

Tracking of an omnidirectional target with a nonholonomic mobile robot*

Guillaume Artus Pascal Morin Claude Samson

INRIA - ICARE Project

2004, route des Lucioles

06902 Sophia Antipolis Cedex, France.

E-mail: First-name.Last-name@sophia.inria.fr

Abstract

A control strategy for tracking an omnidirectional target with a unicycle-like robot is proposed. An originality of the approach is that the target is allowed to move freely in the plane and perform motions which are not feasible by the nonholonomic robot. Control implementation involves the design of an estimator of the target velocity, based on visual and odometrical measurements. Finally, simulation and experimental results are reported.

1 Introduction

We consider the problem of tracking a vehicle — called *target* from now on— with a nonholonomic unicycle-like robot by using vision data acquired with an on-board camera. A distinctive feature of the present study is that no assumption is made on the target's motion so that, due to nonholonomic constraints on the robot which forbid instantaneous lateral motion, perfect tracking of the target is generically impossible. A practical situation which illustrates this problem arises when one is interested in making a robotic car automatically follow another car. This corresponds to a typical car-platooning application except that tracking has to be continued when the leader performs maneuvers that involve changes in the sign of its longitudinal velocity. To our knowledge, this type of problem has seldom been addressed in the literature. As a matter of fact, finding an adequate formulation of the control problem is not even straightforward in this case. Indeed, while following a leading vehicle with positive longitudinal velocity can be solved with classical control techniques, and is well documented in the robotics/automotive literature, what does tracking mean when the leader moves backward? A possible scenario consists in imagining a virtual frame attached behind the vehicle. The problem at hand then basically amounts to controlling the posture error between this frame and the robotic car's body. Zeroing

this error all the time would correspond to perfect tracking. Stated in these terms, the control problem looks alike *trajectory tracking*, another much studied problem [5, 1]. There is however an important difference. In the trajectory tracking case, it is assumed that the reference trajectory is feasible, i.e. compatible with the kinematics of the controlled vehicle. In our case, this assumption does not hold because the velocity of the virtual frame has a lateral component which vanishes only when the vehicle moves along a straight line (as it is simple to verify). Since nonholonomy forbids such a lateral motion for the robotic car, perfect tracking of the virtual frame is usually not possible. Instead, some type of practical stabilization yielding, for instance, uniform ultimate boundedness of the tracking errors (in both position and orientation) has to be considered. The problem addressed in the present paper is a generalization, adapted to the case of a unicycle-type mobile robot, of the control problem evoked in the above example.

The control strategy here considered is based on the transverse function control approach [4] which provides a general framework for the design of control laws yielding practical stabilization for nonlinear controllable driftless systems submitted to additive perturbations. It is closely related to, and may be seen as an extension of, a control proposed in [2], where practical stabilization of feasible trajectories for a unicycle-type robot is studied. Estimation of the target's velocity is understandably useful for tracking purposes and improves the performance of the control scheme. The design of such an estimator, based on visual data —from which the relative configuration of the robot with respect to the target is reconstructed— and odometrical measurements, is described in the paper. This is combined with the problem of filtering measurement noise on the reconstructed target/robot configuration.

The paper is organized as follows. Control models are introduced in Section 2. The control strategy

*This work is supported by the European project CyberCars

is presented in Section 3. The target velocity estimator is presented in Section 4. Since mobile robots equations are nonlinear, the superposition principle of linear control theory does not apply. For this reason, a complementary analysis is needed to prove the stability of the proposed controller/estimator. It is conducted in Section 5. Various implementation issues are addressed in Section 6. Simulation and experimental results are reported in Sections 7 and 8 respectively. Due to space limitations, proofs are omitted.

