Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Wheeled Mobile Robots 4

Motion Control of WMRs: Trajectory Tracking

Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti

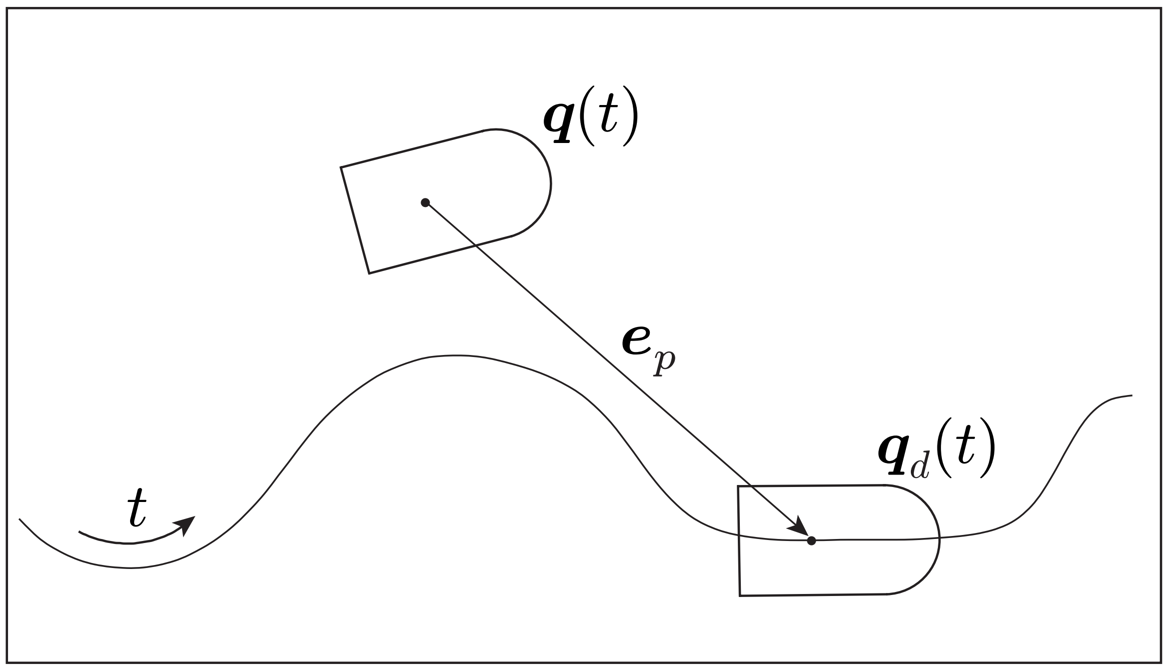
Sapienza
Università di Roma

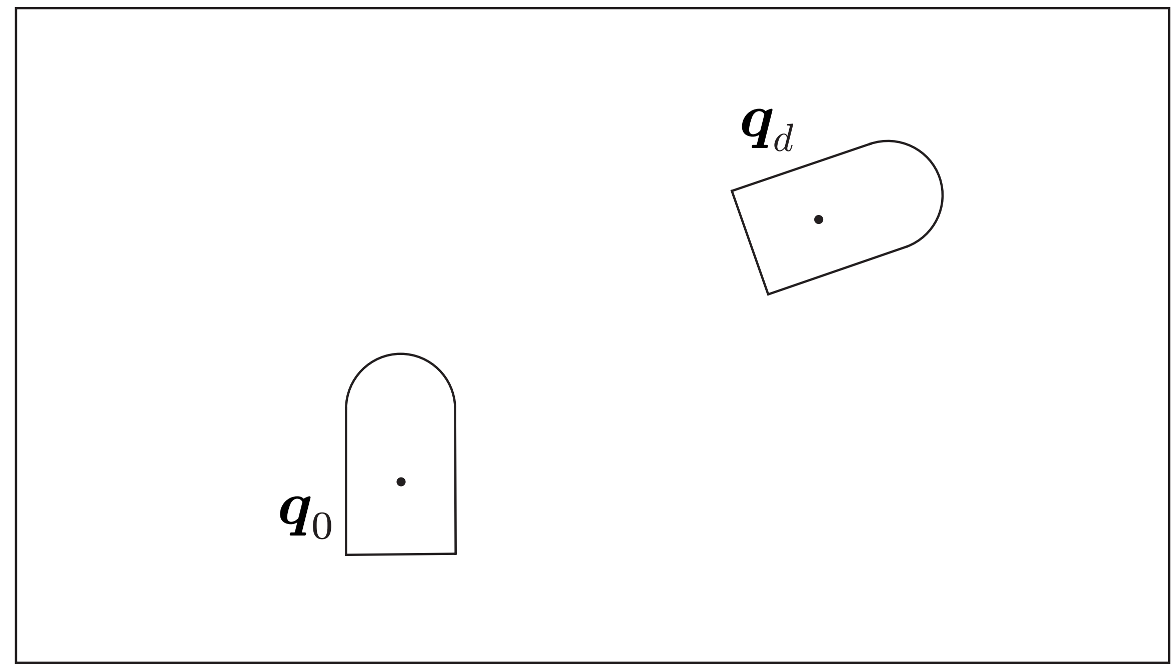
motion control

- feedback control is needed because the application of nominal (open-loop) control inputs would lead to unacceptable performance in practice
- kinematic models are used to design feedback laws because (1) dynamic terms can be canceled via feedback (2) wheel actuators are equipped with low-level PID loops that accept velocities as reference
- we consider a unicycle in the following

