)$ is uniformly continuous, and $V(t)$ tends to some limit value. Using Barbalat lemma, $\dot{V}(t)$ tends to zero. From this and analyzing the system equations, one can show that $(v_d^2 + \omega_d^2)e_i^2$ $(i = 1, 2, 3)$ tends to zero so that, from the persistency of the (state) trajectory, the result follows. ∎

Merging (5.5), (5.6), and (5.11), the resulting control law is

$$v = v_d \cos(\theta_d - \theta) + k_1(v_d, \omega_d)\left[\cos\theta(x_d - x) + \sin\theta(y_d - y)\right]$$

$$(5.12)$$

$$\omega = \omega_d + \bar{k}_2\, v_d\, \frac{\sin(\theta_d - \theta)}{\theta_d - \theta}\left[\cos\theta(y_d - y) - \sin\theta(x_d - x)\right] + k_3(v_d, \omega_d)(\theta_d - \theta).$$

Taking advantage of the previous linear analysis, we can choose the gain functions k_1 and k_3 and the constant gain $\bar{k}_2$ as

$$k_1(v_d(t), \omega_d(t)) = k_3(v_d(t), \omega_d(t)) = 2\zeta\sqrt{\omega_d^2(t) + b v_d^2(t)} \qquad \bar{k}_2 = b,$$

with $b > 0$ and $\zeta \in (0, 1)$.

5.4 Dynamic Feedback Linearisation

A nonlinear controller for trajectory tracking based on exact dynamic feedback linearisation is now designed following [11], [12].

With reference to the general class of nonholonomic driftless systems (3.2), the dynamic feedback linearisation problem consists in finding, if possible, a dynamic state feedback compensator of the form

$$
\begin{aligned}
\dot{\xi} &= a(q, \xi) + b(q, \xi)u \\
w &= c(q, \xi) + d(q, \xi)u,
\end{aligned}
\qquad (5.13)
$$

with ν-dimensional state ξ and m-dimensional external input u, such that the closed-loop system (3.2)–(5.13) is equivalent, under a state transformation $z = T(q, \xi)$, to a linear controllable system.

Only necessary or sufficient (but no necessary and sufficient) conditions exist for the solution of the dynamic feedback linearisation problem. Constructive algorithms, which are essentially based on input-output decoupling, can be found in [16].

The starting point is the definition of an appropriate m-dimensional system output $\eta = h(q)$, to which a desired behaviour can be assigned (in our case, track a desired trajectory). One then proceeds by successively differentiating the output until the input appears in a nonsingular way. At some stage, the addition of integrators on a subset of the input channels may be necessary in order to avoid subsequent differentiation of the original inputs. This *dynamic extension* algorithm builds up the state ξ of the dynamic compensator (5.13). The algorithm terminates after a finite number of differentiations whenever the system is invertible from the chosen output. If the sum of the output differentiation orders equals the dimension $n + \nu$ of the extended state space, full input-state-output linearisation is also obtained. The closed-loop system is then equivalent to a set of decoupled input-output chains of integrators from u_i to η_i $(i = 1, \ldots, m)$.

We illustrate this exact linearisation procedure for the unicycle model (3.3). Define the linearising output vector as $\eta = (x, y)$. Differentiation w.r.t. time then yields

$$\dot{\eta} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix},$$

showing that only v affects $\dot{\eta}$, while the angular velocity ω cannot be recovered from this first-order differential information. In order to proceed, we need therefore to add an integrator (whose state is denoted by ξ) on the linear velocity input

$$v = \xi \qquad \dot{\xi} = a \qquad \Longrightarrow \qquad \dot{\eta} = \xi \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix},$$

being the new input a the *linear acceleration* of the unicycle. Differentiating further

$$\ddot{\eta} = \dot{\xi} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + \xi\dot{\theta} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\xi\sin\theta \\ \sin\theta & \xi\cos\theta \end{bmatrix} \begin{bmatrix} a \\ \omega \end{bmatrix}$$

and the matrix multiplying the modified input (a, ω) is nonsingular provided that $\xi \neq 0$. Under this assumption, we can define

$$\begin{bmatrix} a \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & -\xi\sin\theta \\ \sin\theta & \xi\cos\theta \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

so as to obtain

$$\ddot{\eta} = \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\eta}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = u. \tag{5.14}$$

The resulting dynamic compensator is

$$
\begin{aligned}
\dot{\xi} &= u_1 \cos\theta + u_2 \sin\theta \\
v &= \xi \\
\omega &= \frac{u_2 \cos\theta - u_1 \sin\theta}{\xi}.
\end{aligned}
\tag{5.15}
$$

Since the dynamic compensator is one-dimensional, we have $n+\nu = 3+1 = 4$, equal to the total number of output differentiations in (5.14). Therefore, in the new coordinates

$$
\begin{aligned}
z_1 &= x \\
z_2 &= y \\
z_3 &= \dot{x} = \xi \cos\theta \\
z_4 &= \dot{y} = \xi \sin\theta,
\end{aligned}
\tag{5.16}
$$

the extended system is fully linearised in a controllable form and described by the two chains of second-order input-output integrators given by (5.14), rewritten as

$$
\begin{aligned}
\ddot{z}_1 &= u_1 \\
\ddot{z}_2 &= u_2.
\end{aligned}
\tag{5.17}
$$

Note that the dynamic feedback linearising controller (5.15) has a potential singularity at $\xi = v = 0$, i.e. when the unicycle is not rolling. The occurrence of such singularity in the dynamic extension process has been shown to be structural for nonholonomic systems [12]. This difficulty must be obviously taken into account when designing control laws on the equivalent linear model.

Assume the robot must follow a smooth output trajectory $(x_d(t), y_d(t))$ which is persistent, i.e. such that the nominal control input $v_d = (\dot{x}_d^2 + \dot{y}_d^2)^{1/2}$ along the trajectory does never go to zero. On the equivalent linear and decoupled system (5.17), it is straightforward to design a globally exponentially stabilizing feedback for the desired trajectory (with linear Cartesian transients) as

$$
\begin{aligned}
u_1 &= \ddot{x}_d(t) + k_{p1}(x_d(t) - x) + k_{d1}(\dot{x}_d(t) - \dot{x}) \\
u_2 &= \ddot{y}_d(t) + k_{p2}(y_d(t) - y) + k_{d2}(\dot{y}_d(t) - \dot{y}),
\end{aligned}
\tag{5.18}
$$

with PD gains chosen as $k_{pi} > 0$, $k_{di} > 0$, for $i = 1, 2$.

In the implementation, velocities $\dot{x}$ and $\dot{y}$ can be computed via the last two expressions in (5.16), as a function of the robot state and of the compensator state ξ. Alternatively, one can use estimates of $\dot{x}$ and $\dot{y}$ obtained from odometric measurements. This solution is more robust with respect to unmodelled dynamics.

We conclude the discussion on trajectory tracking via dynamic feedback linearisation by offering some remarks:

- The state of the dynamic compensator should be correctly initialized at the value $\xi(0) = v_d(0)$. This guarantees exact trajectory tracking for a matched initial state of the robot. In this case, the control law (5.15–5.18) reduces to the pure feedforward action.
- Being based purely on an output tracking error definition, this method requires neither the explicit computation of $\theta_d(t)$ nor the measure of the orientation angle $\theta(t)$.
- Even for smooth persistent trajectories, problems may arise if the *actual* command $v = \xi$ crosses zero during an initial transient. However, this situation can be avoided by suitably choosing the initial state of the dynamic compensator. For example, a simple way to keep the actual commands bounded is to reset the state ξ whenever its value falls below a given threshold. This strategy results in an input command v with isolated discontinuities with respect to time.

5.5 Experiments

In the following we report experimental results of SuperMARIO following the eight-shaped reference trajectory of Figure 5.1, defined by

$$x_d(t) = \sin\frac{t}{10} \qquad y_d(t) = \sin\frac{t}{20} \qquad t \in [0,T].$$

The trajectory starts from the origin with $\theta_d(0) = \pi/6$ rad. A full cycle is completed in $T = 2\pi \cdot 20 \approx 125$ s. Note that the reference initial velocities are

$$v_d(0) \simeq 0.1118 \text{ m/s} \qquad \omega_d(0) = 0 \text{ rad/s}.$$

In the first set of experiments, the robot configuration is initially matched with the desired reference trajectory (i.e. with initial state $q(0) = q_d(0)$). Therefore, the feedforward commands (5.2), (5.3) would allow exact trajectory following in ideal conditions. However, if the unicycle starts at rest and non-zero high-level commands $v_d(0)$ and $\omega_d(0)$ are sent to the robot, due to the actuation/vehicle dynamics there will be some transient before these velocities are actually achieved at the physical low level. The addition of a feedback action to the feedforward command is also effective in recovering the induced state error.

