Robotics 1

Wheeled Mobile Robots

Analysis, Planning, and Control

Prof. Alessandro De Luca

Dipartimento di Informatica
e Sistemistica Antonio Ruberti

Sapienza
Università di Roma

Summary

- uses of the kinematic model of WMR
  - controllability analysis* (for nonlinear systems)
  - odometry
  - model transformations
  - time scaling
  - path/trajectory planning
  - control design
    - regulation to a configuration
    - trajectory tracking

* background on differential geometry in appendix

Controllability analysis

$$A(q)\dot{q} = 0 \rightleftharpoons \dot{q} = G(q)v = \sum_{i=1}^{m} g_i(q)v_i$$

are the differential constraints integrable or not?

=

can the robot reach by suitable maneuvers any point q
in the configuration space $\mathcal{C}$*? (or, is the system "controllable"?)*

- tools and answers come from nonlinear control theory

  first tool: *Lie bracket* of vector fields $g_1(q)$ and $g_2(q)$

$$[g_1, g_2](q) = \frac{\partial g_2}{\partial q} g_1(q) - \frac{\partial g_1}{\partial q} g_2(q)$$

  provides a new direction for motion, beyond those given by $g_1(v_2 = 0)$, $g_2(v_1 = 0)$, and their linear combinations $g_1\bar{v}_1 + g_2\bar{v}_2$

Controllability analysis (cont'd)

second tool: *accessibility distribution* $\mathcal{A}$ generated by a set of vector fields (through repeated *Lie bracketing*)

e.g., for m=2

$$\begin{aligned}\mathcal{A} &= \{g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]], \ldots\} \\ &= \text{Lie Algebra generated by } g_1, g_2\end{aligned}$$

Theorem

$$\boxed{\begin{gathered}\operatorname{rank} \mathcal{A} = n \qquad \forall q \in \mathcal{C} \\ \Updownarrow \\ \mathcal{C} \text{ completely accessible} \\ \Updownarrow \\ A(q)\dot{q} = 0 \text{ nonholonomic}\end{gathered}}$$

← with maneuvers!

note 1: is a nonlinear version of the Kalman test for controllability of linear dynamic systems

$$\dot{x} = Ax + Bu \qquad \operatorname{rank}\left[B \ AB \ A^2B \ \ldots \ A^{n-1}B\right] = n$$

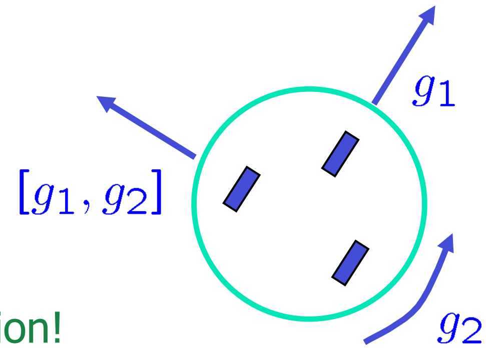
note 2: the "levels" (order) of Lie bracketing needed to obtain the maximum rank is an index of the difficulty of maneuvering the WMR (# of elementary maneuvers to achieve a generic motion grows with the bracketing level)

Controllability of unicycle

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \in \mathcal{C} \qquad g_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\dim \mathcal{C} = n = 3$$

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



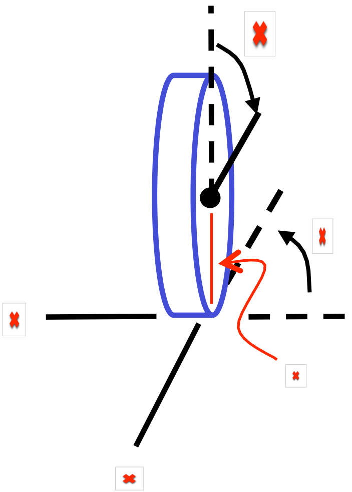
it is the lateral direction!

$$\Rightarrow \quad \text{rank} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta & 0 & -\cos\theta \\ 0 & 1 & 0 \end{bmatrix} = 3 = n!$$

note: in this simple case, even the simpler sequence "rotate-translate-rotate" allows to move the robot between any two configurations …

Controllability of "extended" unicycle

taking into account also the rolling angle of the wheel …



$$q = \begin{bmatrix} x \\ y \\ \psi \\ \theta \end{bmatrix} \in \mathcal{C} \qquad g_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 1/r \\ 0 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\dim \mathcal{C} = n = 4$$

$$[g_1, g_2] = \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \\ 0 \end{bmatrix} \qquad [g_2, [g_1, g_2]] = \begin{bmatrix} -\cos\theta \\ -\sin\theta \\ 0 \\ 0 \end{bmatrix}$$

$\Rightarrow$ rank $\begin{bmatrix} g_1 \; g_2 \; [g_1, g_2] \; [g_2, [g_1 \; g_2]] \end{bmatrix} = 4 = n!$

it is thus possible, by suitable maneuvering, to reach any point (x,y) on the plane, with any desired final wheel orientation θ and also final rolling angle ψ

Controllability of car-like (RD)

$$q = \begin{bmatrix} x \\ y \\ \theta \\ \phi \end{bmatrix} \in \mathcal{C} \qquad g_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi/\ell \\ 0 \end{bmatrix} \qquad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

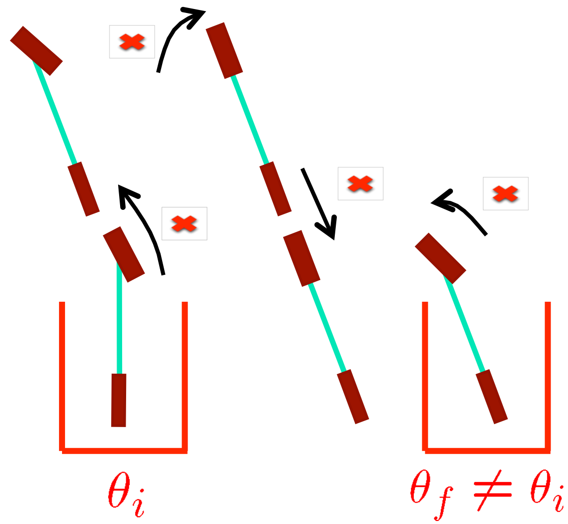
$\dim\mathcal{C} = n = 4$

g_1: "forward" direction

g_2: "steering" direction

$$[g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ -1/\ell\cos^2\phi \\ 0 \end{bmatrix}$$

"new" rotation direction for the robot



$$[g_1, [g_1, g_2]] = -\frac{1}{\ell\cos^2\phi} \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \\ 0 \end{bmatrix}$$

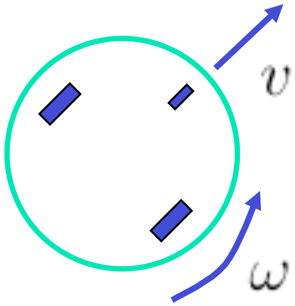
"new" lateral direction for the robot

it is a sequence of 8 elementary commands

$$\text{rank} \begin{bmatrix} g_1 & g_2 & [g_1, g_2] & [g_1, [g_1\, g_2]] \end{bmatrix} = 4 = n!$$

Odometry

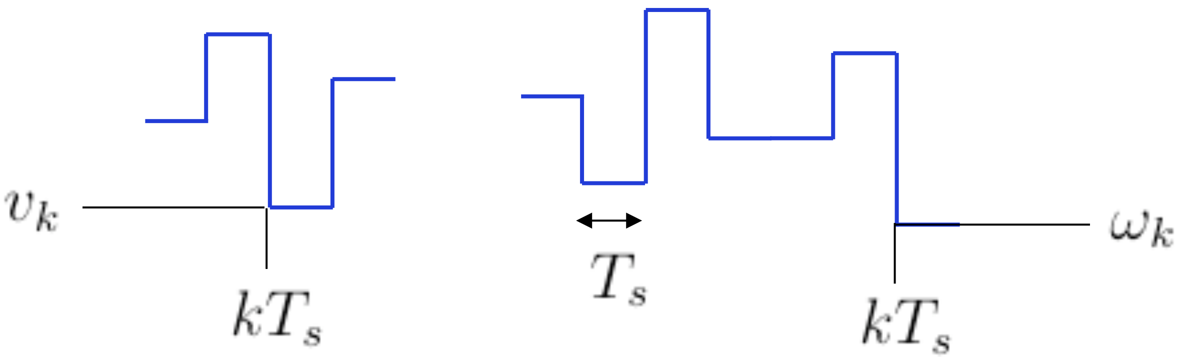
- incremental localization using odometry, i.e., based on proprioceptive measures of the wheels encoders



$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned}$$

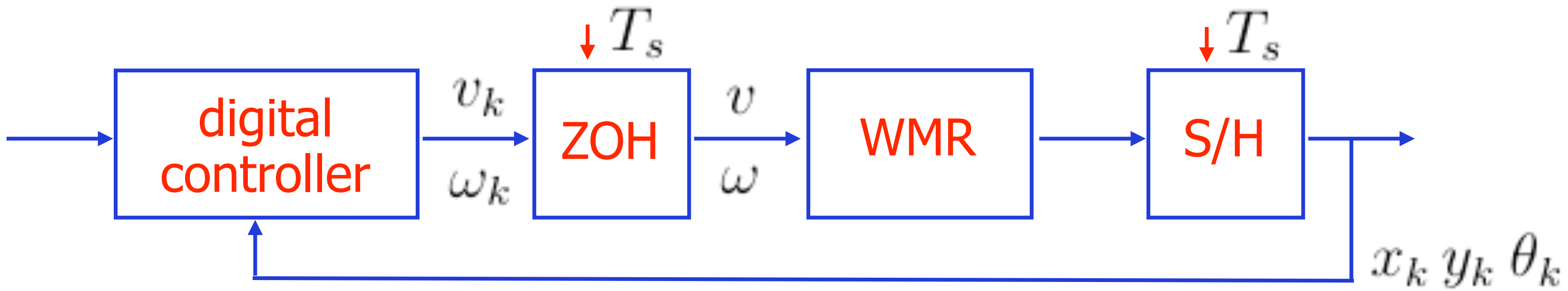
unicycle model (SuperMARIO)