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega$$

motion control problems





trajectory tracking
(predictable transients)

posture regulation
(no prior planning)

trajectory tracking

- the desired (reference) trajectory $(x_d(t), y_d(t))$ must be admissible: there must exist v_d and ω_d such that

$$\dot{x}_d = v_d \cos\theta_d$$
$$\dot{y}_d = v_d \sin\theta_d$$
$$\dot{\theta}_d = \omega_d$$

- thanks to flatness, given $(x_d(t), y_d(t))$ we can compute

$$\theta_d(t) = \mathrm{Atan2}\left(\dot{y}_d(t), \dot{x}_d(t)\right) + k\pi \qquad k = 0, 1$$
$$v_d(t) = \pm\sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)}$$
$$\omega_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}$$

- rather than using directly the state error $\boldsymbol{q}_d - \boldsymbol{q}$, use its rotated version defined as

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

(e_1, e_2) is $\boldsymbol{e}_p$ (previous figure) in a frame rotated by θ

- we find

$$\dot{e}_1 = v_d \cos e_3 - v + e_2\,\omega$$
$$\dot{e}_2 = v_d \sin e_3 - e_1\,\omega$$
$$\dot{e}_3 = \omega_d - \omega$$

- now use the input transformation

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

which is clearly invertible

- we obtain

$$
\dot{\boldsymbol{e}} = \underbrace{\begin{pmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{e}}_{\substack{\text{linear} \\ \text{time-varying}}} + \underbrace{\begin{pmatrix} 0 \\ \sin e_3 \\ 0 \end{pmatrix} v_d}_{\substack{\text{nonlinear} \\ \text{time-varying}}} + \underbrace{\begin{pmatrix} 1 & -e_2 \\ 0 & e_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}}_{\text{nonlinear}}
$$

via approximate linearization

- linearize the error dynamics around the reference trajectory ($\boldsymbol{e} \approx \mathbf{0}$, hence $\sin e_3 \approx e_3$)

$$\dot{\boldsymbol{e}} = \begin{pmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{e} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

- define the linear feedback

$$u_1 = -k_1\, e_1$$
$$u_2 = -k_2\, e_2 - k_3\, e_3$$

- we obtain

$$\dot{\boldsymbol{e}} = \boldsymbol{A}(t)\, \boldsymbol{e} = \begin{pmatrix} -k_1 & \omega_d & 0 \\ -\omega_d & 0 & v_d \\ 0 & -k_2 & -k_3 \end{pmatrix} \boldsymbol{e}$$

- letting

$$k_1 = k_3 = 2\zeta a \qquad k_2 = \frac{a^2 - \omega_d^2}{v_d}$$

with $a>0$, $\zeta\in(0,1)$, the characteristic polynomial of $\boldsymbol{A}(t)$ becomes time-invariant

$$p(\lambda) = \underbrace{(\lambda + 2\zeta a)}_{\substack{\text{real}\\ \text{negative}\\ \text{eigenvalue}}}\underbrace{(\lambda^2 + 2\zeta a\lambda + a^2)}_{\substack{\text{pair of complex}\\ \text{eigenvalues with}\\ \text{negative real part}}}$$

- caveat: this does not guarantee asymptotic stability, unless v_d and ω_d are constant (as on circles and lines); even in this case, asymptotic stability of the unicycle is not global (indirect Lyapunov method)

- the actual velocity inputs v,ω are obtained plugging the feedbacks u_1, u_2 in the input transformation
- note: $(v,\omega) \to (v_d,\omega_d)$ as $\boldsymbol{e} \to \mathbf{0}$ (pure feedforward)
- note: $k_2 \to \infty$ as $v_d \to 0$, hence this controller can only be used with persistent cartesian trajectories (stops are not allowed)