Figures 5.2–5.4 show the results obtained with the linearly designed controller (5.10), using the gains (5.9) with $\zeta = 0.7$ and $b = 10$. The tracking of the reference trajectory of Figure 5.1 is indeed quite accurate. Residual errors are mainly due to quantisation and discretisation of velocity commands, as well as to other nonidealities. In particular, there is a large transient error due to the vehicle/actuator dynamics because of the initial non-zero value of $v_d(0)$. This is clearly shown in Figure 5.5 which shows the norm $\|e_p\|$ of the Cartesian error.
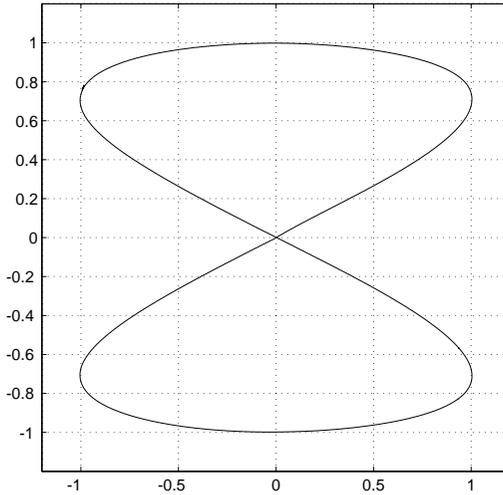
Figure 5.1. An eight-shaped reference trajectory

Similar performance is obtained with the nonlinear controller (5.12), using the same gains as above, and with the dynamic feedback linearisation controller (5.15), choosing the PD gains in (5.18) as $k_{p1} = k_{p2} = 1$, $k_{d1} = k_{d2} = 0.7$ and initializing the dynamic compensator at $\xi(0) = v_d(0)$. To appreciate the slight improvement in performance, compare the norm $\|e_p\|$ of the Cartesian error in Figures 5.6 and 5.7 with the previous result in Figure 5.5. It is found that the average error is reduced from a value of 1 cm (linear design) to 0.5 cm (nonlinear design) and finally to 0.38 cm (dynamic feedback linearisation design).

A second set of experiments was performed letting $q(0) = (0.2, -0.3, \pi/3)$ (m,m,rad), i.e. starting with an initial state error with respect to the assigned trajectory. Only the linearly designed controller and the dynamic feedback controller were compared (see Figures 5.8–5.11), using the same previous settings of control parameters. The obtained transients are quite similar, although a smaller overshoot is experienced with dynamic feedback linearisation, as implied by the choice of the PD gains.
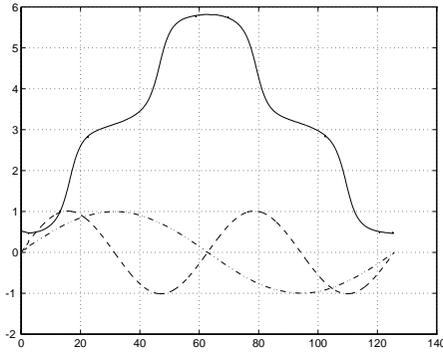
Figure 5.2. Trajectory tracking with linear feedback design: x ($--$), y ($-\cdot$) (m), and θ ($—$) (rad) vs. time (s)
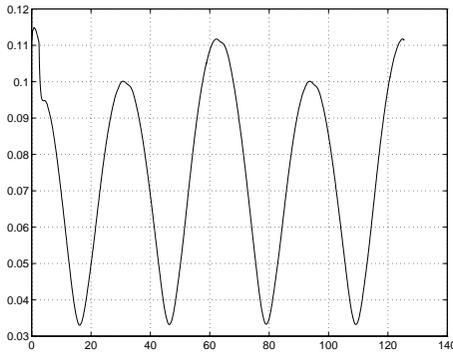
Figure 5.3. Trajectory tracking with linear feedback design: driving velocity v (m/s)
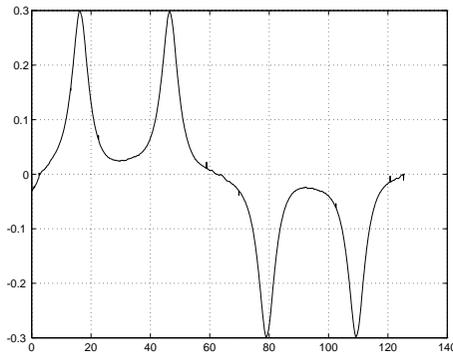
Figure 5.4. Trajectory tracking with linear feedback design: steering velocity ω (rad/s)
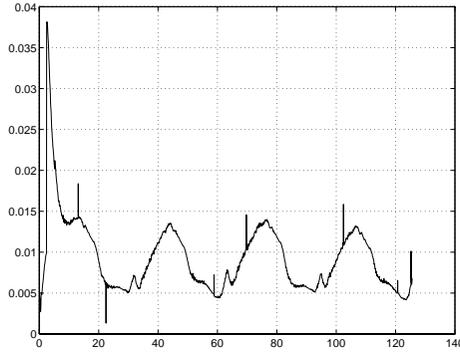
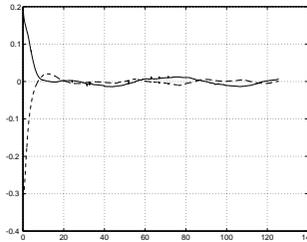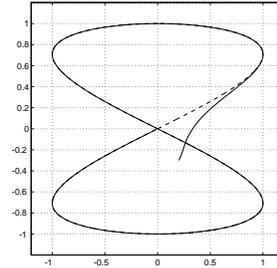**Figure 5.5.** Trajectory tracking with linear feedback design: norm of Cartesian error (m)
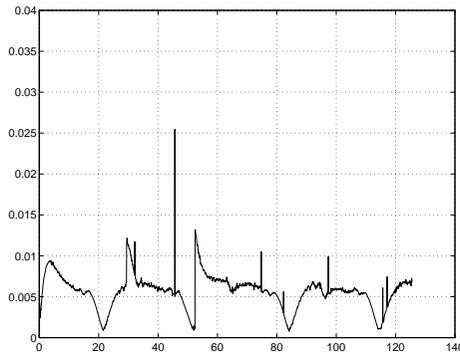
Figure 5.6. Trajectory tracking with nonlinear feedback design: norm of Cartesian error (m)
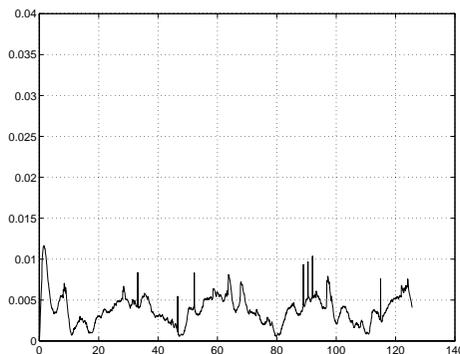
Figure 5.7. Trajectory tracking via dynamic feedback linearisation: norm of Cartesian error (m)
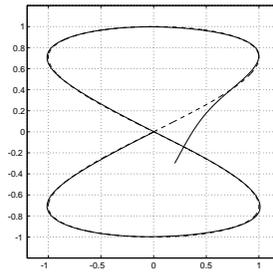
Figure 5.8. Asymptotic trajectory tracking with linear feedback design: Cartesian motion (x, y) (m)

Figure 5.9. Asymptotic trajectory tracking with linear feedback design: Cartesian errors e_x and e_y (m)

Figure 5.10. Asymptotic trajectory tracking via dynamic feedback linearisation: Cartesian motion (x, y) (m)
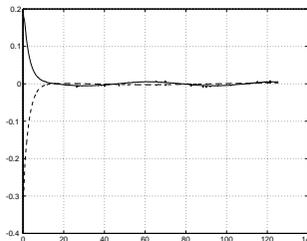
Figure 5.11. Asymptotic trajectory tracking via dynamic feedback linearisation: Cartesian errors e_x and e_y (m)

6 Posture Stabilisation

As mentioned in Section 3, posture stabilisation for nonholonomic WMRs cannot be achieved by smooth static feedback. We present two controllers based on time-varying feedback and two based on discontinuous feedback.

6.1 Smooth Time-varying Control

It has been shown in Section 5.3 that asymptotic stabilisation of a state tracking error can be achieved provided that $v_d(t)$ and $\omega_d(t)$ — which introduce a time-varying signal in the feedback control law — do not both vanish in finite time. This observation suggests that a solution to the posture stabilisation problem can be obtained by designing a desired motion, to be used as a fictitious time-varying reference, which asymptotically vanishes at the origin.