- assumption: the (linear, angular) velocity commands are constant over a control sampling period T_s



Odometry (cont'd)

- digital control scheme



- motion equations can be integrated numerically in an approximate way (exact for the orientation!) using different methods

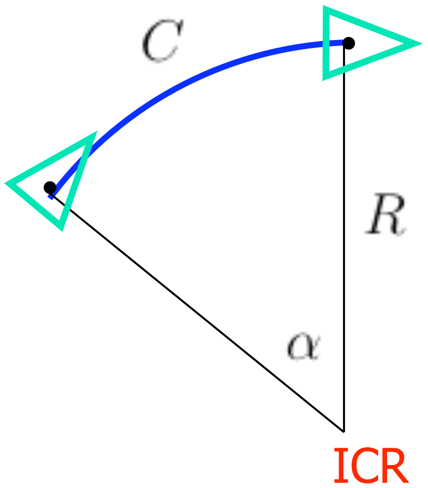
$$\begin{aligned} x_{k+1} &= x_k + v_k T_s \cos\theta_k \\ y_{k+1} &= y_k + v_k T_s \sin\theta_k \\ \theta_{k+1} &= \theta_k + \omega_k T_s \end{aligned}$$

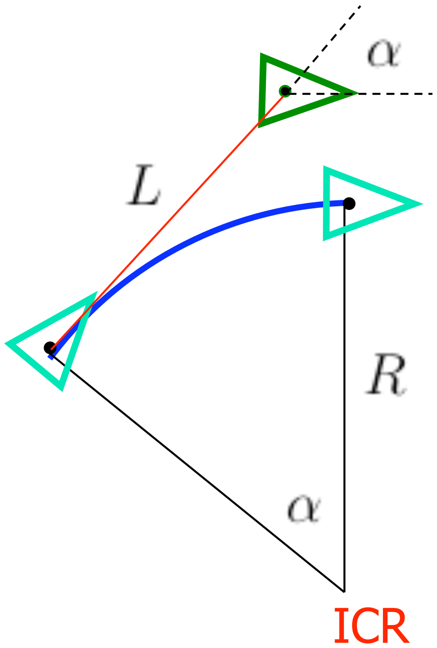
Euler

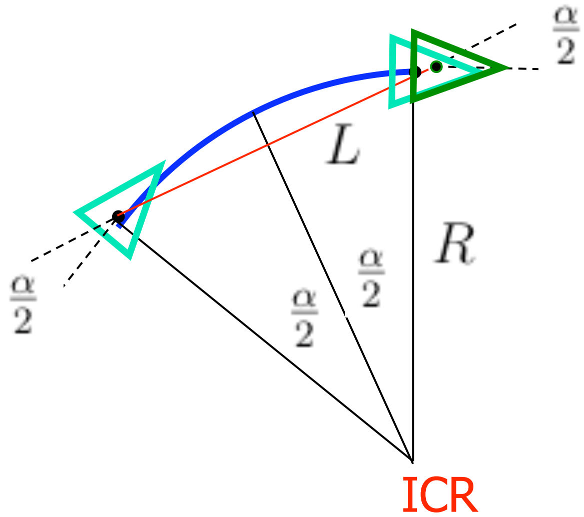
$$\begin{aligned} x_{k+1} &= x_k + v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ y_{k+1} &= y_k + v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ \theta_{k+1} &= \theta_k + \omega_k T_s \end{aligned}$$

2nd order Runge-Kutta

Odometry (cont'd)







exact solution

$$R = \frac{v}{\omega}$$
$$\alpha = \omega T_s$$
$$C = \alpha R = v T_s$$

Euler

$$L = v T_s (= C)$$

large position error
for slow sampling rate
(or high speed)

2nd order
Runge-Kutta

reduced position error
(semi-angle method)

Odometry (cont'd)

- in practice, replace the velocity commands with the encoder readings of right and left wheels (angular increments $\Delta\psi_R$ and $\Delta\psi_L$)

$$
\begin{aligned}
v_k T_s &= r\,\frac{\Delta\psi_R + \Delta\psi_L}{2} \\
\omega_k T_s &= r\,\frac{\Delta\psi_R - \Delta\psi_L}{2d}
\end{aligned}
$$

r = (common) radius of the two wheels

- actually, within a sampling period with constant velocities, the WMR executes an arc of circumference with radius

$$R = \frac{v}{\omega}$$

- odometric computations provide an estimate of the pose localization of the WMR, which is more reliable for small sampling times and in the absence of wheel slippage

Model transformations

- different transformations can be used on the kinematic model of a WMR, yielding equivalent models which allow more direct solutions to the path/trajectory planning and control problems
- a viable transformation is that into the so-called chained form
- for the unicycle model, using the coordinate transformation

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

and the input transformation (both globally invertible)

$$\begin{aligned} v_1 &= \omega \\ v_2 &= v - z_3\omega = v - (x\sin\theta - y\cos\theta)\omega \end{aligned} \quad \longleftrightarrow \quad \begin{aligned} v &= v_2 + z_3 v_1 \\ \omega &= v_1 \end{aligned}$$

we get

$$\boxed{\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1 \end{aligned}} \quad \equiv$$

chained form
for n = 3

a polynomial system, in particular linear if v_1 is constant (for a certain time interval)

Model transformations (cont'd)

- in general, for

$$\dot{q} = g_1(q)u_1 + g_2(q)u_2 \qquad\qquad q \in \mathcal{C}, \dim\mathcal{C} = n \geq 2$$

it may exists (at least) a coordinate and input transformation

$$z = T(q) \qquad v = \beta(q)u$$

such that the WMR kinematic model is in chained form

$$\begin{aligned} \dot{z}_1 &= v_1 \\ \dot{z}_2 &= v_2 \\ \dot{z}_3 &= z_2 v_1 \\ &\vdots \\ \dot{z}_n &= z_{n-1} v_1 \end{aligned}$$

- a chained form always exists (at least locally) for $n \leq 4$ (e.g., unicycle, car-like, with one added trailer) and also for WMR with N trailers all with zero-hooking (each attached to the midpoint of the previous axle)

Time scaling

- decomposition in space s and time t

$$\dot{q} = \frac{dq}{dt} = \frac{dq}{ds}\frac{ds}{dt} = q'\dot{s}$$

- separation of the kinematic model of the unicycle

$$\begin{aligned} x' &= dx/ds = \cos\theta\,\tilde{v} \\ y' &= dy/ds = \sin\theta\,\tilde{v} \\ \theta' &= d\theta/ds = \tilde{\omega} \end{aligned}$$

"geometric" inputs

with
$$\begin{aligned} v(t) &= \tilde{v}(s)\,\dot{s}(t) \\ \omega(t) &= \tilde{\omega}(s)\,\dot{s}(t) \end{aligned}$$

(*s(t)* = common timing law)

- given $\tilde{v}(s), \tilde{\omega}(s)$ for $s \in [0, L]$, the geometric path is uniquely determined, but it can be executed with different timing laws
- the same holds for any (first order) kinematic model of a WMR

Flat outputs

- a nonlinear dynamical system

$$\dot{x} = f(x) + G(x)u$$

has the "differenzial flatness" property, if there exist a set of outputs (flat) such that the system state and inputs can be expressed algebraically in terms of these outputs and of a finite number of their derivatives

$$x = x(y, \dot{y}, \ddot{y}, \ldots, y^{(r)})$$
$$u = u(y, \dot{y}, \ddot{y}, \ldots, y^{(r)})$$

- any smooth trajectory (in time) or path (in space) for the system state and for the system inputs becomes a function of the outputs and of their (geometric or time) derivatives
- useful property for planning a reconfiguration between an initial and a final state and for finding the associated input commands

Flat outputs for the unicycle

- a unicycle is differentially flat with respect to the position coordinates of its "center" (flat outputs)
- for instance, in geometric terms, given $x(s)$ and $y(s)$ for $s \in [0,1]$

$$\theta(s) = \mathrm{Atan2}\left(y'(s), x'(s)\right) + k\pi \qquad k = 0,1$$

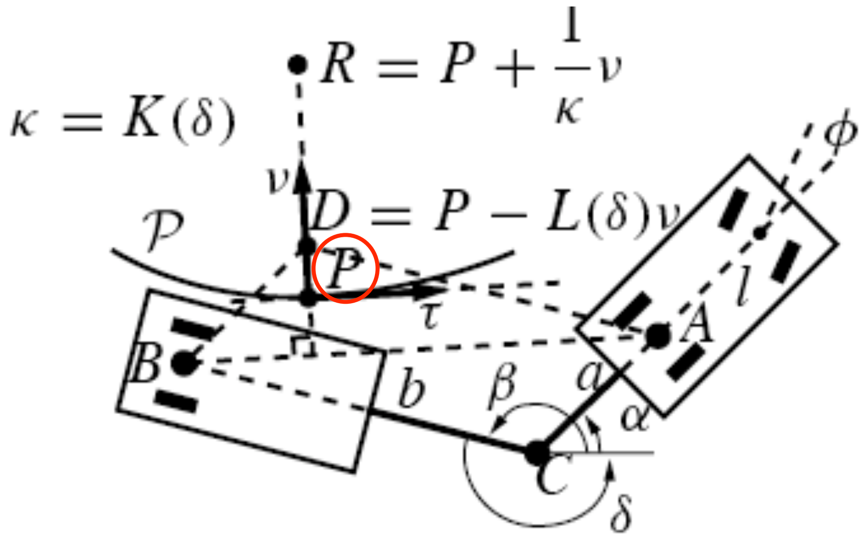
path executed in forward or backward motion