- global stability is guaranteed by a nonlinear version

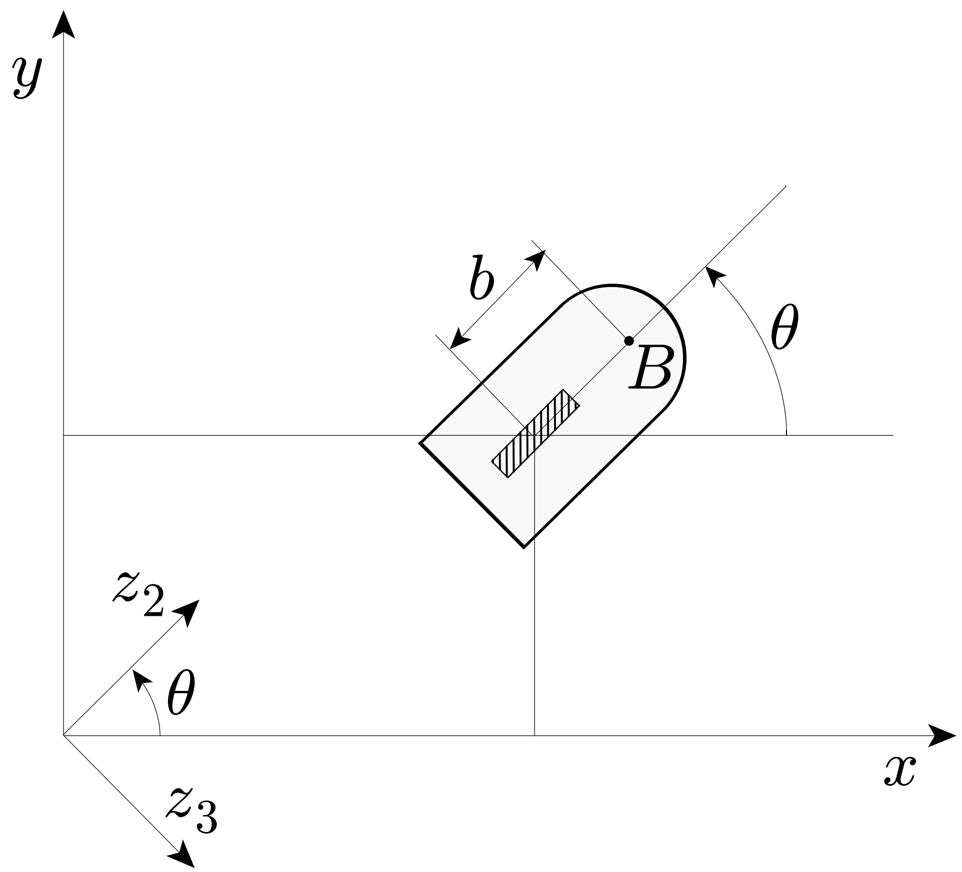
$$u_1 = -k_1(v_d, \omega_d)\, e_1$$
$$u_2 = -k_2\, v_d \, \frac{\sin e_3}{e_3}\, e_2 - k_3(v_d, \omega_d)\, e_3$$

if k_1,k_3 bounded, positive, with bounded derivatives

via input/output linearization

- idea: find an output whose dynamics can be made linear via feedback, i.e., with an input transformation

- choose the cartesian coordinates of point B as output



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

- differentiating wrt time

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{pmatrix}}_{\text{determinant} = b} \begin{pmatrix} v \\ \omega \end{pmatrix} = \boldsymbol{T}(\theta) \begin{pmatrix} v \\ \omega \end{pmatrix}$$

- if $b \neq 0$, we may set

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \boldsymbol{T}^{-1}(\theta) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta/b & \cos\theta/b \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

obtaining

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

- achieve global exponential convergence of y_1, y_2 to the desired trajectory letting

$$\begin{aligned} u_1 &= \dot{y}_{1d} + k_1(y_{1d} - y_1) \\ u_2 &= \dot{y}_{2d} + k_2(y_{2d} - y_2) \end{aligned}$$