Following [26], the structure of a smooth time-varying stabilizing controller is the same of the nonlinear trajectory tracking controller (5.11):

$$
\begin{aligned}
u_1 &= -k_1(v_d(t), \omega_d(t))\, e_1 \\
u_2 &= -\bar{k}_2\, v_d(t)\frac{\sin e_3}{e_3}\, e_2 - k_3(v_d(t), \omega_d(t))\, e_3,
\end{aligned}
\tag{6.1}
$$

with the same notation used in Section 5.3 (in particular, (5.12) is used to generate v and ω). In this case, however, the desired trajectory is itself an additional degree of freedom, subject to the constraint that, under control (6.1), both the state tracking error e and the desired trajectory are asymptotically stabilised to the origin. A simple solution is to set, for all times t, $y_d(t) = 0$ and $\theta_d(t) = 0$ (and thus $\omega_d(t) = 0$), having only x_d in motion. A class of desired velocities is given then by

$$
v_d(t) = \dot{x}_d(t) = -k_4 x_d(t) + g(e, t) \qquad k_4 > 0,
\tag{6.2}
$$

where $g(e, t)$ is a C^2-function uniformly bounded with respect to t, together with its partial derivative, and such that:

A1. $g(0, t) = 0$, for all t;

A2. there exists a diverging sequence of instants $\{t_i\}_{i \in N}$ and a continuous function $\alpha(\cdot)$ for which

$$
\|e\| > \varepsilon > 0 \quad \Rightarrow \quad \left(\frac{\partial g}{\partial t}(e, t_i)\right)^2 > \alpha(\varepsilon) > 0 \quad \forall t_i.
$$

Theorem 6.1. *The smooth time-varying controller (6.1), with $v_d(t)$ given by (6.2) with A1 and A2, globally asymptotically stabilises $e = 0$ and $x_d = 0$.*

Proof. (Sketch) Using the same Lyapunov function as in Theorem 5.1, the tracking error $e(t)$ is shown to be bounded. Therefore, (6.2) is a stable linear system subject to an additive bounded perturbation $g(e, t)$, so that $x_d(t)$ and $v_d(t)$ (as well as $\dot{v}_d(t)$) remain bounded. Theorem 5.1 applies and the rest of the proof uses assumptions A1 and A2 to show first that $v_d(t)$ tends to zero, then that also $\frac{\partial g}{\partial t}(e, t)$ does, and concluding the convergence of e to zero. Finally, using A2, (6.2) implies that also $x_d(t)$ tends to zero. ∎

The *heating* function $g(e, t)$ plays a key role in guaranteeing asymptotic stability by sustaining motion as long as the error e is not zero, but it also determines the transient behaviour. Possible choices satisfying A1 and A2 include $g(e, t) = \|e\|^2 \sin t$ and, if the functions $k_1(\cdot, \cdot)$ and $k_3(\cdot, \cdot)$ are strictly positive,

$$g(e, t) = \frac{\exp(k_5 e_2) - 1}{\exp(k_5 e_2) + 1} \sin t \quad k_5 > 0. \tag{6.3}$$

We tested the performance of the time-varying control (6.1), with desired motion given by (6.2), initialised at $x_d(0) = 0$, and heating function (6.3). The gains have been selected as $k_1 = 0.5$, $\bar{k}_2 = 2$, $k_3 = 1$, $k_4 = 1$, and $k_5 = 50$.

A forward parking task from $q(0) = (-1, -1, 0)$ (m,m,rad) to the origin is assigned. This task will be used as a baseline for comparison of all posture stabilisation methods. Figures 6.1–6.4 show that, after a relatively fast approach, convergence of the smooth time-varying controller becomes extremely slow when the unicycle is close to the goal. In particular, this is evident in Figure 6.2, a stroboscopic view of the robot motion sampled every 10 s. An inherent limitation of this control design is the large number of backup maneuvers, executed with the unicycle approximately aligned with the final desired orientation. The oscillation range of x is contained in the range $(-1, 1)$, thanks to the large value of k_5. The steering velocity command becomes very small in the final phase.

The accuracy in regulation to the origin is determined by the satisfaction of the following terminal bounds, used in all the posture stabilisation experiments:
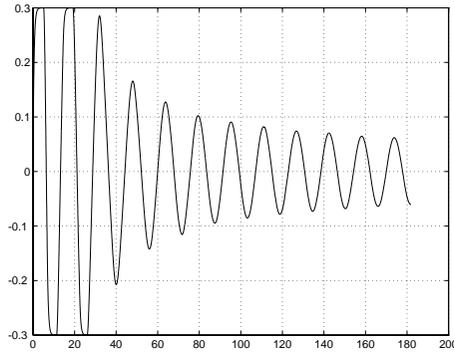
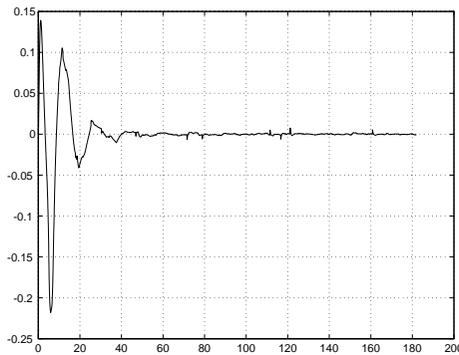$$|x| < 0.2 \text{ cm} \qquad |y| < 0.2 \text{ cm} \qquad |\theta| < 0.02 \text{ rad}.$$
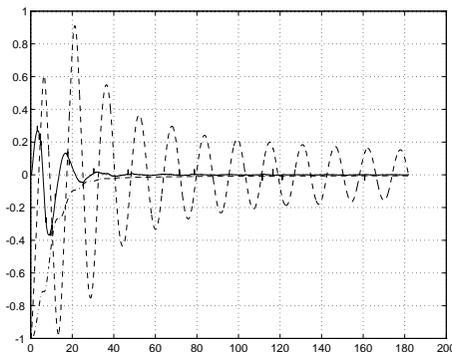
Figure 6.1. Posture stabilisation with smooth time-varying feedback: x (−−), y (−·) (m), and θ (—) (rad) vs. time (s)
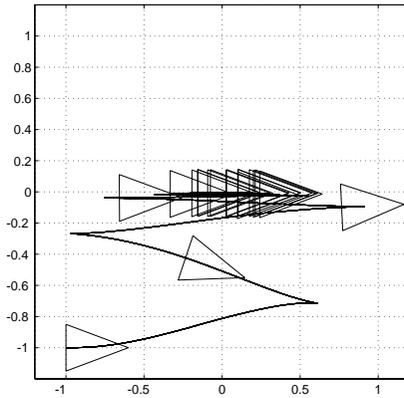
Figure 6.2. Posture stabilisation with smooth time-varying feedback: Cartesian motion (x, y) (m)

Figure 6.3. Posture stabilisation with smooth time-varying feedback: driving velocity v (m/s)

Figure 6.4. Posture stabilisation with smooth time-varying feedback: steering velocity ω (rad/s)

6.2 Nonsmooth Time-varying Control

By giving up the smoothness requirement, several controllers have been proposed for posture stabilisation with improved transient performance. We review here one of the first such designs [32], which applies to nonholonomic systems that can be transformed into chained form. The control law is continuous in time but nonsmooth with respect to the state, which is fed back only at uniformly sampled instants.

Consider the chained-form (3.9) as an equivalent unicycle model, and note that the origin of the (x, y, θ) configuration space (which is the desired posture q_d) maps into the origin of the (z_1, z_2, z_3) space. If the input u_1 is a predefined function of time, $z_{23} = [z_2 \ z_3]^T$ satisfies a linear time-varying equation driven by the input u_2. The command u_1 is obtained by combining a simple open-loop command, which is updated as a function of the state only on a discrete-time basis, with a time-varying exogenous signal, in such a way that z_1 converges exponentially to zero when $\|z_{23}\|$ does. The other input u_2 is chosen so that $\|z_{23}\|$ converges to zero with an exponential rate.

In particular, let a sequence of uniformly spaced instants $\{t_0, t_1, t_2, \ldots\}$ be defined as

$$t_h = hT \qquad T = t_{h+1} - t_h > 0.$$

Define the control u_1 as

$$u_1(t) = k(z(t_h))f(t) \qquad \text{for } t \in [t_h, t_{h+1}). \tag{6.4}$$

This input is a function of the state z at time $t = t_h$, while during the interval (t_h, t_{h+1}) it is defined in an open-loop fashion. Choosing a smooth and periodic function $f(t)$, such as

$$f(t) = \frac{1 - \cos \omega t}{2} \qquad \omega = \frac{2\pi}{T}, \tag{6.5}$$

$u_1(t)$ is guaranteed to be continuous in time. The control function $k(z(t_h))$ is given by

$$k(z(t_h)) = -\beta \left[z_1(t_h) + \text{sgn}(z_1(t_h)) \, \gamma(\|z_{23}(t_h)\|) \right], \tag{6.6}$$

where

$$\text{sgn}(z_1) = \begin{cases} 1, & \text{if } z_1 \geq 0, \\ -1, & \text{if } z_1 < 0, \end{cases}$$

and

$$\beta = \frac{1}{\int_{t_h}^{t_{h+1}} f(\tau)d\tau} = \frac{2}{T} \qquad \gamma(\|z_{23}\|) = \kappa \|z_{23}\|^{\frac{1}{2}} \quad \kappa > 0.$$

The control law for u_2 is designed based on the *backstepping* principle [30]. Assume that the variable z_2 in the third equation of the chained form

$$\dot{z}_3 = z_2 u_1(t) \tag{6.7}$$

is a 'dummy' control input z_2^d. In order to stabilise this subsystem, we choose

$$z_2^d = -\frac{\lambda_3 f^3(t) z_3}{u_1(t)} = -\frac{\lambda_3}{k(z(t_h))} f^2(t) z_3 \qquad \lambda_3 > 0. \tag{6.8}$$

It can be shown that z_3 exponentially converges to zero with a rate depending on λ_3. In the presence of a transient difference $\tilde{z}_2 = z_2 - z_2^d$, one can also prove that z_3 exponentially converges to zero if $\tilde{z}_2$ does. Therefore, u_2 should be designed so as to make z_2 converge to z_2^d, enforcing thus the desired behaviour for z_3. Since the second equation of the chained form is

$$\dot{z}_2 = u_2,$$

this is accomplished, with exponential convergence, by choosing

$$u_2 = -\lambda_2(z_2 - z_2^d) + \dot{z}_2^d \qquad \lambda_2 > 0,$$

where, from (6.8),

$$\dot{z}_2^d = -\lambda_3 \left(2f(t)\dot{f}(t)\frac{z_3}{k(z(t_h))} + f^3(t) z_2 \right),$$

in which we have used (6.4), (6.7), and the fact that $k(z(t_h))$ is constant over $[t_h, t_{h+1})$.