$$\tilde{v}(s) = \pm\sqrt{(x'(s))^2 + (y'(s))^2}$$

$$\tilde{\omega}(s) = \frac{y''(s)x'(s) - x''(s)y'(s)}{(x'(s))^2 + (y'(s))^2}$$

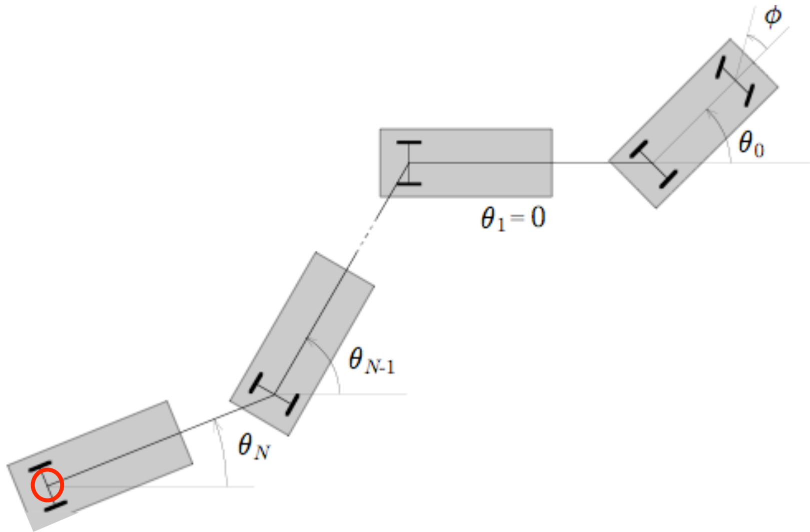
Flat outputs for more complex WMRs

- car + 1 trailer with general hooking



flat output: point P (variable with the geometry as a function of δ only)

- car + N trailers with zero-hooking

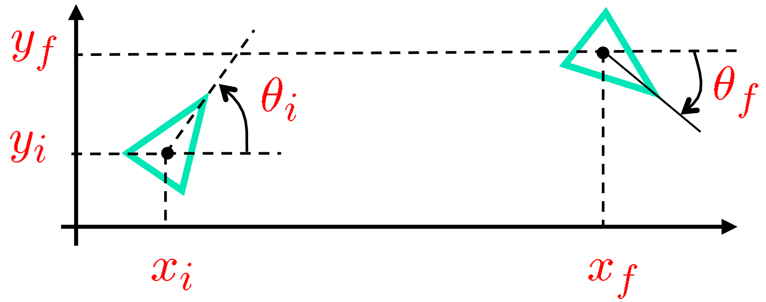


flat output: the center point of the axle of last trailer

- all "driftless" systems with two inputs (transformable) in chained from (flat output: z_1 and z_3)

Path planning

find a point-to-point path (between configurations) for the unicycle



motion task

an interpolation problem: use cubic polynomials for the flat outputs

$$
\begin{aligned}
x(s) &= -(s-1)^3 x^i + s^3 x^f + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \\
y(s) &= -(s-1)^3 y^i + s^3 y^f + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2
\end{aligned}
\qquad s \in [0,1]
$$

which automatically satisfy the boundary conditions on positions …

$$
\begin{aligned}
x(0) &= x^i & \qquad x(1) &= x^f \\
y(0) &= y^i & \qquad y(1) &= y^f
\end{aligned}
$$

… and have enough parameters for imposing initial and final orientations

Path planning (cont'd)

a solution is found by imposing other four conditions

$$x'(0) = K^i \cos\theta^i \qquad x'(1) = K^f \cos\theta^f$$
$$y'(0) = K^i \sin\theta^i \qquad y'(1) = K^f \sin\theta^f$$

with two free parameters $K^i \neq 0,\ K^f \neq 0$ (having same positive signs!)

it is important to "optimize" this value ...

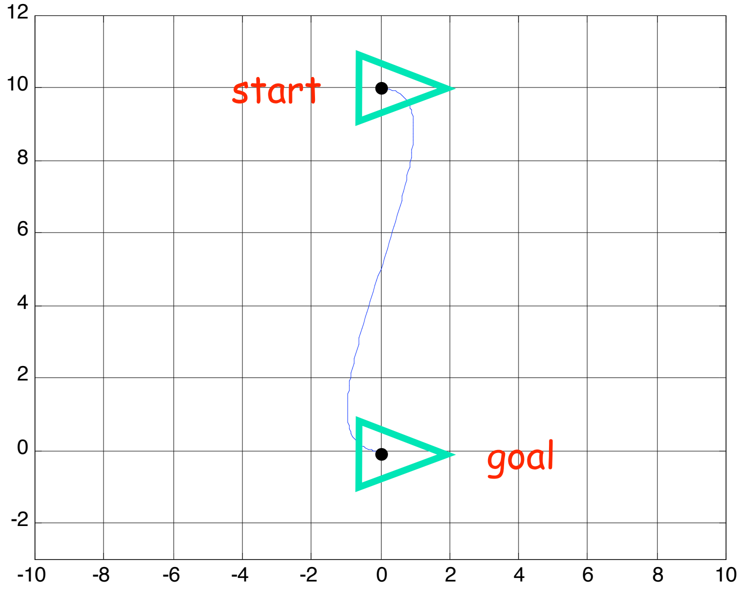
setting, e.g., $K^i = K^f = K$

$$\Rightarrow \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} K\cos\theta^f - 3x^f \\ K\sin\theta^f - 3y^f \end{bmatrix}, \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} K\cos\theta^i + 3x^i \\ K\sin\theta^i + 3y^i \end{bmatrix}$$

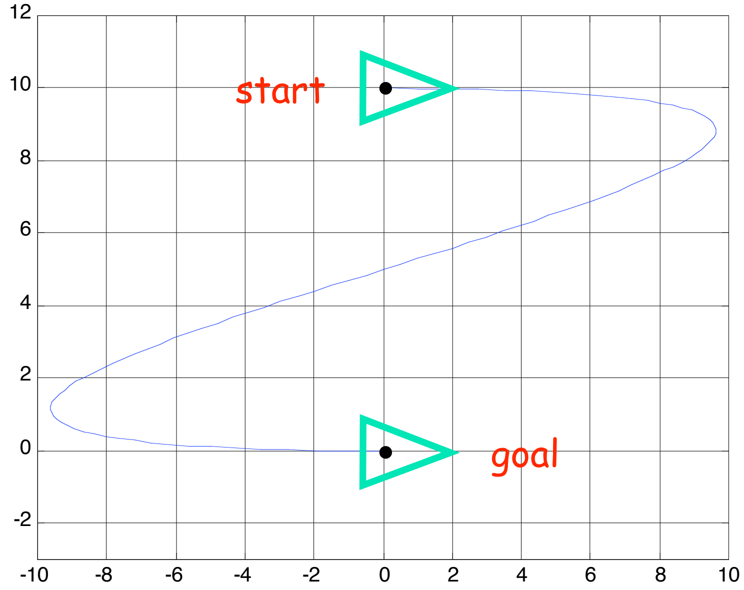
Numerical results

$$q = (x, y, \theta) \qquad q^i = (0, 10, 0) \rightarrow q^f = (0, 0, 0)$$

$K = 10$



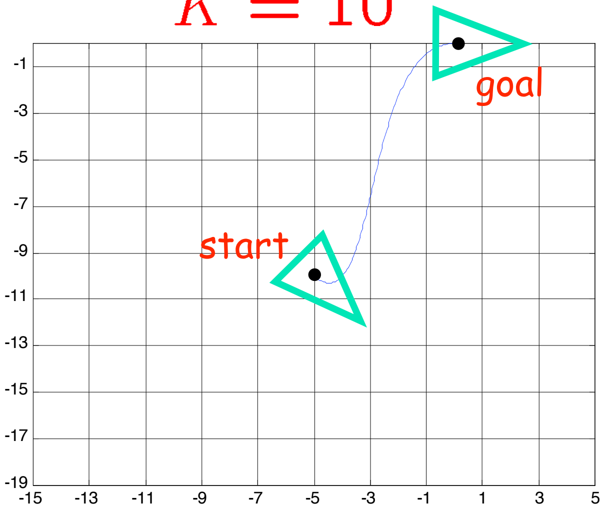
$K = 100$



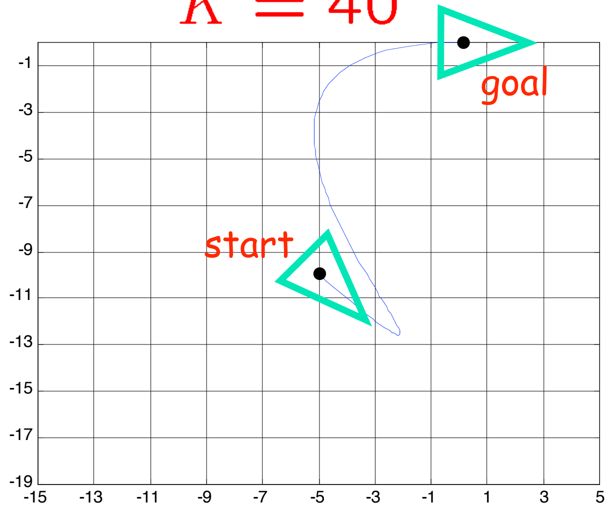
"parallel" parking

Numerical results (cont'd)