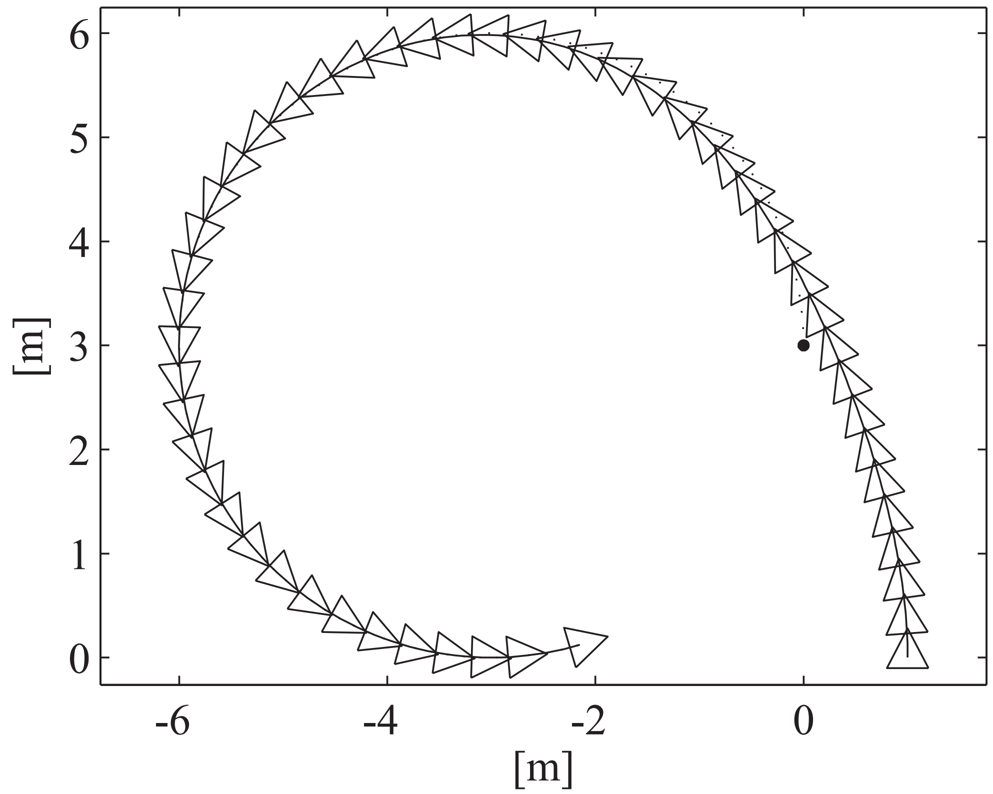
  with $k_1, k_2 > 0$

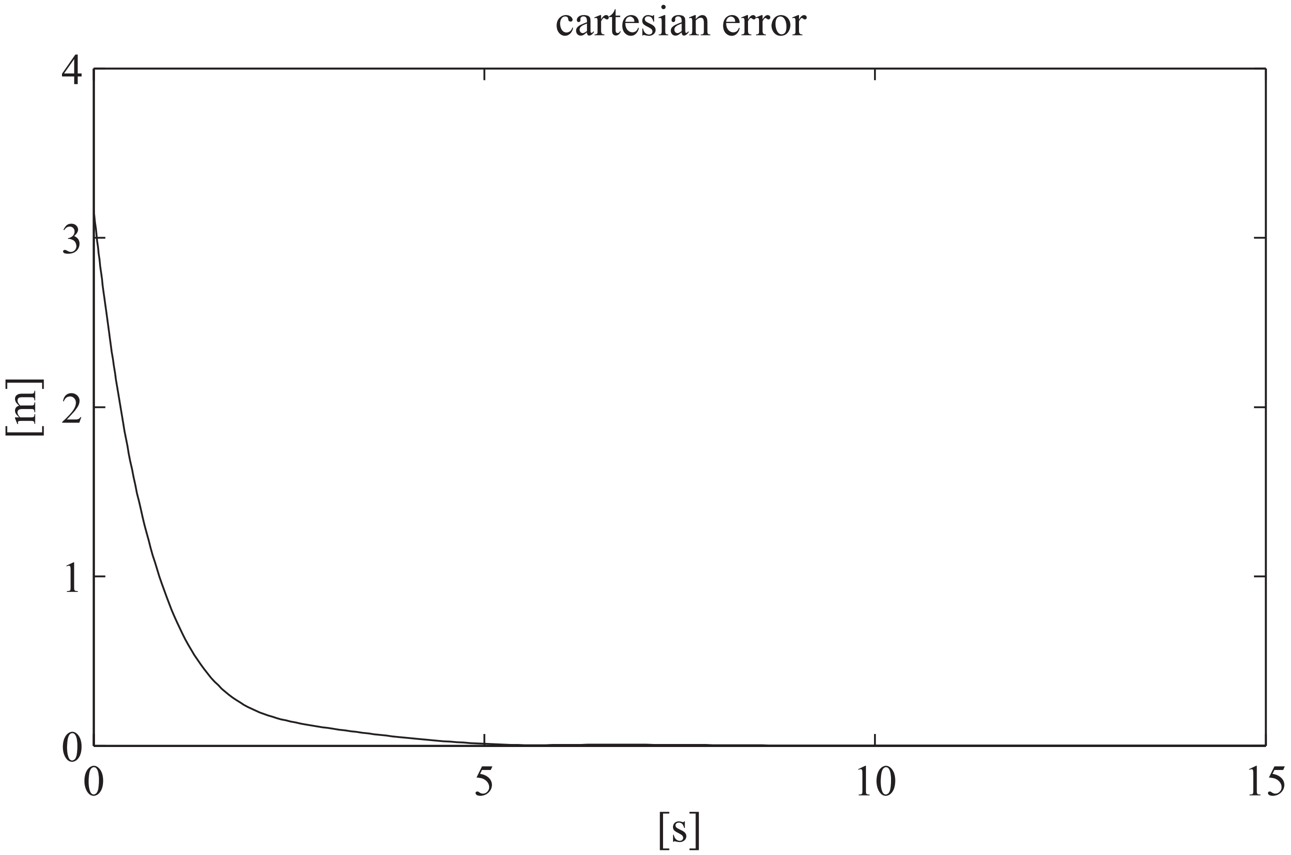
- θ is not controlled with this scheme, which is based on output error feedback (compare with the previous)