2 Modeling

Let us consider the three frames represented on Fig. 1: \mathcal{F}_0 is a fixed frame, \mathcal{F} is a frame attached to the unicycle, and \mathcal{F}_r is a frame attached to the target. Let (x_m, y_m) denote the coordinates of \overrightarrow{OP} in \mathcal{F}_0 , and α_m denote the angle between \vec{i}_0 and \vec{i} (see Fig. 1). The control inputs of the robot are the longitudinal velocity u_1 along the vector \vec{i} of \mathcal{F} and the angular velocity $u_2 = \dot{\alpha}_m$. With these notations, the well known kinematic equations of the unicycle are

$$\begin{cases} \dot{x}_m = u_1 \cos \alpha_m \\ \dot{y}_m = u_1 \sin \alpha_m \\ \dot{\alpha}_m = u_2 \end{cases} \quad (1)$$

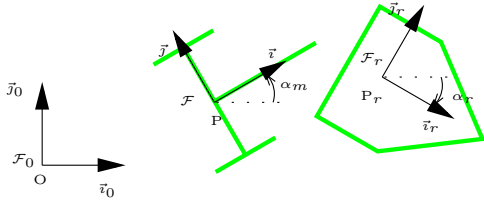


Figure 1: Configuration frames

For stabilization purposes, we want to describe the kinematics of the robot with respect to the target. Let (x_r, y_r) denote the coordinates of \overrightarrow{OP}_r in \mathcal{F}_r , and α_r denote the angle between \vec{i}_0 and \vec{i}_r . The two components of the velocity of P_r , expressed in the basis of the target frame, are denoted as a and b , and the target angular velocity is denoted as c , i.e.

$$\begin{cases} \frac{d\overrightarrow{OP}_r}{dt} = a\vec{i}_r + b\vec{j}_r \\ \dot{\alpha}_r = c \end{cases} \quad (2)$$

We denote as $u_t = (a, b, c)^T$ the target velocity vector. If x and y are the coordinates of the position error vector between the robot and the target, expressed in the target frame \mathcal{F}_r , i.e. $\overrightarrow{P_rP} = x\vec{i}_r + y\vec{j}_r$, and $\alpha = \alpha_m - \alpha_r$, one infers from (1) and (2)

$$\begin{cases} \dot{x} = u_1 \cos \alpha + cy - a \\ \dot{y} = u_1 \sin \alpha - cx - b \\ \dot{\alpha} = u_2 - c \end{cases} \quad (3)$$

System (3) represents the error system associated with the target tracking problem. It can be rewritten as

$$\dot{g} = u_1 b_1(g) + u_2 b_2 + b_0(g, u_t) \quad (4)$$

with $g = (x, y, \alpha)^T$ and

$$b_0(g, u_t) = (cy - a, -cx - b, -c)^T \quad (5)$$

$$b_1(g) = (\cos \alpha, \sin \alpha, 0)^T, \quad b_2 = (0, 0, 1)^T \quad (6)$$

3 Control strategy

It is based on the concept of *transverse functions* developed in [4]. We only present here the elements of the approach which are needed for the considered application. Let us temporarily focus on System (4) without the drift term b_0 , i.e.

$$\dot{g} = u_1 b_1(g) + u_2 b_2 \quad (7)$$

By definition, a bounded function $f : \mathbb{T} \rightarrow G$ — with $\mathbb{T} \triangleq \mathbb{R}/2\pi\mathbb{Z}$ — is called a *transverse function* for System (7) if, for any $\theta \in \mathbb{T}$, the matrix

$$H(\theta) \triangleq \begin{pmatrix} b_1(f(\theta)) & b_2 & -\frac{\partial f}{\partial \theta}(\theta) \end{pmatrix} \quad (8)$$

is invertible. It is not difficult to show that for any $\varepsilon_1 > 0$ and any $\varepsilon_2 \in (0, \frac{\pi}{2})$ the function f defined by

$$f(\theta) = \begin{pmatrix} \varepsilon_1 \sin \theta \\ \frac{\varepsilon_1 \varepsilon_2}{4} \sin 2\theta \\ \varepsilon_2 \cos \theta \end{pmatrix} \quad (9)$$

satisfies this condition. Such a transverse function allows one to use $\dot{\theta}$ as a new —virtual— control input for the system. In order to be more specific about this point, let us first recall a few basic definitions.