The complete nonsmooth time-varying controller is then

$$\begin{aligned} u_1 &= -k(z(t_h)) f(t) \\ u_2 &= -(\lambda_2 + \lambda_3 f^3(t)) z_2 - \lambda_3(\lambda_2 f^2(t) + 2f(t)\dot{f}(t))\frac{z_3}{k(z(t_h))}. \end{aligned} \tag{6.9}$$

Equation (6.9) should be used in conjunction with (3.8) in order to generate the actual velocity inputs v and ω.

Defining a *class $\mathcal{K}$ function* as a strictly increasing function $h : \mathbb{R}^+ \mapsto \mathbb{R}^+$ such that $h(0) = 0$, the main result is summarised in the following theorem.

Theorem 6.2. *Consider the unicycle in chained form (3.9) under the control (6.9), with the definitions (6.5) and (6.6). Then, the origin $z = 0$ is $\mathcal{K}$-exponentially stable, i.e. there exist a constant $\lambda_z > 0$ and a class $\mathcal{K}$ function $h_z(\cdot, T)$ such that*

$$\|z(t)\| \leq h_z(\|z(0)\|, T) \, e^{-\lambda_z t} \qquad \forall z(0) \in \mathbb{R}^3, \forall t \geq 0.$$

*Proof.* It is a specialisation of the general proof for n-dimensional chained-form systems, see [32]. ∎

Figures 6.5–6.8 show the results of the application of the control law (6.9), with $T = 6$ s and $\kappa = 0.8$, $\lambda_2 = \lambda_3 = 0.4$, for executing the baseline forward parking task. The rate of convergence of the nonsmooth time-varying controller is somewhat improved but still quite slow. A stroboscopic view of the unicycle motion sampled every 5 s is reported in Figure 6.6. Note that the approach in the y direction is very uniform, while maneuvers in the vicinity of the goal are aimed at adjusting θ rather than x. This is intrinsic in the structure of the chained form used for the control design.
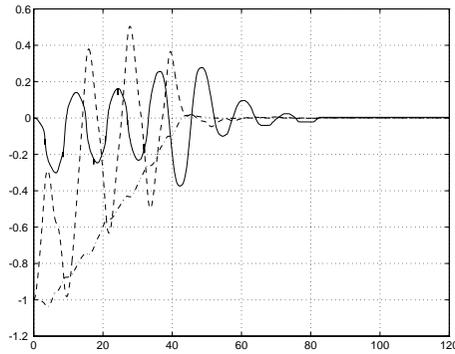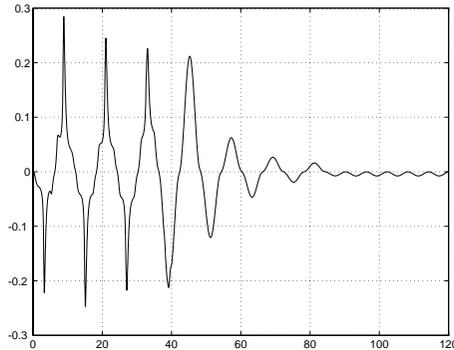
Figure 6.5. Posture stabilisation with nonsmooth time-varying feedback: x $(--)$, y $(-\cdot)$ (m), and θ $(—)$ (rad) vs. time (s)
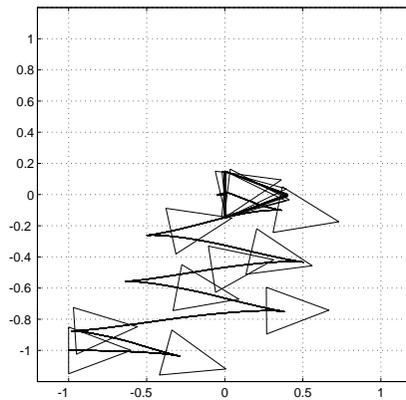
Figure 6.6. Posture stabilisation with nonsmooth time-varying feedback: Cartesian motion (x, y) (m)
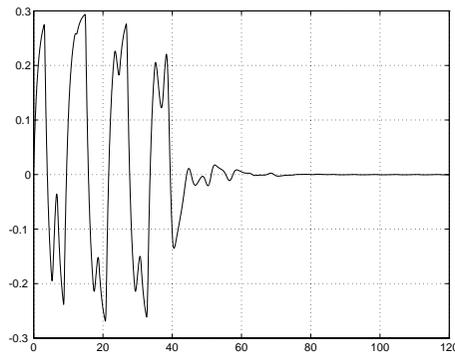
Figure 6.7. Posture stabilisation with nonsmooth time-varying feedback: driving velocity v (m/s)

Figure 6.8. Posture stabilisation with nonsmooth time-varying feedback: steering velocity ω (rad/s)

6.3 Control Based on Polar Coordinates

Another technique which allows to overcome the obstruction of Brockett theorem is to apply a change of coordinates such that the input vector fields of the transformed equations are singular at the origin. This approach has been proposed in [1], where a Lyapunov-like design of a posture stabilizing controller is carried out using a polar coordinate transformation. The control law, once rewritten in terms of the original state variables, is discontinuous at the origin of the configuration space $\mathcal{Q}$.

With reference to Figure 6.9, we define the following set of *polar* coordinates for the unicycle. Let ρ be the distance of the reference point (x, y) of the unicycle from the goal (the origin), γ be the angle of the pointing vector to the goal w.r.t. the unicycle main axis, and δ be the angle of the same pointing vector w.r.t. the x axis (orientation error), i.e.

$$
\begin{aligned}
\rho &= \sqrt{x^2 + y^2} \\
\gamma &= \text{ATAN2}(y, x) - \theta + \pi \\
\delta &= \gamma + \theta.
\end{aligned}
$$

Angles γ and δ are undefined for $x = y = 0$. In practice, during the vehicle motion, when $\rho \to 0$ one must retain the values for γ and δ assumed in the final approaching phase. In these coordinates, the unicycle equations become

$$
\begin{aligned}
\dot{\rho} &= -\cos\gamma\, v \\
\dot{\gamma} &= \frac{\sin\gamma}{\rho} v - \omega \\
\dot{\delta} &= \frac{\sin\gamma}{\rho} v.
\end{aligned}
\tag{6.10}
$$

The input vector field associated to v is singular for $\rho = 0$.
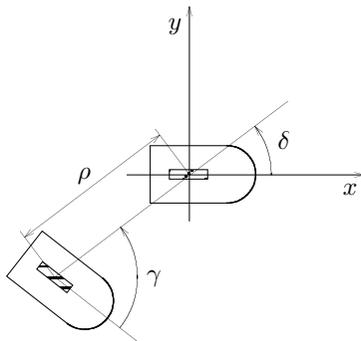
Figure 6.9. Definition of polar coordinates for the unicycle

In order to achieve the goal posture, variables e, γ, and δ should all converge to zero. This is guaranteed by the following result, adapted from [1].

Theorem 6.3. *Consider the polar coordinate description (6.10) of the unicycle and the feedback control*

$$
\begin{aligned}
v &= k_1 \rho \cos \gamma \\
\omega &= k_2 \gamma + k_1 \frac{\sin \gamma \cos \gamma}{\gamma} (\gamma + k_3 \delta),
\end{aligned}
\tag{6.11}
$$

with k_1 and k_2 positive constants. The closed-loop system (6.10)–(6.11) is then globally asymptotically driven to the posture $(\rho, \gamma, \delta) = (0, 0, 0)$.