- the desired trajectory for B can be arbitrary; in particular, square corners may be included

simulations

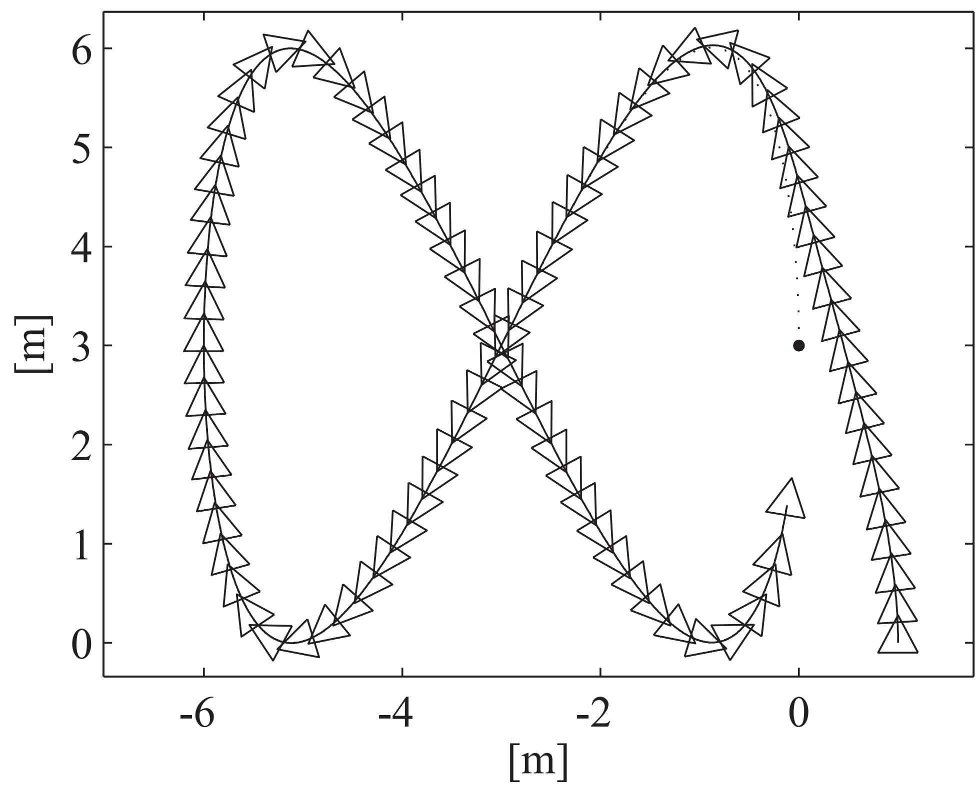
tracking a circle via approximate linearization