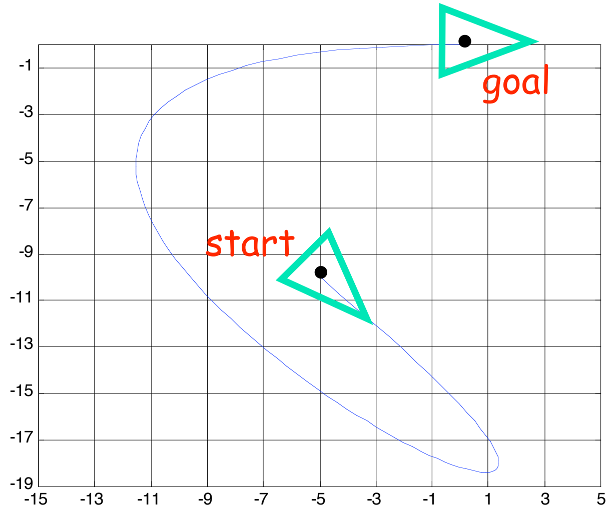
$K = 10$



$K = 40$



$K = 100$



$$q^i = (-5, -10, -\pi/4)$$

$$\downarrow$$

$$q^f = (0, 0, 0)$$

Path planning in chained form

as an alternative solution, we can work on the chained form

$$(x, y, \theta)^i \longrightarrow (z_1, z_2, z_3)^i$$
$$(x, y, \theta)^f \longrightarrow (z_1, z_2, z_3)^f$$

taking into account that: $z_2 = z_3'/z_1'$

and, as before, proceed with cubic polynomials (for the flat outputs)

however, one can reduce to a minimum the number of parameters to be found by using a linear/cubic combination

$$z_1(s) = z_{1,f}s - (s-1)z_{1,i}$$
$$z_3(s) = s^3 z_{3,f} - (s-1)^3 z_{3,i} + \alpha_3 s^2(s-1) + \beta_3 s(s-1)^2$$

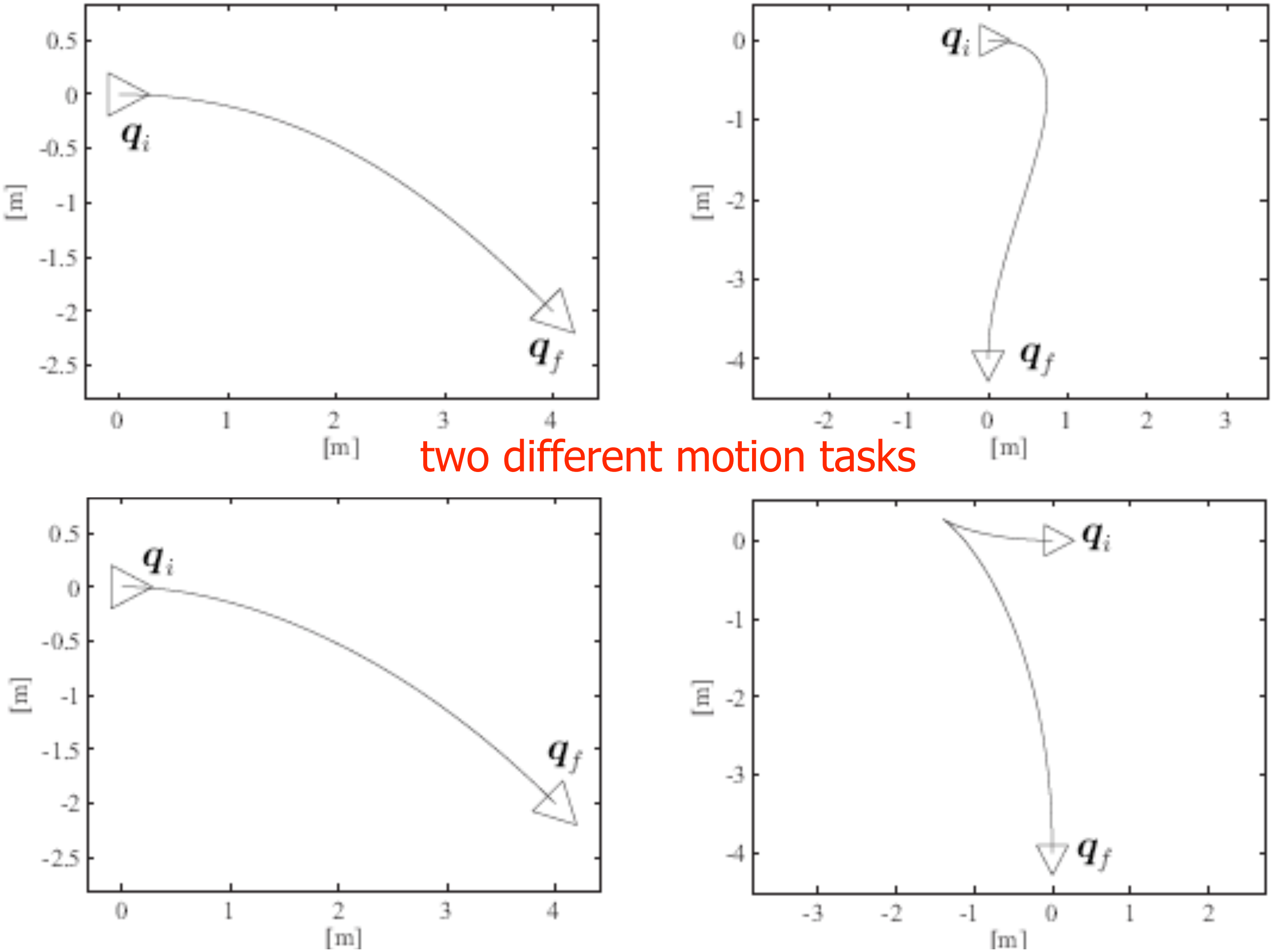
$s \in [0, 1]$

and imposing the boundary conditions

$$\frac{z_3'(0)}{z_1'(0)} = z_{2i} \qquad \frac{z_3'(1)}{z_1'(1)} = z_{2f} \qquad \Rightarrow \qquad \begin{aligned} \alpha_3 &= z_{2,f}(z_{1,f} - z_{1,i}) - 3z_{3,f} \\ \beta_3 &= z_{2,i}(z_{1,f} - z_{1,i}) + 3z_{3,i}. \end{aligned}$$

feasible only if $z_1'(s) = z_{1,f} - z_{1,i} = \theta_f - \theta_i \neq 0$ (else a "via point" is needed with a different orientation)

Comparative numerical results

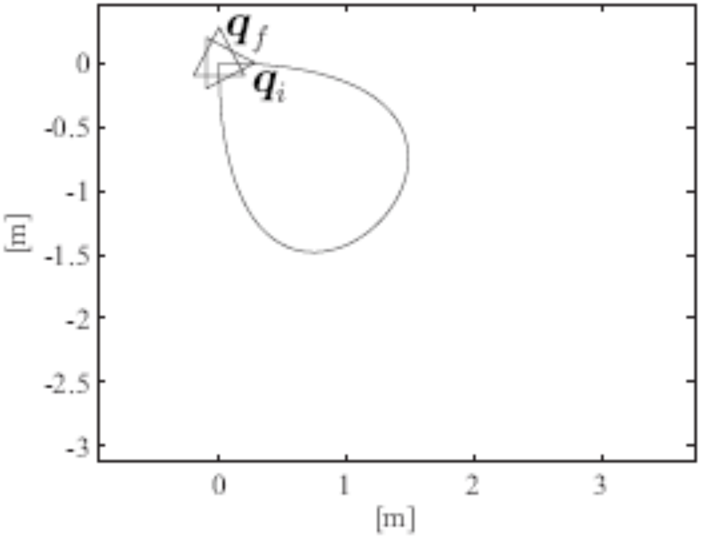


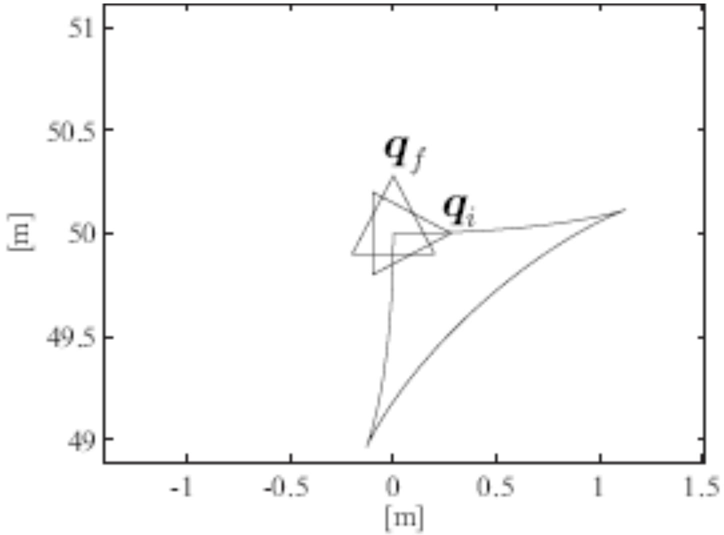
two different motion tasks

with Cartesian cubic polynomials
(K = 5)

using the chained form

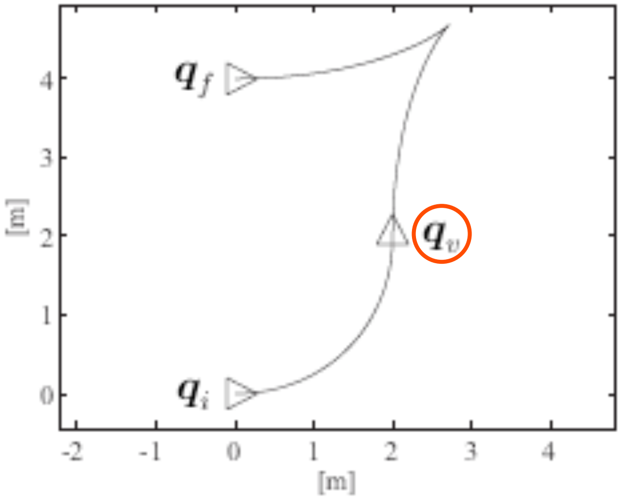
Comparative numerical results (cont'd)





task: a pure 90° reorientation

Cartesian cubic polynomials (K = 10) ⟷ using chained form



parallel parking with "via point" needed for planning in chained form (same initial and final orientations)