A Lie group G is a differentiable manifold equipped with a smooth group operation. The one here considered is $G = \mathbb{R}^2 \times \mathbb{T}$ endowed with the group operation $(g, g') \mapsto gg'$ defined by

$$gg' = \begin{pmatrix} p + R(\alpha)p' \\ \alpha + \alpha' \end{pmatrix} \quad (10)$$

with $g = \begin{pmatrix} p \\ \alpha \end{pmatrix}$, $g' = \begin{pmatrix} p' \\ \alpha' \end{pmatrix}$, p and $p' \in \mathbb{R}^2$, and $R(\alpha)$ the rotation matrix of angle α . We denote by $e = (0, 0, 0)^T$ the unit element of the group.

It is well known, and straightforward to verify, that System (7) defines a left-invariant control system on G , i.e., for any solution $g(\cdot)$ of (7) and any $g_0 \in G$,

$$\frac{d}{dt}(g_0 g(t)) = u_1 b_1(g_0 g(t)) + u_2 b_2$$

The following proposition provides the expression of a —dynamic— feedback law which ensures practical stability of the origin of System (4). In this proposition, d stands for the differential, r_h and l_h denote

the right and left translations by h respectively, i.e. $r_h(g) = gh$ and $l_h(g) = hg$.

Proposition 1 *Let $\mathcal{U} \subset G$ denote a neighborhood of e , and $f : \mathbb{T} \rightarrow \mathcal{U}$ a transverse function. Let $\bar{u} \triangleq (u_1, u_2, \dot{\theta})^T$. Consider the following dynamic feedback law*

$$\bar{u} = -H(\theta)^{-1} (dl_z(g) b_0(g, u_t) + dr_g(z) Kz) \quad (11)$$

with

$$dl_z(g) = \begin{pmatrix} R(\varepsilon_2 \cos \theta - \alpha) & 0 \\ 0 & 1 \end{pmatrix}$$

$$dr_g(z) = \begin{pmatrix} I_2 & R(\varepsilon_2 \cos \theta - \alpha) \begin{pmatrix} -y \\ x \end{pmatrix} \\ 0 & 1 \end{pmatrix}$$

$$z \triangleq f(\theta)g^{-1} = f(\theta) - \begin{pmatrix} R(\varepsilon_2 \cos \theta - \alpha) \begin{pmatrix} x \\ y \end{pmatrix} \\ \alpha \end{pmatrix}$$

Applying this control to (4) yields $\dot{z} = Kz$. Therefore, if K is a Hurwitz stable matrix, \bar{u} ensures the practical stabilization of $g = e$ in the sense that the set $f(\mathbb{T})$ is exponentially stable for the closed-loop system (4)-(11). In particular, if $K = -\text{diag}\{k_1, k_2, k_3\}$ with $k_{1,2,3} > 0$, any solution $g(\cdot)$ of (4) such that $|\alpha(0) - f_3(\theta(0))| < \pi$ converges exponentially to $f(\mathbb{T})$.

Since $f(\theta)$, as given by (9), tends to zero when the parameters ε_1 and ε_2 tend to zero, $\|g(t)\|$ is ultimately bounded by a number which can be rendered as small as desired by choosing these parameters small enough (but different from zero). Moreover, this bound is independent of the target's motion. Note, however, that a small value for this bound may not be necessary, nor desirable, in practice.

4 Estimator design

The knowledge of the target velocity u_t is needed to implement the feedback law (11). Since this velocity is not directly measured, an estimator is built from measurements of the tracking error $g(t)$. These measurements are themselves obtained from visual data via standard geometric calculations which will not be detailed here. The estimator is designed to also provide filtered values of the measurements of $g(t)$. This is important for our experiments because the data delivered by the on-board camera happens to be quite noisy.