Proof. (Sketch) Consider the function

$$
V = \frac{1}{2} \left(\rho^2 + \gamma^2 + k_3 \delta^2 \right),
$$

as a candidate Lyapunov function. Its time derivative along the solutions of the closed-loop system is nonincreasing since

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

Thus, 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 ρ and γ do. Analyzing the closed-loop system, one can show that $\dot{\rho}$ and $\dot{\delta}$ converge to zero, that δ converges to some finite limit $\bar{\delta}$ while $\dot{\gamma}$ tends to the finite limit $-k_1 k_3 \bar{\delta}$ and is uniformly continuous. Again through Barbalat lemma, this finite limit must be zero and thus also δ converges to zero.   ∎

Figures 6.10–6.13 refer to the results of the application of controller (6.11), with gains $k_1 = 1$, $k_2 = 3$ and $k_3 = 2$, for executing the baseline forward parking task. The convergence to the goal is very fast and natural. In Figure 6.11, a stroboscopic view of the unicycle motion sampled every 1 s is given. Both velocity commands are saturated for 2 to 3 seconds.

In the second experiment, a parallel parking task from $q(0) = (0, -1.5, 0)$ (m,m,rad) to the origin of the configuration space is assigned. The same gains as in the previous experiment have been used. The obtained results are shown in Figures 6.14–6.17 and indicate that there is no backup maneuver in this case. If the robot had been initially closer to the positive x axis, the controller (6.11) would have automatically driven the robot backwards and then in forward motion to the goal. This is a general property of the controller: in the final phase, the vehicle will always approach the goal in forward motion, having executed at most one backup maneuver.

We finally note that the behaviour of the controlled system is not continuous with respect to the initial state. For example, assume that the initial configuration is $x(0) = a > 0$, $y(0) = \varepsilon$ and $\theta(0) = 0$. Positive and negative arbitrarily small values of ε will lead to different transient motions to the goal (in fact, symmetric with respect to the x axis).
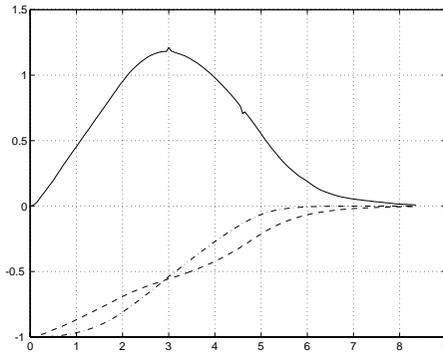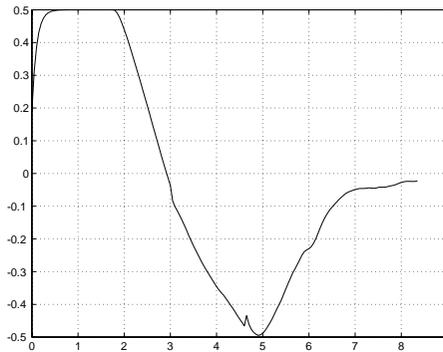

Figure 6.10. Posture stabilisation using feedback in polar coordinates (forward parking): x $(--)$, y $(-\cdot)$ (m), and θ $(—)$ (rad) vs. time (s)
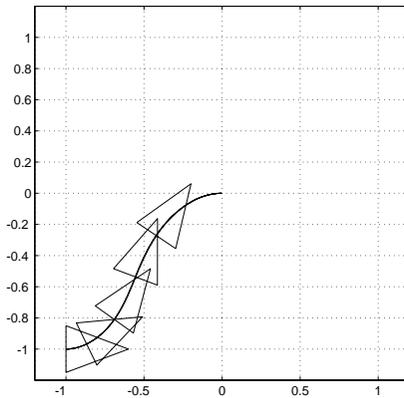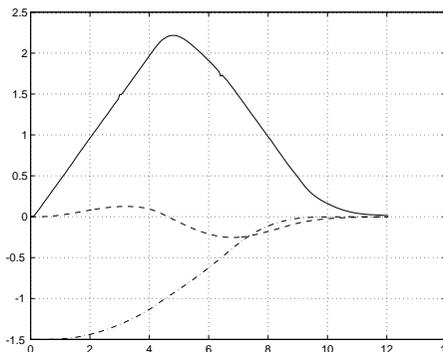

Figure 6.11. Posture stabilisation using feedback in polar coordinates (forward parking): Cartesian motion (x, y) (m)
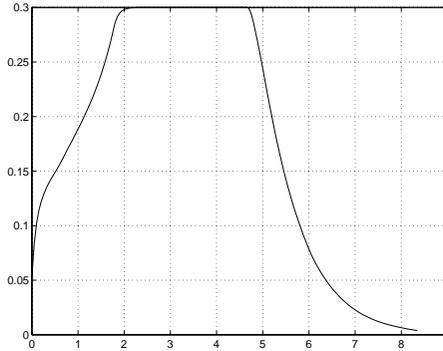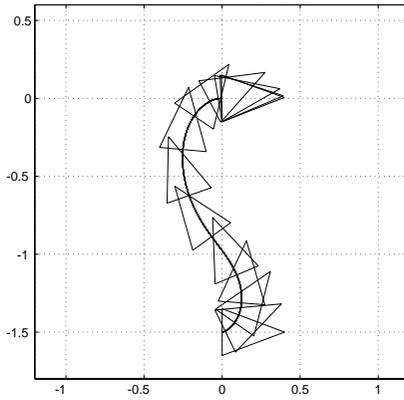
Figure 6.12. Posture stabilisation using feedback in polar coordinates (forward parking): driving velocity v (m/s)
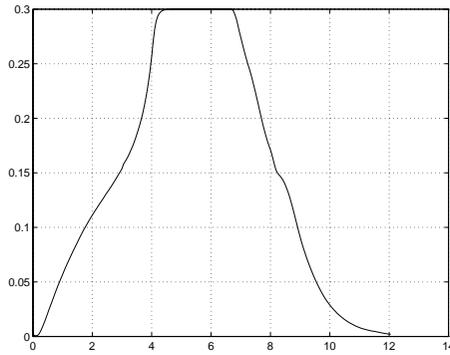

Figure 6.13. Posture stabilisation using feedback in polar coordinates (forward parking): steering velocity ω (rad/s)
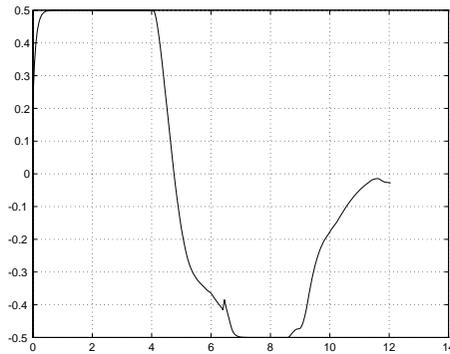

Figure 6.14. Posture stabilisation using feedback in polar coordinates (parallel parking): x $(--)$, y $(-\cdot)$ (m), and θ $(—)$ (rad) vs. time (s)

Figure 6.15. Posture stabilisation using feedback in polar coordinates (parallel parking): Cartesian motion (x, y) (m)

Figure 6.16. Posture stabilisation using feedback in polar coordinates (parallel parking): driving velocity v (m/s)

Figure 6.17. Posture stabilisation using feedback in polar coordinates (parallel parking): steering velocity ω (rad/s)

6.4 Dynamic Feedback Linearisation

Following [14], we show how to extend the trajectory tracking controller based on dynamic feedback linearisation to address the posture stabilisation problem, while avoiding the intrinsic singularity that occurs when the robot comes to a stop. This simply requires an appropriate choice of the PD gains and a suitable initialisation of the dynamic compensator state ξ.

We denote by

$$\mathcal{Q}^* = \{q \in \mathcal{Q} : (x = 0, \cos \vartheta \geq 0) \text{ OR } (y = 0, \cos \vartheta = -1)\}$$

a subset of $\mathcal{Q}$ which will require special attention. The remaining part $\mathcal{Q}/\mathcal{Q}^*$ of the configuration space can be partitioned in two regions:

$$\begin{aligned}
\mathcal{Q}^r &= \{q \in \mathcal{Q}/\mathcal{Q}^* : x \geq 0\} \\
\mathcal{Q}^l &= \{q \in \mathcal{Q}/\mathcal{Q}^* : x < 0\}.
\end{aligned}$$

The main result is the following.

Theorem 6.4. *Consider the unicycle system (3.3) under the action of the dynamic compensator (5.15). A PD control law on the Cartesian error*

$$\begin{aligned}
u_1 &= -k_{p1}x - k_{d1}\dot{x} \\
u_2 &= -k_{p2}y - k_{d2}\dot{y}
\end{aligned} \tag{6.12}$$

yields exponential convergence from any initial configuration $q(0) \in \mathcal{Q}/\mathcal{Q}^*$ to the origin, if the following assumptions hold:

A1. The control gains $k_{pi} > 0$, $k_{di} > 0$ ($i = 1, 2$) satisfy the conditions

$$k_{d1}^2 - 4k_{p1} = k_{d2}^2 - 4k_{p2} > 0 \tag{6.13}$$

$$k_{d2} - k_{d1} > 2\sqrt{k_{d2}^2 - 4k_{p2}}. \tag{6.14}$$

A2. The initial state of the dynamic compensator is chosen as

$$\begin{aligned}
\xi(0) &< 0 \quad \text{(backward motion)} \quad &\text{if } q(0) \in \mathcal{Q}^r \\
\xi(0) &> 0 \quad \text{(forward motion)} \quad &\text{if } q(0) \in \mathcal{Q}^l,
\end{aligned}$$

but its value is otherwise arbitrary, except for the additional condition

$$\xi(0) \neq 2 \frac{k_{p1}x(0)\sin\theta(0) - k_{p2}y(0)\cos\theta(0)}{k_{d2} - k_{d1}}. \tag{6.15}$$

Proof. Use of control (6.12) in (5.17) implies that the Cartesian coordinates x and y converge to zero exponentially, provided that the original control inputs v and ω given in (5.15) remain bounded. To show this, we must prove that *(i)* ξ does not go to zero in finite time, and *(ii)* ω tends to zero for $t \to \infty$, in spite of its denominator ξ vanishing.