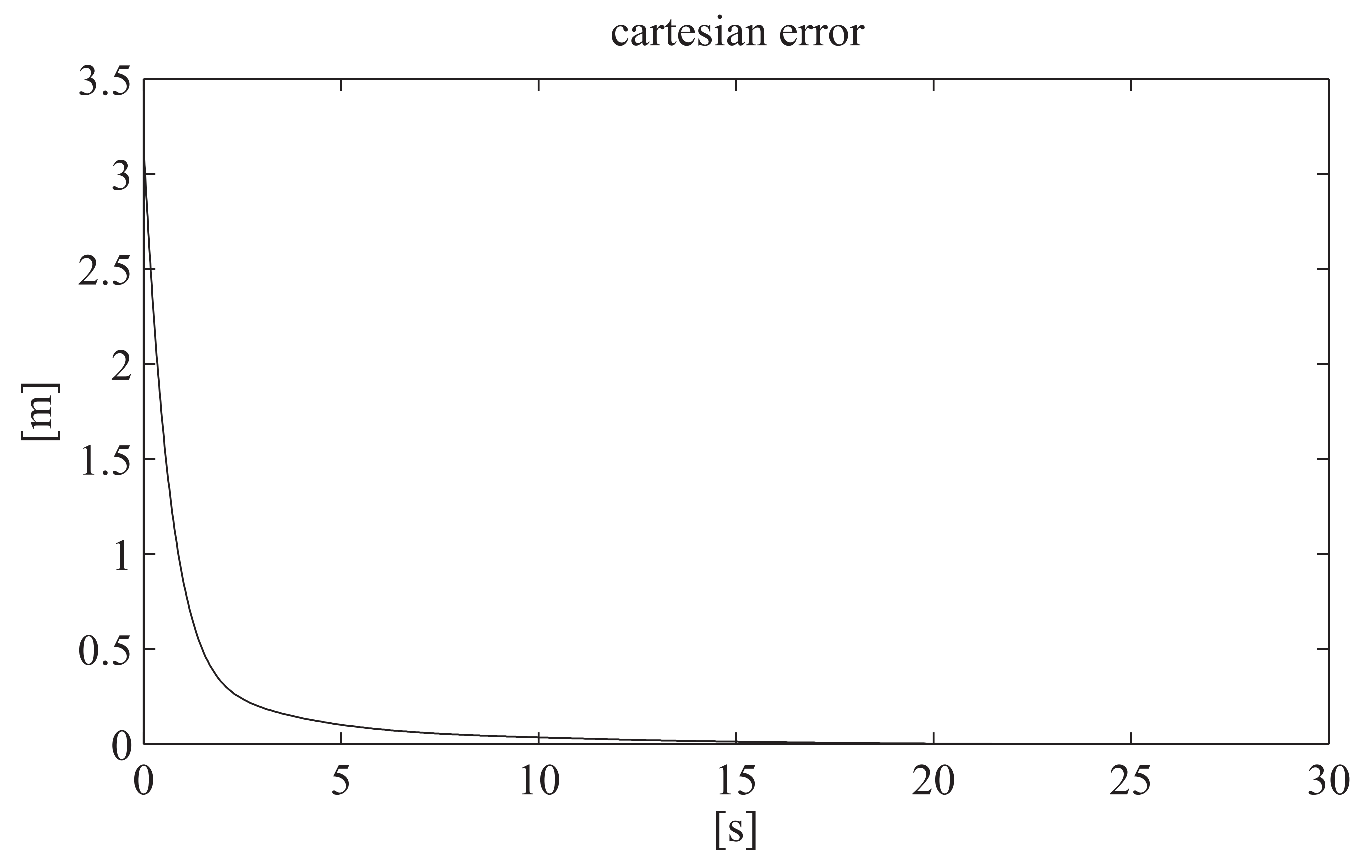




simulations

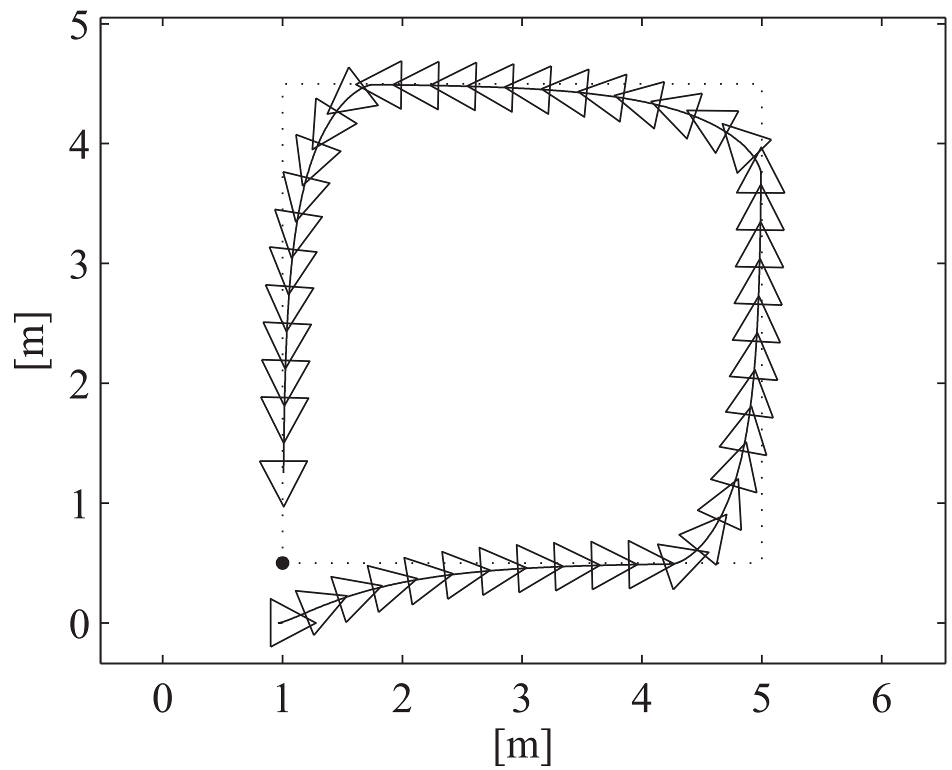
tracking an 8-figure via nonlinear feedback