In the first place, a model of the target dynamics has to be chosen. When the target velocity, with respect to the fixed frame, is constant we infer from (2) that

$$\begin{cases} \dot{a} &= cb \\ \dot{b} &= -ca \\ \dot{c} &= 0 \end{cases}$$

In order to account for possible variations of this velocity we will use the following model

$$\begin{cases} \dot{a} &= cb + v_a \\ \dot{b} &= -ca + v_b \\ \dot{c} &= v_c \end{cases} \quad (12)$$

where v_a , v_b , and v_c are bounded acceleration inputs.

The robot velocities u_1 and u_2 are assumed to be measured via odometry. Note that, for estimation purposes, odometrical measurements are usually preferred to *desired* control values, as specified by the control law (11), because they often describe the reality more accurately. However, odometrical information is slightly corrupted by noise and it is further degraded by signal processing induced delays. Undetected slippage of the actuating wheels may also account for episodic discrepancies between the odometrical measurements and the actual velocity of the robot body. Let u_1^o and u_2^o denote these measurements. The above considerations lead us to rewrite System (3) as

$$\begin{cases} \dot{x} &= (u_1^o + v_1) \cos \alpha + cy - a \\ \dot{y} &= (u_1^o + v_1) \sin \alpha - cx - b \\ \dot{\alpha} &= (u_2^o + v_2) - c \end{cases} \quad (13)$$

where v_1 and v_2 can be interpreted as small bounded perturbations associated with odometrical measurements.

Measurements of $p = (x, y)^T$ and α , obtained from visual data, are denoted as p^v and α^v . Due to the imperfection of these measurements we have

$$p^v = p + v_p = p + (v_x, v_y)^T \quad (14a)$$

$$\alpha^v = \alpha + v_\alpha \quad (14b)$$

where v_x , v_y , and v_α represent bounded measurement errors. In the forthcoming analysis, v_{max} denotes the maximum amplitude over all the terms v_{index} that we have introduced.

Now, the six equations in the models (12) and (13) can be regrouped and rewritten as

$$\dot{X}_p = A_p(c)X_p + U_p(\alpha) + V_p \quad (15a)$$

$$\dot{X}_\alpha = A_\alpha X_\alpha + U_\alpha + V_\alpha \quad (15b)$$

with

$$X_p = (x, y, a, b)^T = (p^T, a, b)^T, \quad X_\alpha = (\alpha, c)^T$$

$$U_p = (u_1^o \cos \alpha, u_1^o \sin \alpha, 0, 0)^T, \quad U_\alpha = (u_2^o, 0)^T$$

$$V_p = (v_1 \cos \alpha, v_1 \sin \alpha, v_a, v_b), \quad V_\alpha = (v_2, v_c)^T$$

$$A_p(c) = \begin{pmatrix} 0 & c & -1 & 0 \\ -c & 0 & 0 & -1 \\ 0 & 0 & 0 & c \\ 0 & 0 & -c & 0 \end{pmatrix}, \quad A_\alpha = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}$$

The linear dynamics of X_α , and the nonlinear ones of X_p , appear clearly in (15). We propose an estimator

in the following form:

$$\dot{\hat{X}}_p = A_p(\hat{c})\hat{X}_p + U_p(\hat{a}) + K_p(p^v - \hat{p}) \quad (16a)$$

$$\dot{\hat{X}}_\alpha = A_\alpha\hat{X}_\alpha + U_\alpha + K_\alpha(\alpha^v - \hat{\alpha}) \quad (16b)$$

where K_p and K_α are matrix-valued gains, chosen in accordance with the stability analysis summarized in the following proposition.