(i) Since from (5.16) it is $\xi^2 = z_3^2 + z_4^2$, one has $\xi(\bar{t}) = 0$ iff $z_3(\bar{t}) = z_4(\bar{t}) = 0$, for a generic instant $\bar{t} \geq 0$. Integrating the closed-loop system (5.17) under control (6.12), we have

$$z_3(t) = a_{31}e^{\lambda_{11}t} + a_{32}e^{\lambda_{12}t} \qquad (6.16)$$

$$z_4(t) = a_{41}e^{\lambda_{21}t} + a_{42}e^{\lambda_{22}t}, \qquad (6.17)$$

where coefficients a_{kj} and eigenvalues λ_{ij} are easily obtained as functions of initial state and PD gains. From these expressions and condition (6.13), it is possible to show that a finite $\bar{t} > 0$ such that $\xi(\bar{t}) = 0$ exists iff

$$\frac{\xi(0)\cos\theta(0) - x(0)\lambda_{11}}{\xi(0)\cos\theta(0) - x(0)\lambda_{12}} = \alpha \frac{\xi(0)\sin\theta(0) - y(0)\lambda_{21}}{\xi(0)\sin\theta(0) - y(0)\lambda_{22}},$$

with $\alpha = \lambda_{11}\lambda_{22}/\lambda_{12}\lambda_{21}$. From this, a quadratic equation in $\xi(0)$ is derived which has the single nonzero root

$$\xi(0) = \frac{\lambda_{11}x(0)\sin\theta(0) + \lambda_{22}y(0)\cos\theta(0) - \alpha(\lambda_{12}x(0)\sin\theta(0) + \lambda_{21}y(0)\cos\theta(0))}{(1-\alpha)\sin\theta(0)\cos\theta(0)}.$$

Once rewritten in terms of the PD gains, this expression leads to the forbidden initialisation condition (6.15).

(ii) First of all, it is straightforward to prove that assumption A1 implies that the eigenvalues are real and ordered as $\lambda_{11} < \lambda_{12} < \lambda_{21} < \lambda_{22} < 0$. From (5.15), we rewrite ω as

$$\omega = \frac{-z_4 u_1 + z_3 u_2}{\xi^2},$$

and using (6.16–6.17) and (6.12), the numerator of ω takes the form

$$-z_4 u_1 + z_3 u_2 = \gamma_1 e^{(\lambda_{11}+\lambda_{21})t} + \gamma_2 e^{(\lambda_{12}+\lambda_{21})t} + \gamma_3 e^{(\lambda_{11}+\lambda_{22})t} + \gamma_4 e^{(\lambda_{12}+\lambda_{22})t},$$

with $\gamma_i \in \mathbb{R}$. Its asymptotic rate of convergence is certainly larger than $2|\lambda_{22}|$ due to the eigenvalue ordering. As for the denominator, squaring and adding (6.16) and (6.17) gives

$$\xi^2 = \eta_1 e^{2\lambda_{11}t} + \eta_2 e^{(\lambda_{11}+\lambda_{12})t} + \eta_3 e^{2\lambda_{12}t} + \eta_4 e^{2\lambda_{21}t} + \eta_5 e^{(\lambda_{21}+\lambda_{22})t} + \eta_6 e^{2\lambda_{22}t},$$

with $\eta_i \in \mathbb{R}$. Since the asymptotic rate of convergence of this quantity is exactly $2|\lambda_{22}|$, we conclude that ω tends to zero as $t \to \infty$.

To finish the proof, it is necessary to show that also the orientation θ converges to zero. This is easily understood from the following facts:

- The unicycle reaches the origin with a horizontal tangent ($\theta = 0$ or π), because y approaches zero faster than x in view of the eigenvalue ordering.
- Motion inversions do not occur since $v = \xi$ never crosses zero, as shown in the first part of the proof.
- The trajectory is confined to the region (either $\mathcal{Q}_r$ or $\mathcal{Q}_l$) from which the unicycle starts. In fact, x and y never change sign because the eigenvalues are real and thanks to the choice of sign for $\xi(0)$ in assumption A2.

Finally, also the convergence of θ to zero is exponential. Indeed, since ω goes exponentially to zero, the same is true for its integral θ. ■

Some remarks are needed at this point.

- As the Cartesian position transients are linear, the unicycle trajectories obtained with the proposed controller are completely predictable and can be easily shaped by choosing the PD control gains. Note that the unicycle can reach the goal either with a forward or with a backward motion.
- The equality part of condition (6.13) in Theorem 6.4 is by no means necessary; it is only used for deriving a closed form for the forbidden initialisation (6.15) of the dynamic compensator.
- In view of the discontinuity at the origin of the linearising controller with respect to the state (x, y, θ, ξ) of the extended system, as well as of the fact that the initial configuration should belong to $\mathcal{Q}/\mathcal{Q}^*$, the proposed feedback controller does not yield Lyapunov stability in a strict sense, but simply exponential convergence.

If the initial configuration $q(0)$ belongs to $\mathcal{Q}^*$, Theorem 6.4 cannot be applied. In fact, the PD control (6.12) would bring the unicycle to the origin with the wrong orientation, namely, $\theta = \pi$ if $\cos\theta(0) > 0$, $\theta = \pm\pi/2$ if $\cos\theta(0) = 0$, $\theta = -\pi$ if $\cos\theta(0) = -1$. In such a situation, it is necessary to reset the compensator state at some time $t_v > 0$, so as to invert the motion at a configuration $q \in \mathcal{Q}/\mathcal{Q}^*$. A simple way to obtain this is to introduce a via point q_v in the regulation procedure, as illustrated later in this section by the parallel parking experiment. This should not be seen as a drawback of the method; a suitable choice of the via point allows better control of the path shape while approaching the goal configuration. In particular, the resulting stabilisation motion contains at most one backup maneuver.

Figures 6.18–6.21 refer to the baseline forward parking task, using dynamic feedback linearisation plus PD control with gains $k_{p1} = 2$, $k_{d1} = 3$, $k_{p2} = 12$, $k_{d2} = 7$, and compensator initialisation $\xi(0) = v_{max} = 0.3$ (m/s). The convergence to the goal is fast and very natural, as shown in Figure 6.19, a stroboscopic view of the robot motion sampled every 1.5 s. Note that saturation occurs on both inputs during the transient phase.
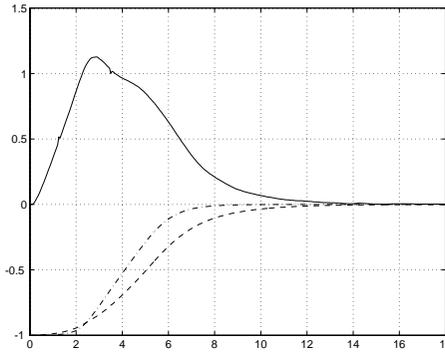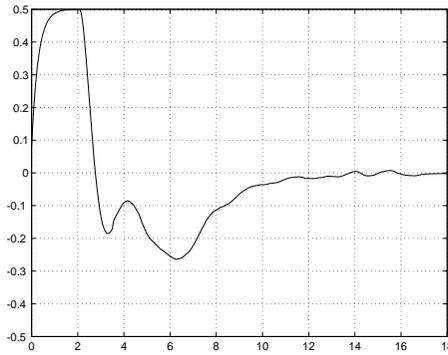
Figure 6.18. Posture stabilisation using dynamic feedback linearisation (forward parking): x $(--)$, y $(-\cdot)$ (m), and θ $(—)$ (rad) vs. time (s)
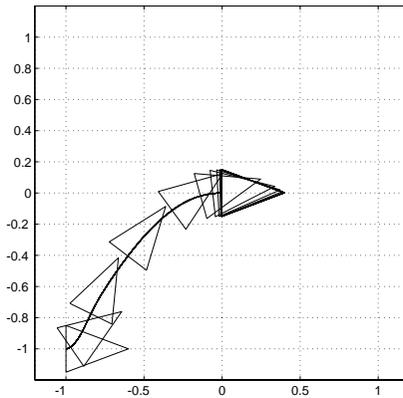
Figure 6.19. Posture stabilisation via dynamic feedback linearisation (forward parking): Cartesian motion (x, y) (m)
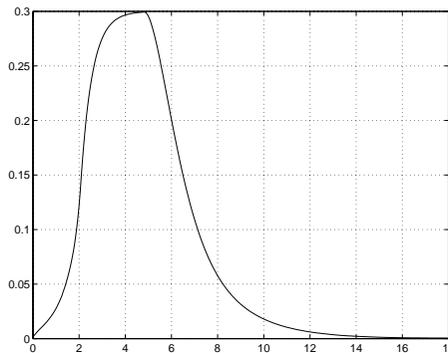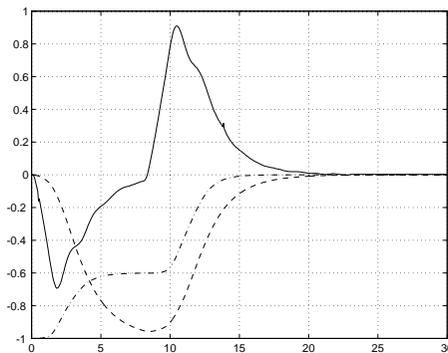