Motion control

- control schemes for unicycle-type WMR
  - posture regulation *(surprisingly, a more difficult problem here!)*
    - without loss of generality $(x_d,y_d,\theta_d) = (0,0,0)$, the origin
    1. based on a transformation in polar coordinates
    2. based on the exact linearization of the full kinematic model by means of dynamic feedback (DFL)
  - trajectory tracking *(of more practical interest …)*
    1. again with DFL (modifying method 2. for regulation)
    2. based on exact linearization/decoupling of the input-output map by means of static feedback (I-O SFL)
  - all control schemes use nonlinear feedback from WMR state

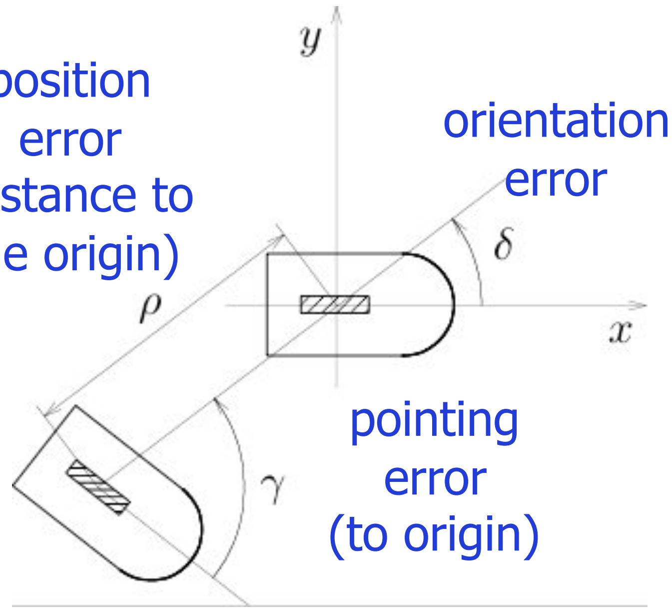
$$\boxed{\begin{aligned}\dot{x} &= v\cos\theta\\ \dot{y} &= v\sin\theta\\ \dot{\theta} &= \omega\end{aligned}}$$

kinematic model
of a unicycle WMR

Regulation using polar coordinates

- coordinate transformation

$$\rho = \sqrt{x^2 + y^2}$$
$$\gamma = \mathrm{ATAN2}(y,\, x) - \theta + \pi$$
$$\delta = \gamma + \theta$$



- control law (with $k_1, k_2, k_3 > 0$)

$$v = k_1 \rho \cos\gamma$$
$$\omega = k_2 \gamma + k_1 \frac{\sin\gamma\cos\gamma}{\gamma}(\gamma + k_3 \delta)$$

- asymptotic convergence to zero of the error is proven using a Lyapunov-based analysis

Regulation using DFL

1. introduce a single state ξ in the controller, and command the WMR with the dynamic law

$$\dot{\xi} = u_1 \cos\theta + u_2 \sin\theta$$
$$v = \xi$$
$$\omega = \frac{u_2 \cos\theta - u_1 \sin\theta}{\xi}$$

2. coordinate transformation

$$z_1 = x$$
$$z_2 = y$$
$$z_3 = \dot{x} = \xi \cos\theta$$
$$z_4 = \dot{y} = \xi \sin\theta$$

3. the resulting system is linear: two decoupled double integrators

$$\ddot{z}_1 = u_1$$
$$\ddot{z}_2 = u_2$$

4. regulation by PD feedback

$$u_1 = -k_{p1} x - k_{d1} \dot{x}$$
$$u_2 = -k_{p2} y - k_{d2} \dot{y}$$

$$k_{pi} > 0,\, k_{di} > 0 \ (i = 1,\, 2)$$

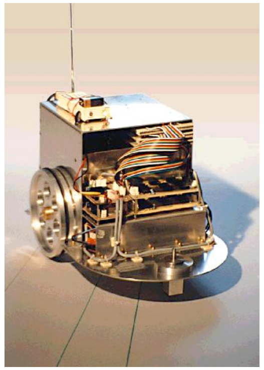
- exponential convergence of the error to zero provided we choose

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

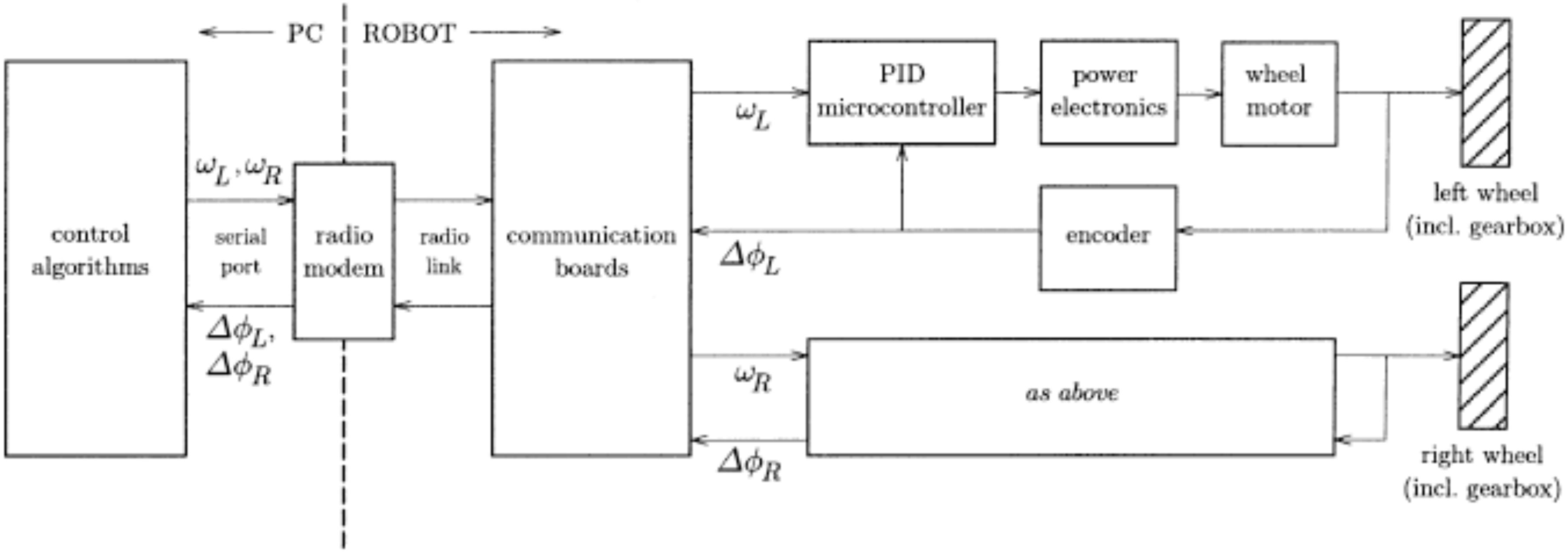
$$k_{d2} - k_{d1} > 2\sqrt{k_{d2}^2 - 4k_{p2}}$$

DFL = Dynamic Feedback Linearization

Control architecture

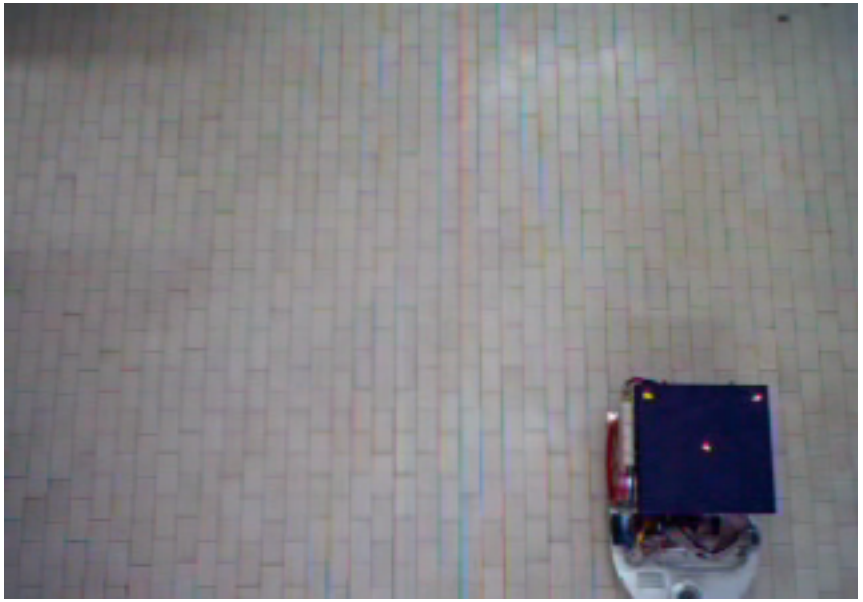


- SuperMARIO
  - with odometry (encoders on two wheels)
  - with external localization (from overhead camera)