Proposition 2 *Let*

$$K_p = \begin{pmatrix} \lambda_p & 0 \\ 0 & \lambda_p \\ -\beta_p & 0 \\ 0 & -\beta_p \end{pmatrix}, K_\alpha = \begin{pmatrix} \lambda_\alpha \\ -\beta_\alpha \end{pmatrix} \quad (17)$$

with $\lambda_p, \beta_p, \lambda_\alpha, \beta_\alpha > 0$, and denote the estimation error by $\tilde{X} \triangleq (X_p - \hat{X}_p, X_\alpha - \hat{X}_\alpha)$. If the robot velocities u_1, u_2 , and X_p are bounded functions of time, then $\|\tilde{X}(t)\| \leq c_1 v_{max} + c_2 e^{-\gamma t} \|\tilde{X}(0)\|$ for some constants $c_1, c_2, \gamma > 0$.

A way to further specify the estimator's gains in (17) consists in assuming that all terms of the form v_{index} are uncorrelated zero-mean white noises with given covariances and calculating the corresponding steady-state optimal Kalman gains. Let us denote by $\omega_{index} = E[v_{index}^2]$ the covariance of v_{index} . The linear subsystem (15b) associated with the orientation variable X_α poses no difficulty for the calculation of the optimal gain K_α (see, for instance, [3, Sec. 4.4], for an exposition of the Kalman filter). As for Subsystem (15a), a difficulty arises from the nonlinearity of this system. It is circumvented by assuming that c is constant in $A_p(c)$, and $v_1 = 0$ in V_p . One obtains after calculations

Lemma 1 *If $\omega_2 \ll \sqrt{\omega_c \omega_\alpha}$, the steady-state optimal Kalman gain K_α associated with (14b)-(15b) is given by (17) with*

$$\lambda_\alpha \approx \sqrt{2}(\omega_c/\omega_\alpha)^{\frac{1}{4}}, \quad \beta_\alpha \approx (\omega_c/\omega_\alpha)^{\frac{1}{2}} \quad (18)$$

Similarly, the steady-state optimal Kalman gain K_p associated with (14a)-(15a), when c is constant, $\omega_x = \omega_y, \omega_a = \omega_b$, and $\omega_1 = 0$, is given by (17) with

$$\lambda_p \approx \sqrt{2}(\omega_a/\omega_x)^{\frac{1}{4}}, \quad \beta_p \approx (\omega_a/\omega_x)^{\frac{1}{2}} \quad (19)$$

5 Controller/estimator stability

The control (11) of Section 3 is a function of g, u_t , and θ , i.e. $\bar{u} = \bar{u}(g, u_t, \theta)$. The estimator (16) of Section 4 provides estimated values \hat{g} and \hat{u}_t for g and u_t . Since visual measurements g^v of g are also available, one can use either \hat{g} or g^v , or a suitable combination of both, in the control calculation. In particular, whenever $|\hat{g}_i - g_i^v| \geq 2v_{max}$, g_i^v provides

an estimation of g_i which is better than \hat{g}_i . The following proposition establishes the joint stability of the controller/estimator.

Proposition 3 *Consider the control law (11) with K diagonal negative and the estimator (16). Assume that the target velocity u_t is bounded, and that the feedback $\bar{u}(g^e, \hat{u}_t, \theta)$ is applied to System (4), with g^e equal to either \hat{g} or g^v , or any other estimation of g such that $\|g^e - g\| \leq \gamma v_{max}$ for some constant γ . Then, if v_{max} is small enough, the closed-loop system is stable and $\lim_{t \rightarrow \infty} (\|z(t)\| + \|\tilde{X}(t)\|) \leq C v_{max}$ for some constant C .*

In practice, the choice $g^e = g^v$ will typically be preferred to $g^e = \hat{g}$ when abrupt variations of the target velocity, or slippage of the actuating wheels, prevent the estimator (16) from performing well. The opposite choice will normally provide a better estimation for g when the target velocity is known to be constant, or almost constant. This choice will therefore often be based on extra information about the nature and amplitude of the perturbation terms v_{index} . This information may either be available in advance or gathered online by using other sensors.