Figure 6.20. Posture stabilisation via dynamic feedback linearisation (forward parking): driving velocity v (m/s)
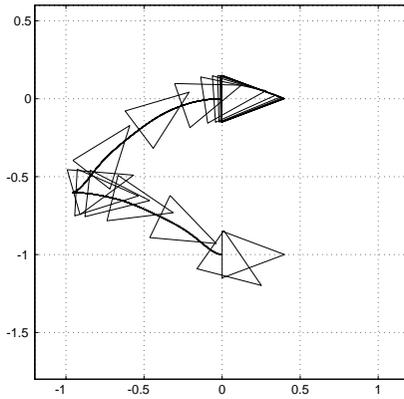
Figure 6.21. Posture stabilisation via dynamic feedback linearisation (forward parking): steering velocity ω (rad/s)

We also report the results for a parallel parking task with initial configuration given by $q(0) = (0, -1, 0)$ (m,m,rad). The via point is chosen as $q_v = (-1, -0.6, 0)$ (m,m,rad). The PD gains are the same as before, while the compensator state initialisation is chosen here as $\xi(0) = -v_{max}$ for the first phase, performed in backward motion, and $\xi(t_v) = v_{max}$ for the second phase, which is started in a neighbourhood of q_v and performed in forward motion. The results are shown in Figures 6.22–6.25.

The simple first-order linear filter introduced to account for actuator dynamics is also effective in smoothing the discontinuity in the driving velocity generated by the reset procedure. On the other hand, the presence of the same filter for the steering velocity, coupled with the software velocity saturation, neutralises the effect of the singularity in ω due to the zero crossing of the filtered driving velocity.

Figure 6.22. Posture stabilisation via dynamic feedback linearisation (parallel parking): x ($--$), y ($-\cdot$) (m), and θ ($—$) (rad) vs. time (s)

Figure 6.23. Posture stabilisation via dynamic feedback linearisation (parallel parking): Cartesian motion (x, y) (m)
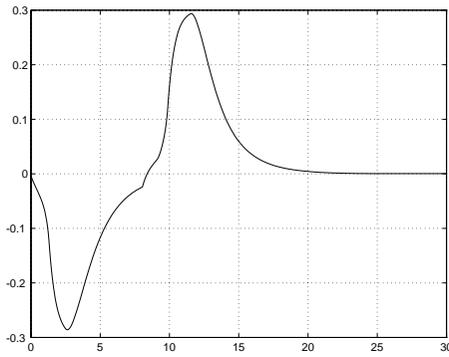
Figure 6.24. Posture stabilisation via dynamic feedback linearisation (parallel parking): driving velocity v (m/s)
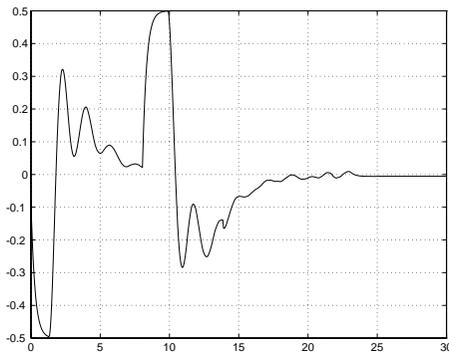
Figure 6.25. Posture stabilisation via dynamic feedback linearisation (parallel parking): steering velocity ω (rad/s)

7 Guidelines for End-users

7.1 Summary and Comparison

We have performed several motion tasks with SuperMARIO using the presented control laws. The experimental tests presented in this chapter are representative of the average performance of the controllers. We summarise hereafter our acquired experience in general observations that can be useful guidelines for implementation of the same control strategies for other vehicles.

First of all, the computational load for all methods is quite similar in the case of the unicycle. Basically, both trajectory tracking and posture stabilisation controllers can be implemented with on-board computing power. Our choice of separating high-level control routines, performed on a remote server, from the low-level control loops in charge of realizing the reference velocity commands on each wheel reflects only the choice of a modular structure. It is however expected that such decomposition would become more convenient or even mandatory for wheeled mobile robots with more complex kinematics, such as a tractor vehicle towing a number of trailers.

The three reviewed control methods for trajectory following tasks show similar performance. All of them can be generalised to more complex vehicles, provided their models are transformable in chained-form. Such generalisations can be found in [27] and [13]. From the point of view of control parameters tuning, especially for more complex WMRs, the dynamic feedback linearisation technique promises to be simpler since it always boils down to the choice of stabilizing gains for a chain of integrators [13]; in particular, it can be carried out on the original equations without resorting to the transformation in chained form.

In Table 7.1, the control results for posture stabilisation tasks are compared in terms of performance, ease of control parameters tuning, sensitivity to nonidealities, generalisability to more complex nonholonomic WMRs, and relations with the design of tracking controllers.

Time-varying controllers, both smooth and nonsmooth, exhibit a rather slow final convergence to the goal in spite of substantial progress during the first motion phase. In general, the nonsmooth controller should behave better because it achieves an exponential rate of convergence, but the dependence of this rate on the available gains is critical. The oscillatory behaviour of the vehicle during the approach to the goal, which makes the motion rather erratic, is an intrinsic characteristic of both time-varying control laws. In fact, the exogenous time dependence needed for stabilisation is introduced through the oscillatory motion of a virtual reference vehicle in the method of Section 6.1 and through the periodic time function weighting the first command in the method of Section 6.2. In any case, the presence of several motion inversions makes these methods quite sensitive to mechanical nonidealities (e.g. backlash) of the wheels. This situation may introduce a remarkable difference

	Smooth time-varying stabilisation	Nonsmooth time-varying stabilisation	Design with polar coordinates	Dynamic feedback linearisation
Achieved performance	very slow erratic	slow erratic	fast natural	fast natural
Ease of control tuning	a few parameters problematic	a few parameters critical	simple	simple PD ξ initialisation
Sensitivity to nonidealities	many backups	backups and sampled feedback	good	good integral action
Generalisation to other WMRs	yes if chained form	yes if chained form	no	yes if chained form
Relation with tracking control	extended from tracking	none	none	same PD control law

Table 7.1. A comparison of the posture stabilisation controllers presented in this chapter

between computed configuration from the internal odometry and actual displacement of the vehicle on the ground. In our experience, this behaviour was confirmed also in experiments performed with a car-like vehicle (the MARIO robot [25]), where a nonnegligible backlash on the steering angle of the front wheels led to a substantial error in the final positioning. Another potential problem with the presented nonsmooth controller is that, being based on a low-rate sampled state feedback (see (6.4)), the robot could in principle 'miss' the final goal even if passing through it. Among the positive features of these time-varying control laws, we mention that they can readily be generalised to more complex WMRs allowing a chained-form representation (see [27], [32]). Also, the smooth time-varying controller is a direct outgrowth of the trajectory tracking controller of Section 5.3.

The controller based on polar coordinates transformation performed very well. The resulting vehicle path is very natural (in the sense that is similar to the one followed by an experienced human driver) and practical convergence is quite fast, with a weak dependence on the choice of the few control gain parameters. Since at most one backup maneuver is needed, disturbances due to wheel backlash are minimised. Unfortunately, a direct extension of such controller is not available for vehicles with more complex kinematics. The idea of using a state-space transformation that is singular at the goal configuration, however, stands on its own and has been exploited by other researchers, e.g. in [3]. Also, the polar coordinates controller has been adapted

in [1] for stabilisation about successive via points extracted from a desired path, but in this way only approximate trajectory tracking can be obtained.