“Parking” of SuperMARIO

- all maneuvers are driven by the feedback control using the current error with respect to the final desired configuration (posture regulation task, without planning!), processing the visual image (three LEDs seen by a camera fixed to the ceiling) for WMR localization



using polar coordinates

using dynamic linearization

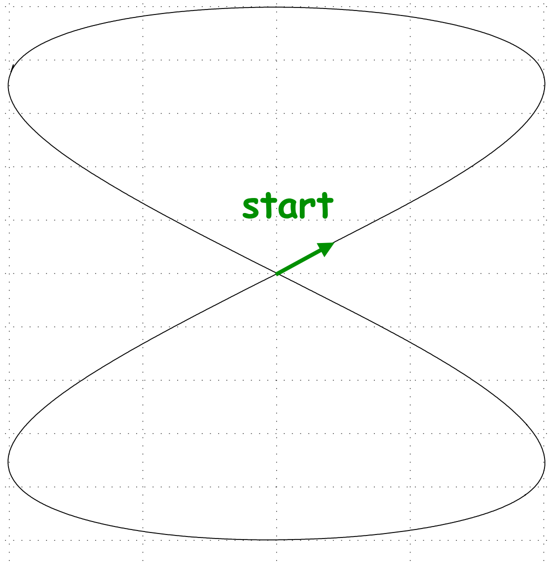
Trajectory tracking using DFL

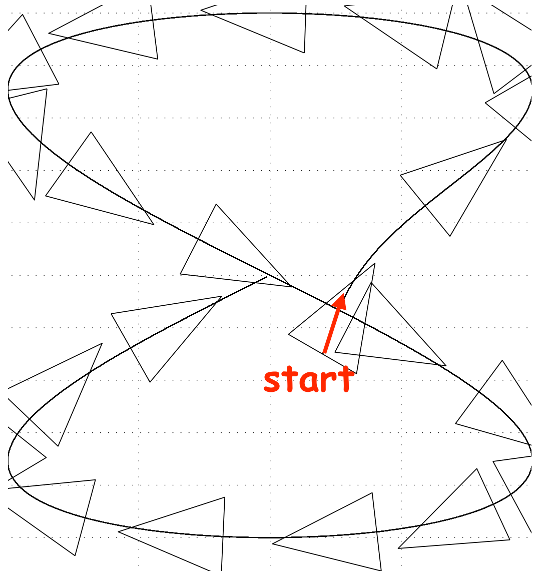
- based on dynamic feedback linearization, it is sufficient to modify step 4. stabilizing the tracking error with a PD + acceleration feedforward

$$u_1 = \ddot{x}_d + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x})$$
$$u_2 = \ddot{y}_d + k_{p2}(y_d - y) + k_{d2}(\dot{y}_d - \dot{y})$$

$$k_{pi} > 0,\ k_{di} > 0\ (i = 1,\ 2)$$

"historical" video of SuperMARIO



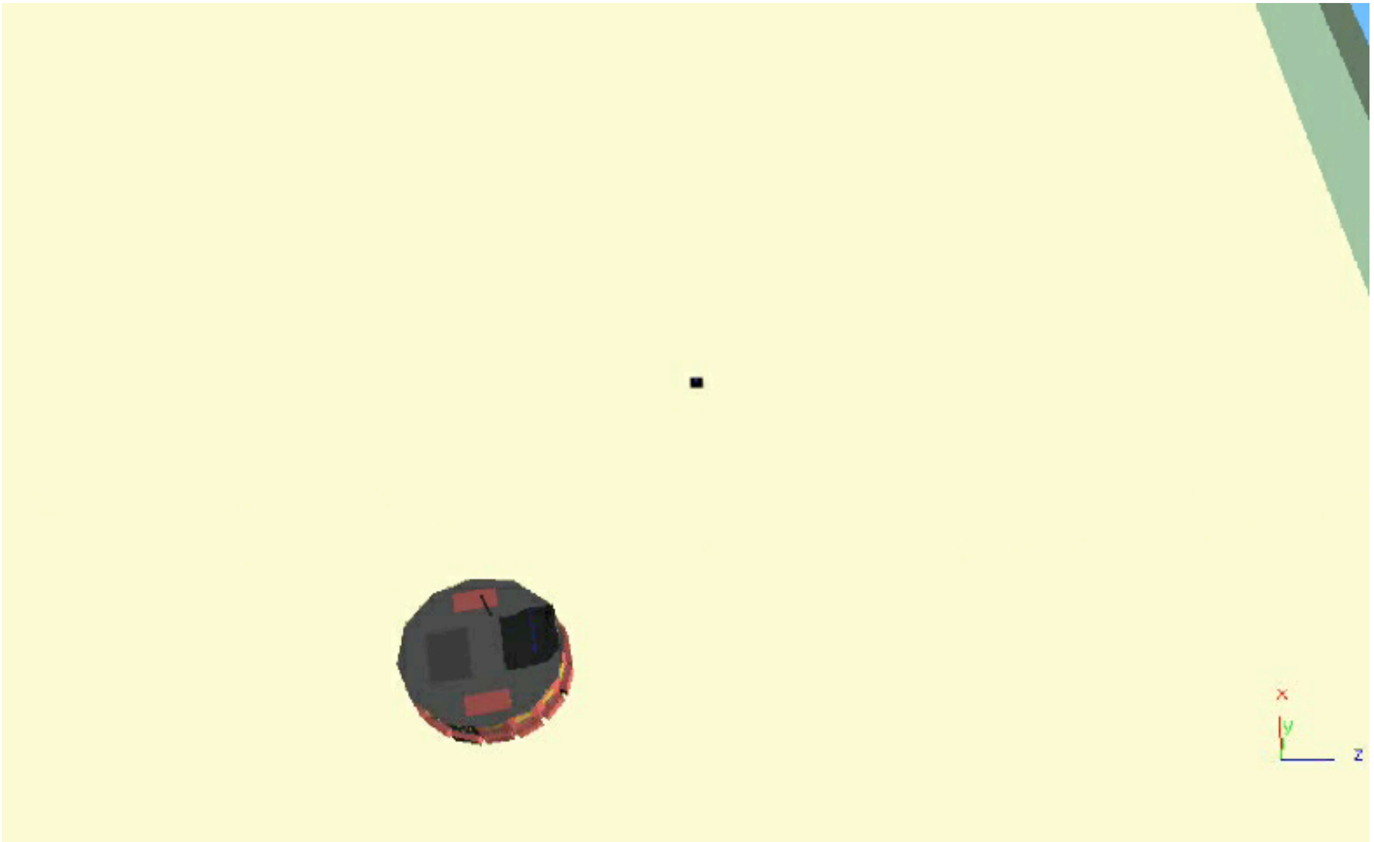


eight-shaped motion task with zero initial error

with initial error

Trajectory tracking using DFL

- simulation of Magellan Pro using Webot (control sampling at 32 msec)
- starting with an initial error w.r.t. desired trajectory

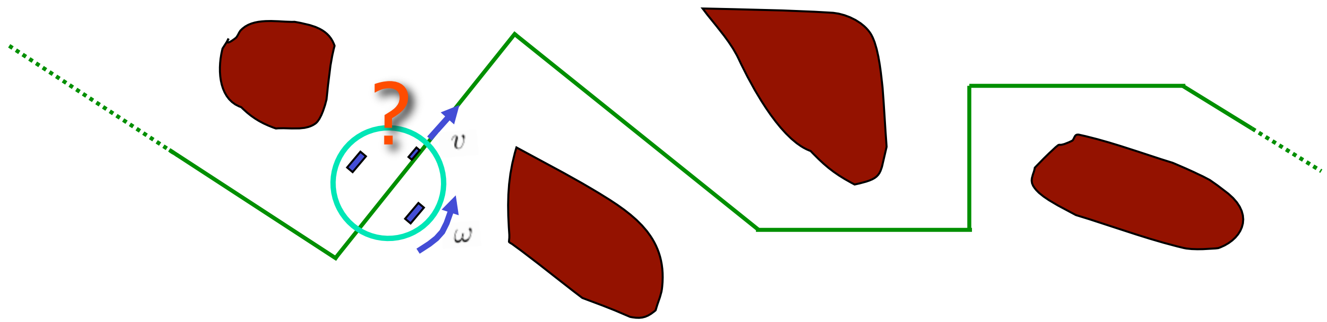


Comments on DFL method

- this control scheme is rather complex to implement
  - critical state initialization of the dynamic controller
  - problems at start and stop (when linear velocity vanishes)
- needs reference trajectories with sufficient smoothness
  - planner must generate path with continuous curvature
  - at possible discontinuities of the curvature (or even of the tangent), trajectory tracking is temporary lost, unless the robot stops (due to the choice of the timing law)
- other control laws are available for trajectory tracking, based on nominal feedforward + (linear or nonlinear) feedback regulation of the trajectory error
  - each has some operative restrictions: feasibility of the nominal trajectory (thus paths with continuous tangent) and/or persistency of motion trajectory, with small errors (local validity of linear feedback)

Motivation
for trajectory control method based on I-O SFL

is it possible for a unicycle to follow exactly and with a constant velocity a path that has discontinuous tangent?



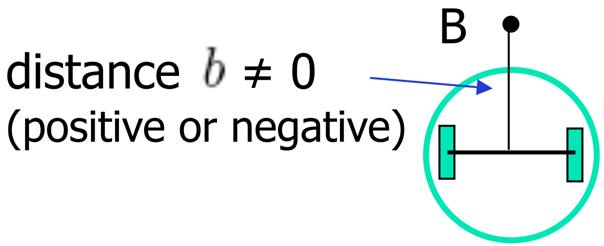
- a correct answer depends on the point taken as reference on the robot (system output), the point which should execute the desired motion
  - for example, the axle center point (x,y) can never have a “lateral” velocity with respect to vehicle orientation (thus, the answer in this case is no)
  - same for any other point along the (common) wheel axis …

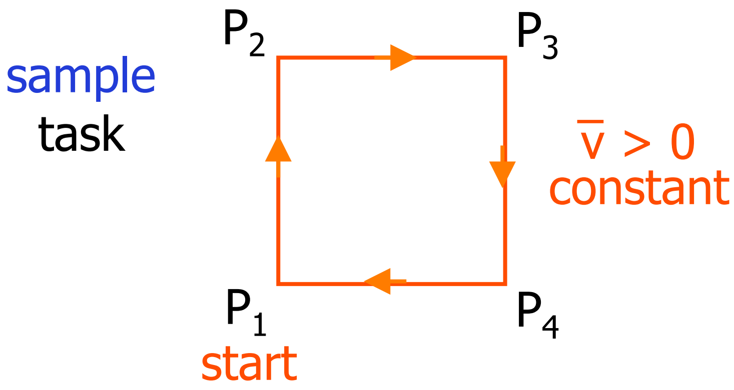
Trajectory tracking using I-O SFL

- however, by taking as system output a point B out of the wheel axle namely of coordinates

$$x_B = x + b\cos\theta \qquad y_B = y + b\sin\theta$$

it is possible to control the WMR motion so that point B will execute even paths with discontinuous tangent with a linear speed always different from zero (e.g., constant)