6 Implementation issues

In this section, T denotes the sampling period (corresponding to the video rate of the camera, for instance) and $X[k] = X(kT)$ is the value of X at time kT ($k \in \mathbb{N}$). The control vector is periodically updated and kept constant between two sampling time instants.

Estimator: The estimator (16) is implemented in discrete time via a two-steps prediction/correction computation. With \hat{c} and \hat{a} kept constant on $[kT, (k+1)T)$, System (16) is linear on this interval. Analytic integration of this linear system yields the following prediction of $X[k+1]$ just before using the visual measurements at time $(k+1)T$:

$$\begin{aligned} \hat{X}_p^- [k+1] &= \begin{pmatrix} R_c & -TR_c \\ 0 & R_c \end{pmatrix} \hat{X}_p [k] + \begin{pmatrix} R_\alpha \Delta p^o [k] \\ 0 \end{pmatrix} \\ \hat{X}_\alpha^- [k+1] &= \begin{pmatrix} 1 & -T \\ 0 & 1 \end{pmatrix} \hat{X}_\alpha [k] + \begin{pmatrix} \Delta \alpha^o [k] \\ 0 \end{pmatrix} \end{aligned}$$

with R_c the rotation matrix of angle $-T\hat{c}[k]$, R_α the rotation matrix of angle $\hat{a}^- [k+1]$, $\Delta p^o [k]$ the robot's displacement in position, measured by odometry, between times kT and $(k+1)T$ expressed in the frame \mathcal{F} at time $(k+1)T$, and $\Delta \alpha^o [k] = \alpha^o [k+1] - \alpha^o [k]$ the change in orientation of the robot between times kT and $(k+1)T$, also measured by odometry.

The estimation of $X[k+1]$, which uses visual measurements at time $(k+1)T$, is then given by

$$\begin{aligned} \hat{X}_p [k+1] &= \hat{X}_p^- [k+1] + TK_p(p^v - \hat{p}^-)[k+1] \\ \hat{X}_\alpha [k+1] &= \hat{X}_\alpha^- [k+1] + TK_\alpha(\alpha^v - \hat{\alpha}^-)[k+1] \end{aligned}$$

Transverse function: The control function (11) depends on the variable θ . In fact, one easily verifies that it depends on $\eta_1 = \sin \theta$ and $\eta_2 = \cos \theta$. Since $\dot{\eta}_1 = \dot{\theta}\eta_2$ and $\dot{\eta}_2 = -\dot{\theta}\eta_1$, the discretization of η_1 and η_2 , when $\dot{\theta}$ is constant on $[kT, (k+1)T)$, yields

$$\Psi[k+1] = R(-T\dot{\theta}[k])\Psi[k] \quad (20)$$

with $\Psi[k] = (\eta_1[k], \eta_2[k])^T$ and $R(-T\dot{\theta}[k])$ the rotation matrix of angle $-T\dot{\theta}[k]$. In order to guarantee the numerical stability of $\|\Psi[k]\| - 1 = 0$, the following modified version of (20) is implemented:

$$\Psi[k+1] = R(-T\dot{\theta}[k])\Psi[k] (1 + \beta(\|\Psi[k]\|^2 - 1)) \quad (21)$$

with $\beta \in (-1, 0)$.

When implementing (21), an initial value $\Psi[0]$ or, equivalently, an initial value $\theta(0)$ must be chosen. We choose $\theta(0)$ so as to minimize $|z_3(0)| = |\varepsilon_2 \cos \theta(0) - \alpha(0)|$. More precisely, if $|\alpha(0)| \leq \varepsilon_2$, $|\theta(0)|$ is chosen so as to have $z_3(0) = 0$. In this case, if K in (11) is diagonal, then $z_3(t)$ (theoretically) remains equal to zero all the time. As for the sign of $\theta(0)$, we choose it according to the position of the robot with respect to the target. More precisely, we take $\theta(0)$ negative if $x(0)$ is positive, and positive otherwise. This choice tends to reduce initial transient oscillations. If $|\alpha(0)| > \varepsilon_2$, we let $\theta(0) = 0$ if $\alpha(0) \geq 0$, and $\theta(0) = \pi$ otherwise.