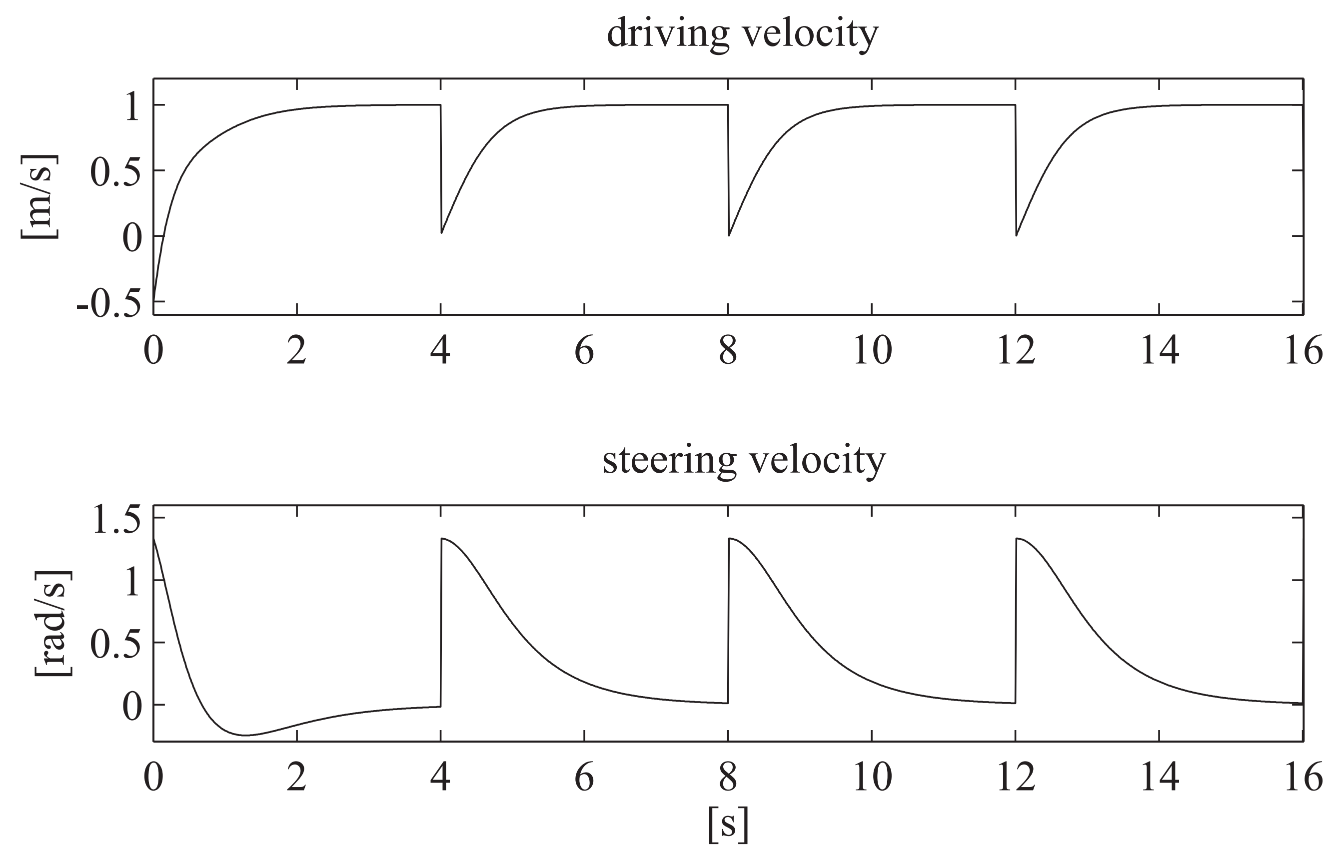




simulations

tracking a square via i/o linearization

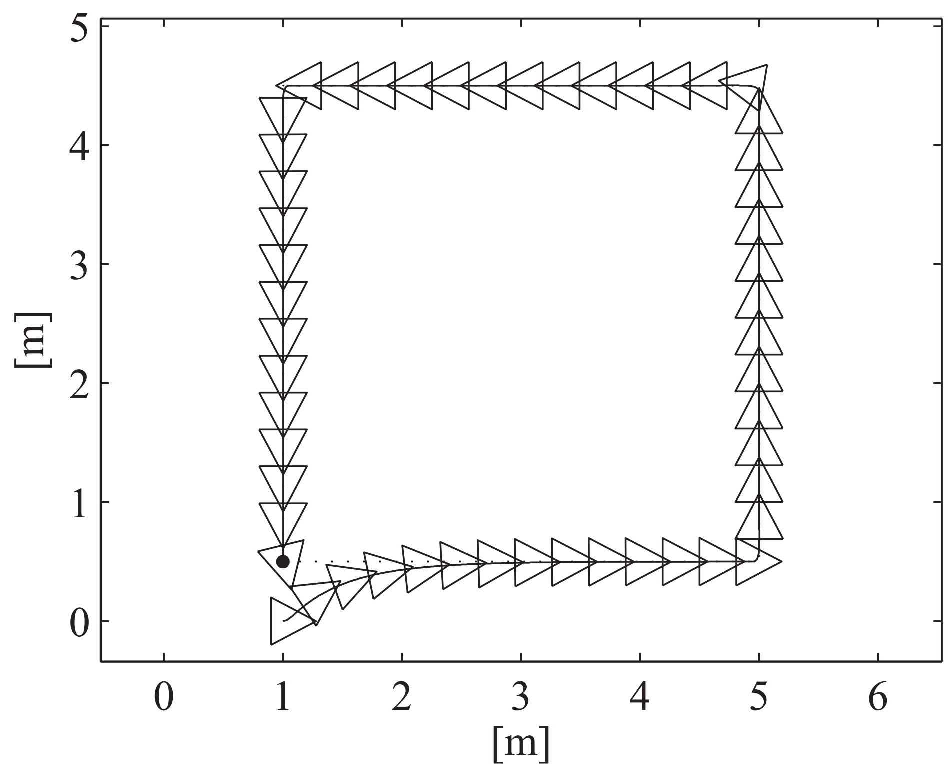


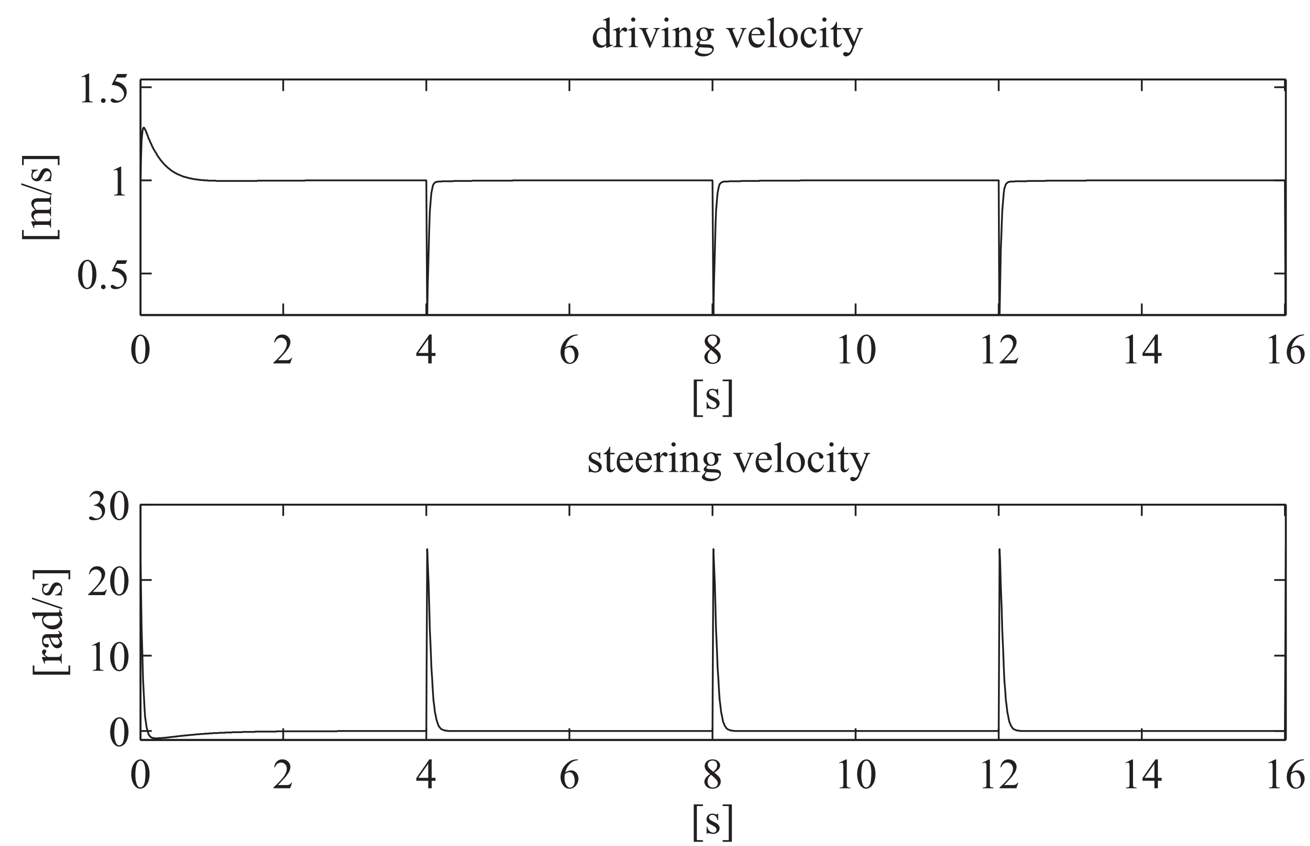


$b = 0.75 \Rightarrow$ the unicycle rounds the corners

simulations

tracking a square via i/o linearization





$b=0.2 \Rightarrow$ accurate tracking but velocities increase