I-O SFL = Input-Output Static Feedback Linearization

Input-output exact linearization

1. with the coordinate transformation $(x, y, \theta) \rightarrow (x_B, y_B, \theta)$ one has

$$\begin{aligned}\dot{x}_B &= v\cos\theta - \omega b\sin\theta\\ \dot{y}_B &= v\sin\theta + \omega b\cos\theta\\ \dot{\theta} &= \omega\end{aligned}$$

2. in the first two equations, the dependence on inputs is invertible

$$\det\begin{bmatrix}\cos\theta & -b\sin\theta\\ \sin\theta & b\cos\theta\end{bmatrix} = b \neq 0$$

3. defining the static control law (in terms of two new inputs)

$$\begin{bmatrix}v\\ \omega\end{bmatrix} = \begin{bmatrix}\cos\theta & -b\sin\theta\\ \sin\theta & b\cos\theta\end{bmatrix}^{-1}\begin{bmatrix}v_{\mathrm{dx}}\\ v_{\mathrm{dy}}\end{bmatrix} = \begin{bmatrix}v_{\mathrm{dx}}\cos\theta + v_{\mathrm{dy}}\sin\theta\\ \frac{1}{b}(v_{\mathrm{dy}}\cos\theta - v_{\mathrm{dx}}\sin\theta)\end{bmatrix}$$

leads to a linear and decoupled (the input-output channels) system

$$\begin{aligned}\longleftarrow\quad \dot{x}_B &= v_{\mathrm{dx}} \quad\longleftarrow\quad \text{x direction}\\ \longleftarrow\quad \dot{y}_B &= v_{\mathrm{dy}} \quad\longleftarrow\quad \text{y direction}\\ \dot{\theta} &= \frac{1}{b}(v_{\mathrm{dy}}\cos\theta - v_{\mathrm{dx}}\sin\theta)\end{aligned}$$

Input-output exact linearization (cont'd)

4. for initial WMR conditions "matched" with the desired trajectory, the following choices of the auxiliary inputs allow perfect execution of the "square path with constant speed"

for the given sample task

$$
\begin{array}{llll}
P_1 \to P_2 & : & v_{\mathrm{dx}} = 0 & v_{\mathrm{dy}} = \bar{v} \\
P_2 \to P_3 & : & v_{\mathrm{dx}} = \bar{v} & v_{\mathrm{dy}} = 0 \\
P_3 \to P_4 & : & v_{\mathrm{dx}} = 0 & v_{\mathrm{dy}} = -\bar{v} \\
P_4 \to P_1 & : & v_{\mathrm{dx}} = -\bar{v} & v_{\mathrm{dy}} = 0
\end{array}
$$

5. to handle an initial error (or arising at any time), a feedback term is added proportional to the (position) trajectory error

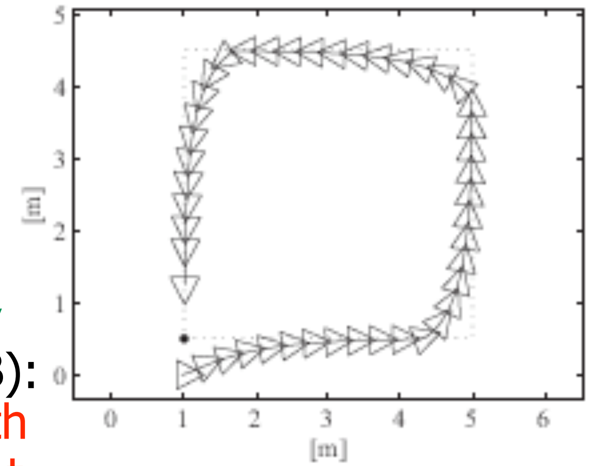
$$e = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x_{Bd}(t) - x_B \\ y_{Bd}(t) - y_B \end{bmatrix} \qquad \begin{bmatrix} v_{\mathrm{dx}} \\ v_{\mathrm{dy}} \end{bmatrix} \xrightarrow{k_x, k_y > 0} \begin{bmatrix} v_{\mathrm{dx}} + k_x e_x \\ v_{\mathrm{dy}} + k_y e_y \end{bmatrix}$$

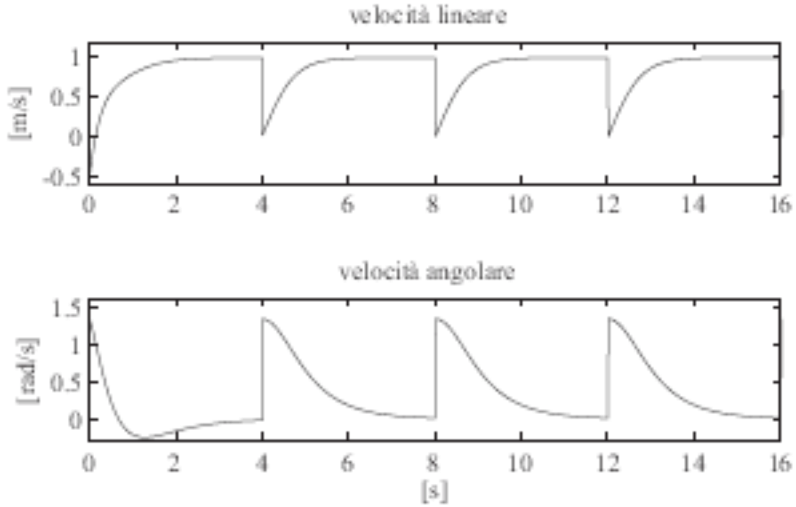
⇒ the error converges exponentially to zero, in an independent way for each Cartesian component

$$\dot{e}_x = -k_x e_x \quad \dot{e}_y = -k_y e_y$$

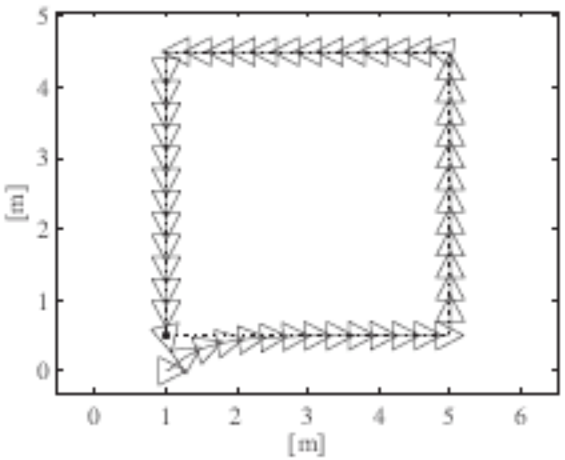
Simulation results

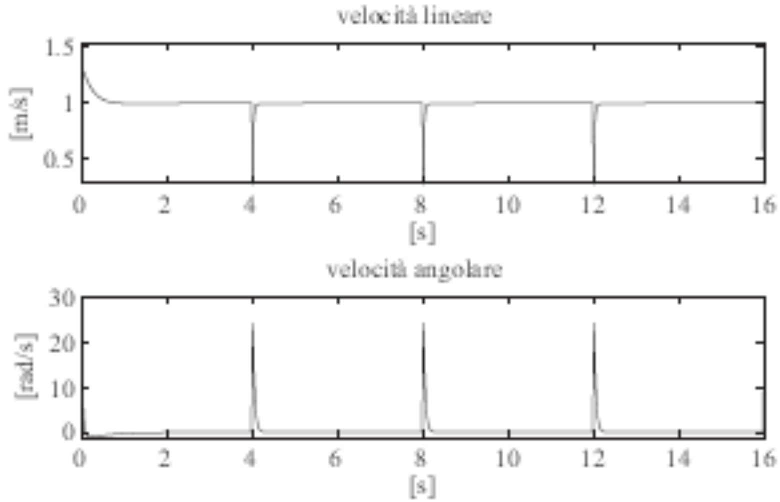
desired trajectory (for point B): square path executed at v = 1 m/s (with initial position error)





b = 0.75



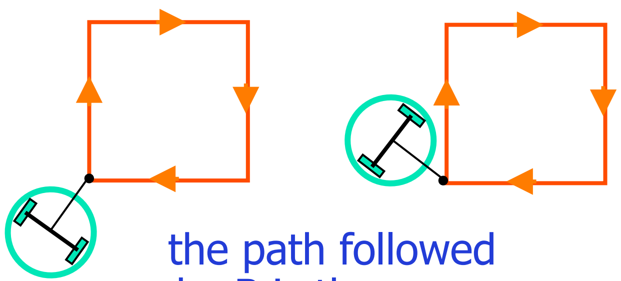


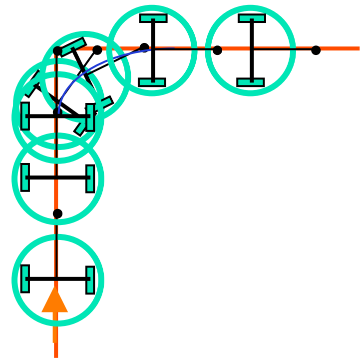
b = 0.2

… excessive peaks on ω

Comments on I-O SFL method

- exact reproduction of the desired trajectory for the off-set point B is independent from initial WMR orientation
  - (x,y) point "rounds off" the tangent discontinuities on the path