Similar positive comments can be drawn on the performance of the posture stabilisation method designed via dynamic feedback linearisation. In particular, this scheme allows also parallel parking with backward/forward motion, which is a very natural maneuver. The control tuning requires the choice from a very large feasible set of PD gains. The relationship with the analogous controller for trajectory tracking is very simple: it is sufficient to add the feedforward terms, i.e. the reference output position, velocity and acceleration (compare (5.18) with (6.12)). As for the use of an additional dynamics in the control law, it has some pros and cons. On one side, this design automatically takes into account the nonideality of a first-order kinematic model of the unicycle, by bringing linear acceleration into the picture. On the other side, it is necessary to prevent zeroing of the compensator state and the consequent singularity of the control commands; this may be achieved in practice by the simple strategy of filtering plus saturating the velocity commands. The generalisation to point-to-point motion tasks for WMRs with more complex kinematics is under way. It basically consists in extending the idea of suitably shaping the transient behaviour on the linear side of the problem by appropriate selection of the gain structure (a PD^{n-2} for n generalised coordinates), so as to achieve a smooth and correct 'entrance' into the goal for the two outputs representing the robot Cartesian position.

All the controllers reviewed in this chapter use a measure of the state reconstructed on the basis of the robot odometry. In principle, the actual motion of SuperMARIO on the ground may be quite different and should be computed with the aid of exteroceptive sensors. However, in our experiment, this difference could not be appreciated visually (as shown by the videos on the web page `http://labrob.ing.uniroma1.it/projects/ramsete.html`). The satisfactory performance of our dead reckoning localisation system is of course related to the execution of relatively slow motion tasks.

A final remark is needed about the application of the control methods reviewed in this chapter when workspace obstacles are present. In a completely known environment, it may be convenient to tackle the navigation problem of a WMR using a three-layer control structure. The highest layer is devoted to motion planning and takes care of the nominal avoidance of obstacles; the nonholonomic motion constraints of the WMR may or may not be taken into account at this stage. The second layer takes charge of motion execution and uses one of the trajectory tracking controllers given in this chapter. In the vicinity of the goal, fine posture regulation (docking) can be obtained at the lowest layer by means of one of the presented stabilizing controllers.

7.2 Future Directions

There are some important issues barely mentioned in this chapter that deserve further research.

Inclusion of dynamics. For massive vehicles and/or at high speeds, consideration of robot dynamics is necessary for realistic control design. The dynamics of general nonholonomic systems is thoroughly analysed in [5] and, more specifically for WMRs, in [7]. Interestingly, nonlinear static state feedback can be used to cancel, in the nominal case, all inertial parameters so as to lead to a purely (second-order) kinematic model of the form

$$\dot{q} = G(q)w, \qquad \dot{w} = a,$$

in place of (3.2), with (q, w) as the $(n+m)$-dimensional state and acceleration a as the new control input. The control laws used in this chapter do not directly apply to this case (they may have finite jumps in the velocity), but it is relatively easy to rework feasible modifications.

Robust control design. Very few papers have explicitly addressed robustness issues in the control of nonholonomic systems. Robust stabilisation of WMRs in chained form was obtained in [4] and [19] by applying iteratively an open-loop command; exponential convergence to the desired equilibrium is obtained for small perturbations in the kinematic model. Another possible approach to the design of effective control laws in the presence of nonidealities and uncertainties is represented by learning control, as shown in [25]. We also note that perturbations acting on nonholonomic mobile robots are not of equal importance: a deviation in a direction compatible with the vehicle mobility (e.g. slippage of the wheels on the ground) is clearly not as severe as a deviation which violates the kinematic constraints of the system (e.g. lateral sliding).

Use of exteroceptive feedback. Proprioceptive sensors, such as incremental encoders, are obviously unreliable in the presence of wheel slippage. As a result, the robot may progressively 'lose' itself in the environment. A possible solution is to close the feedback loop using exteroceptive sensors, which provide absolute information about the robot localisation in its workspace; sensor fusion techniques are relevant at this stage. The design of sensor-level controllers for nonholonomic robots is at the beginning stage but growing fast. An example of an on-board visual servoing for trajectory tracking is presented in [20].

Extension to WMRs not transformable in chained form. Among the open problems in motion control of general WMRs, we mention the case of multibody vehicles that do not admit a transformation in chained form, such as a unicycle or car-like tractor with two or more trailers hooked at a non-zero distance from the axle of the previous moving body. A possible approach to posture stabilisation, using iterative steering of a nilpotent approximation model, can be found in [33].

Acknowledgements

References

1. Aicardi M, Casalino G, Bicchi A, Balestrino A 1995 Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robotics and Automation Magazine* 2(1):27–35
2. Alexander J C, Maddocks J H 1989 On the kinematics of wheeled mobile robots. *International Journal of Robotics Research* 8(5):15–27
3. Astolfi A 1996 Discontinuous control of nonholonomic systems. *Systems and Control Letters* 27:37–45
4. Bennani M K, Rouchon P 1995 Robust stabilization of flat and chained systems. In: *Proceedings of 3rd European Control Conference* Rome, I, pp 2642–2646
5. Bloch A M, Reyhanoglu M, McClamroch N H 1992 Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control* 37:1746–1757
6. Brockett R W 1983 Asymptotic stability and feedback stabilization. In: Brockett R W, Millman R S, Sussmann H J (Eds) *Differential Geometric Control Theory* Birkhäuser, Boston, MA, pp 181–191
7. Campion G, Bastin G, d'Andréa-Novel B 1996 Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation* 12:47–62
8. Campion G, d'Andréa-Novel B, Bastin G 1991 Modeling and state feedback control of nonholonomic mechanical systems. In: *Proceedings of 30th IEEE Conference on Decision and Control* Brighton, UK, pp 1184–1189
9. Canudas de Wit C, Khennouf H, Samson C, Sørdalen O J 1993 Nonlinear control design for mobile robots. In: Zheng Y F (Ed) *Recent Trends in Mobile Robots* World Scientific Publisher, pp 121–156
10. Canudas de Wit C, Sørdalen O J 1992 Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control* 37:1791–1797
11. d'Andréa-Novel B, Bastin G, Campion G 1995 Control of nonholonomic wheeled mobile robots by state feedback linearization. *International Journal of Robotics Research* 14:543–559
12. De Luca A, Di Benedetto M D 1993 Control of nonholonomic systems via dynamic compensation. *Kybernetica* 29:593–608
13. De Luca A, Oriolo G, Samson C 1998 Feedback control of a nonholonomic car-like robot. In: Laumond J P (Ed) *Robot Motion Planning and Control* Springer, London, UK, pp 171–253

14. De Luca A, Oriolo G, Vendittelli M 2000 Stabilization of the unicycle via dynamic feedback linearization. In: *Preprints of 6th IFAC Symposium on Robot Control* Vienna, A, pp 397–402

15. Fliess M, Lévine J, Martin P, Rouchon P 1995 Design of trajectory stabilizing feedback for driftless flat systems. In: *Proceedings of 3rd European Control Conference* Rome, I, pp 1882–1887

16. Isidori A 1995 *Nonlinear Control Systems* (3rd ed) Springer, London, UK

17. Jiang Z-P, Nijmeijer H 1999 A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Transactions on Automatic Control* 44:265–279

18. Jones J L, Flynn A M 1993 *Mobile Robots: Inspiration to Implementation* A K Peters, Wellesley, MA

19. Lucibello P, Oriolo G 2001 Robust stabilization via iterative state steering with an application to chained-form systems. *Automatica* 37:71–79

20. Ma Y, Kočeckà J, Sastry S 1999 Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation* 15:521–536

21. M'Closkey R T, Murray R M 1997 Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. *IEEE Transactions on Automatic Control* 42:614–628

22. Morin P, Samson C 1997 Application of backstepping techniques to time-varying exponential stabilisation of chained systems. *European Journal of Control* 3:15–36

23. Murray R M 1993 Control of nonholonomic systems using chained forms. *Fields Institute Communications* 1:219–245

24. Neimark J I, Fufaev F A 1972 *Dynamics of Nonholonomic Systems* American Mathematical Society, Providence, RI

25. Oriolo G, Panzieri S, Ulivi G 1998 An iterative learning controller for nonholonomic robots. *International Journal of Robotics Research* 17:954–970

26. Samson C 1993 Time-varying feedback stabilization of car-like wheeled mobile robots. *International Journal of Robotics Research* 12:55–64

27. Samson C 1995 Control of chained systems. Application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control* 40:64–77

28. Samson C, Ait-Abderrahim K 1991 Feedback control of a nonholonomic wheeled cart in cartesian space. In: *Proceedings of 1991 IEEE International Conference on Robotics and Automation* Sacramento, CA, pp 1136–1141

29. Schraft R D, Schmierer G 1998 *Serviceroboter* Springer, Berlin, D

30. Sepulchre R, Janković M, Kokotović P 1997 *Constructive Nonlinear Control* Springer, London, UK

31. Sontag E D 1990 Feedback stabilization of nonlinear systems. In: Kaashoek M A, van Schuppen J H, Ran A C M (Eds) *Robust Control of Linear Systems and Nonlinear Control* Birkhäuser, Cambridge, MA, pp 61–81

32. Sørdalen O J, Egeland O 1995 Exponential stabilization of nonholonomic chained systems. *IEEE Transactions on Automatic Control* 40:35–49

33. Vendittelli M, Oriolo G 2000 Stabilization of the general two-trailer system. In: *Proceedings of 2000 IEEE International Conference on Robotics and Automation* San Francisco, CA, pp 1817–1823