Input saturations: In order to take into account physical velocity limitations $|u_i| \leq u_{i,\max}$ ($i = 1, 2$), the control \bar{u} in (11) is rescaled as $\bar{u}_{new} = \tau^{-1}\bar{u}$ with $\tau = \max\{1, |u_i|/u_{i,\max}, i = 1, 2\}$. One can show that this rescaling does not prevent the convergence of z to zero, provided that the target maintains its velocity within the range compatible with the saturations. Moreover, in the case where the target is motionless the geometric path followed by the robot is not modified by this rescaling.

7 Simulation results

In the simulation results presented in Fig. 2-4, the matrix K in (11) is given by $K = -0.5I_3$ with I_3 the identity matrix. The transverse function (9) is used with $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 1$. The observer gains are defined by (17) with $\lambda_\alpha = \lambda_p = \beta_\alpha = \beta_p = 2$. The estimation $g^e = \hat{g}$ is used in the controller. The discretized version of the control law (see Section 6) is applied, with the sampling period $T = 40\text{ms}$.

The target motion is as follows. Initially, the target's posture correspond to $(x_r, y_r) = (1, 0.5)$ and $\alpha_r = 0$. After 10s, the target moves with constant longitudinal speed along the fixed axis \vec{v}_0 . For $t \in (20\text{s}, 30\text{s})$, it is motionless. For $t \in (30\text{s}, 40\text{s})$, the target moves laterally along the axis \vec{j}_0 . This corresponds to a non-feasible motion for the robot. For $t \in (40\text{s}, 50\text{s})$ the target moves backward with constant longitudinal speed along the axis \vec{v}_0 , and is motionless thereafter. One can observe from Fig. 4 that

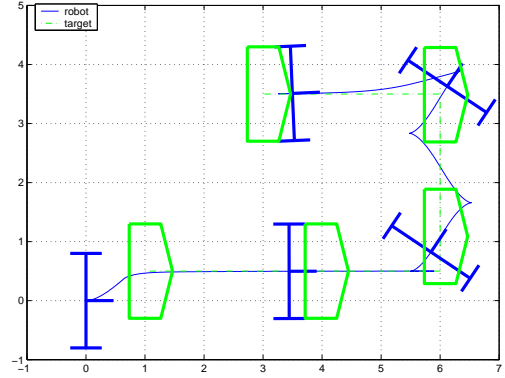


Figure 2: Cartesian motion: robot and target

z sometimes departs from zero. This corresponds to abrupt changes in the target's velocity yielding transient estimation errors which in turn reflect on the control performance.

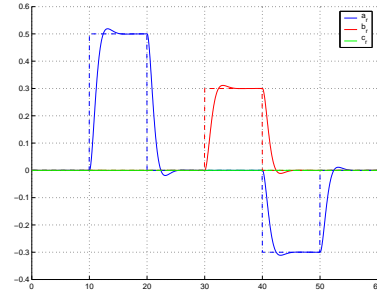


Figure 3: Estimated target speed

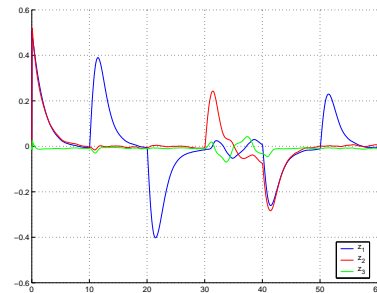


Figure 4: $z \triangleq f(\theta)g^{-1}$

8 Experimental results

The proposed control law has also been tested experimentally on a unicycle-like robot carrying a 6-d.o.f