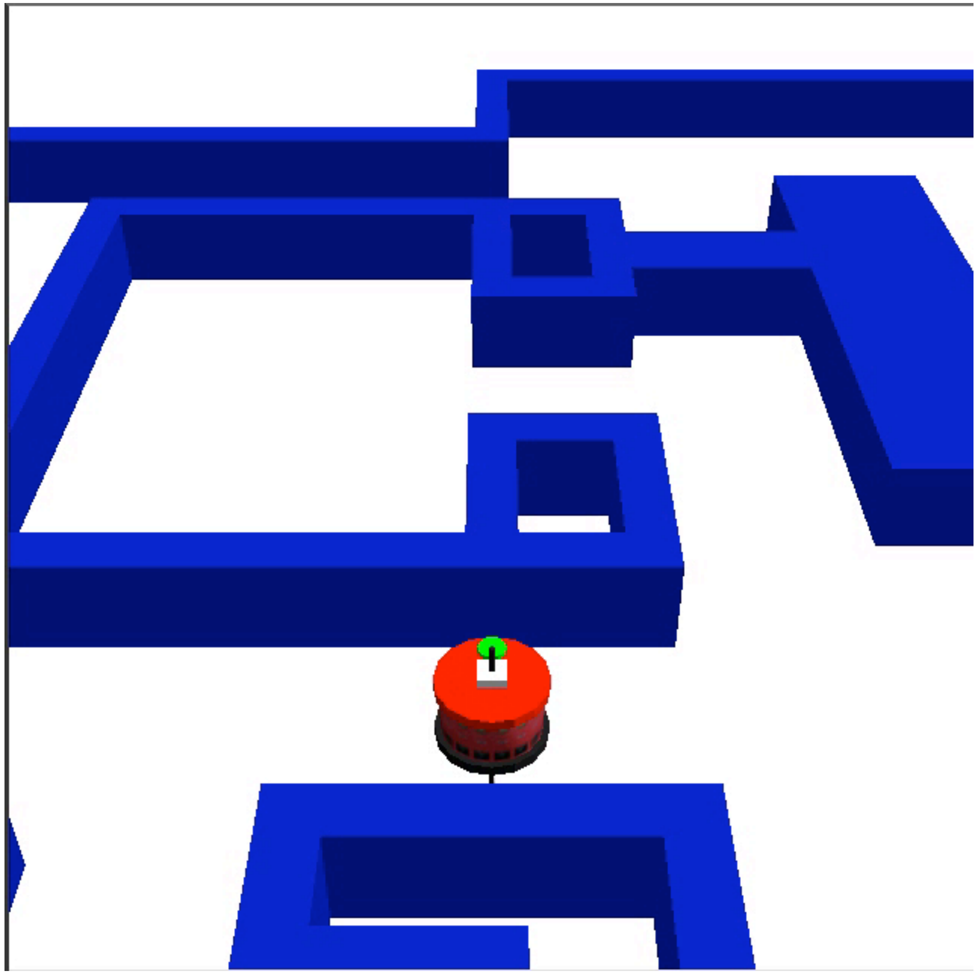
- this control scheme provides a general solution to the trajectory tracking problem for the unicycle
  - a suitable free channel is needed around the robot to account for the area "swept" by the vehicle during direction changes
  - when choosing b<0, the preferred motion is "backward"

Further interesting topics …

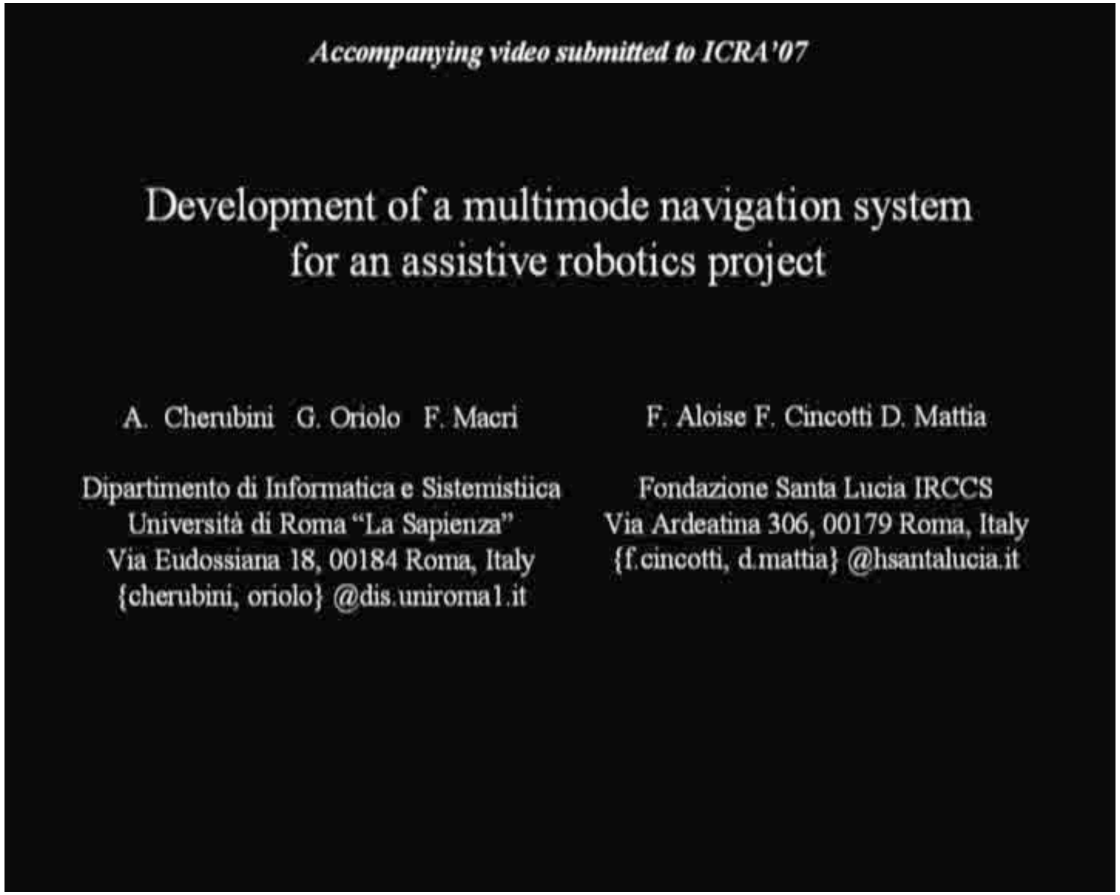
- planning in the presence of bounds/constraints
  - bounded velocity inputs
  - limited steering angle (car-like)
  - minimum length paths and minimum time trajectories
  - presence of obstacles
- navigation
  - WMR localization in the (known) environment
  - build an environmental map using exteroceptive sensors
  - SLAM = Simultaneous Localization And Mapping
  - exploration of unknown environments
- legged locomotion (two, four, or more)

Randomized exploration

Magellan
robot
with on-board
SICK laser scanner



Legged locomotion

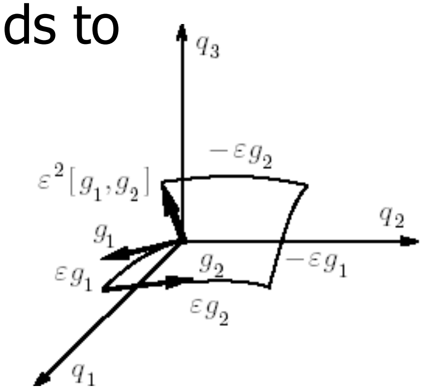


- ASPICE project (Telethon)
- SONY Aibo robot
- input commands via BCI = Brain Computer Interface

Appendix
Differential geometry - 1

- a (differentiable) vector field $f : I\!R^n \mapsto T_q I\!R^n$ is an application from any point in $I\!R^n$ to its tangent space $T_q I\!R^n$
- for a differential equation $\dot{q} = f(q)$, the flow $\phi(q)_t^f$ of the vector field f is the application that associates to each q the "solution" starting from this point: $\frac{d}{dt}\phi_t^f(q) = f(\phi_t^f(q))$
- the flow has the group property $\phi_t^f \circ \phi_s^f = \phi(q)_{t+s}^f$
- in linear dynamic systems, $f(q) = Aq$, the flow is $\phi(q)_t^f = e^{At}$
- starting from q_0, an infinitesimal flow by a time ε along g_1, then along g_2, then along $-g_1$, and finally along $-g_2$, leads to

$$\begin{aligned} q(4\epsilon) &= \phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(q_0) \\ &= q_0 + \epsilon^2 \underbrace{\left(\frac{\partial g_2}{\partial q} g_1(q_0) - \frac{\partial g_1}{\partial q} g_2(q_0)\right)}_{\text{Lie bracket of the two vector fields!}} + O(\epsilon^3) \end{aligned}$$



Appendix
Differential geometry - 2

- properties of Lie bracket operation
  1) $[\alpha f + \beta f', g] = \alpha[f, g] + \beta[f', g], \quad \forall \alpha, \beta \in \mathbb{R}$
  2) $[f, g] = -[g, f]$
  3) $[f, [g, h]] + [h, [f, g]] + [g, [h, f]] = 0$ (Jacobi identity)
- a distribution Δ associated to a set of differentiable vector fields $\{g_1, \ldots, g_m\}$ assigns to each point q a subspace of its tangent space

$$\begin{array}{rcl} \Delta & = & \text{span}\{g_1, \ldots, g_m\} \\ & \Updownarrow & \\ \Delta_q & = & \text{span}\{g_1(q), \ldots, g_m(q)\} \subset T_q\mathbb{R}^n \end{array}$$

- a distribution is regular if $\dim \Delta_q = \text{cost}, \forall q$ ("cost" is its dimension)
- the set of differentiable vector fields on $\mathbb{R}^n$ equipped with the Lie bracket operation is a Lie algebra
- a distribution Δ is involutive if it is closed under the Lie bracket operation $[g_i, g_j] \in \Delta \quad \forall g_i, g_j \in \Delta$