manipulator arm with a CCD camera at its extremity. The measurements g^v of the robot's configuration with respect to the target are reconstructed from visual data. More details on the experimental set-up can be found in [6]. Since we do not have sensors providing absolute position and target velocity measurements, only experimentation results with a fixed target are reported. They illustrate the influence of the ε_i 's on the robot behaviour: $\varepsilon_2 = 0.3$ for Fig. 5, $\varepsilon_2 = 1$ for Fig. 7. In both cases, $\varepsilon_1 = 0.5$, and the gain matrices are those used for the simulation results of Section 7. The cartesian motion is drawn from the estimation $g^e = \hat{g}$. Simulation results with the same initial conditions and control gains are also reported on Fig. 6 and 8.

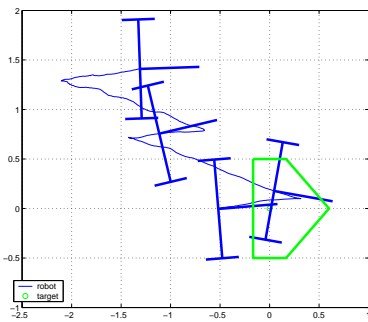


Figure 5: Cartesian motion of the robot – Experiment with $\varepsilon_2 = 0.3$

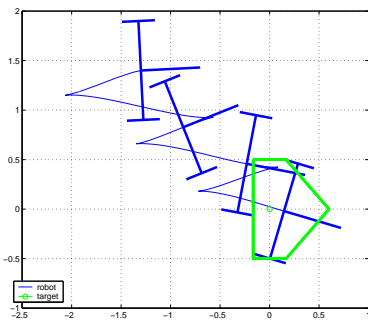


Figure 6: Cartesian motion of the robot – Simulation with $\varepsilon_2 = 0.3$

References

- [1] A. de Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In J.-P. Laumond, editor, *Robot motion planning and control*, volume 229 of *LNCIS*. Springer Verlag, 1998.
- [2] W.E. Dixon, D.M. Dawson, E. Zergeroglu, and F. Zhang. Robust tracking and regulation control

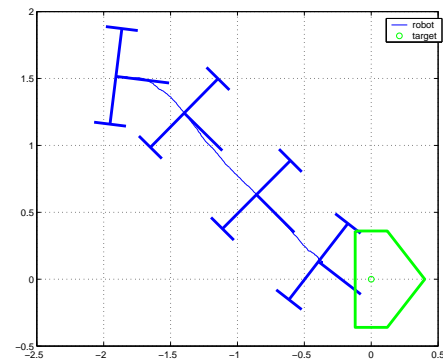


Figure 7: Cartesian motion of the robot – Experiment with $\varepsilon_2 = 1.0$

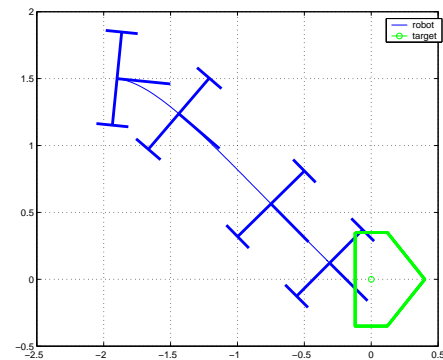


Figure 8: Cartesian motion of the robot – Simulation with $\varepsilon_2 = 1.0$

for mobile robots. *International Journal of Robust and Nonlinear Control*, 10:199–216, 2000.

- [3] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, 1972.
- [4] P. Morin and C. Samson. Practical stabilization of driftless systems on Lie groups. In *IEEE CDC*, 2002.
- [5] C. Samson. Control of chained systems. Application to path following and time-varying point-stabilization. *IEEE Trans. on Automatic Control*, 40:64–77, 1995.
- [6] D.P. Tsakiris, K. Kapellos, C. Samson, P. Rives, and J.J. Borrelly. Experiments in real-time vision-based point stabilization of a nonholonomic mobile manipulator. In *5th Inter. Symp. on Exp. Rob.*, 1997.