Control of Mobile Robots

Trajectory tracking

Prof. Luca Bascetta (luca.bascetta@polimi.it)

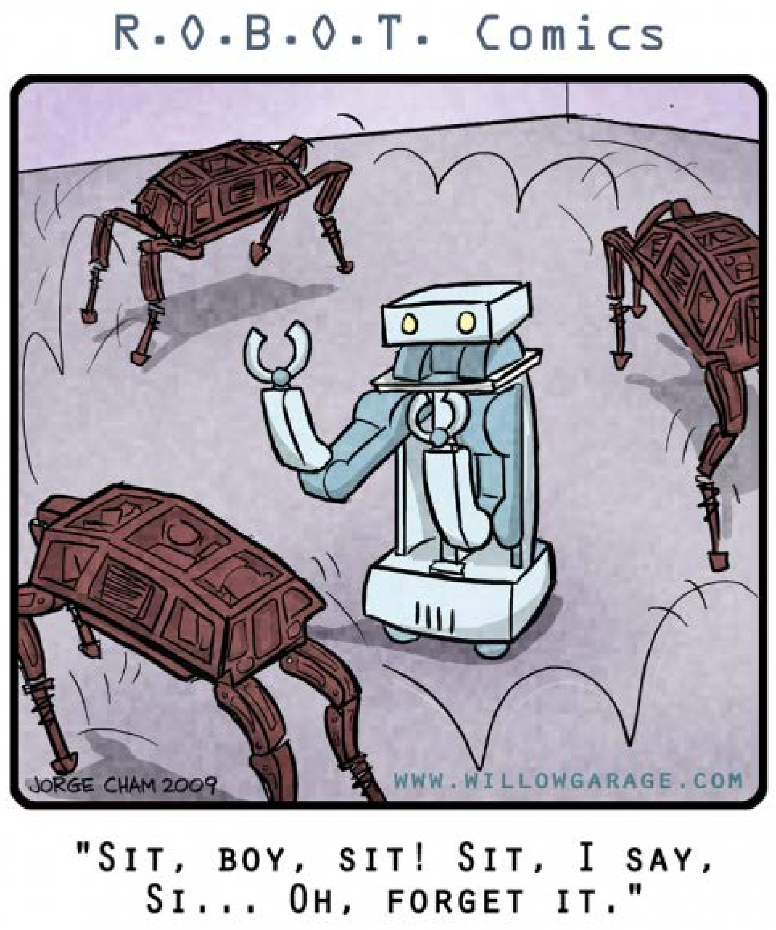
Politecnico di Milano – Dipartimento di Elettronica, Informazione e Bioingegneria

Organization and motivations

We complete the navigation part introducing the control problem and control techniques for mobile robots.

The main topics on trajectory tracking are

- control of industrial/mobile robots
- control of omnidirectional robots
- canonical nonholonomic mobile robot model
- trajectory tracking controller based on canonical model
- exact linearization
- trajectory tracking controller based on exact linearization
- fundamental of odometric localization



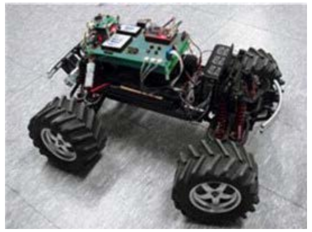
Control of industrial and mobile robots: differences and similarities

Control of industrial robots:

- a unique model to represent different industrial robots
- different, generic, well-known control approaches (IJC, computed-torque, …)

Control of mobile robots:

- different models for different robots characterized by different kinematic constraints (omnidirectional, unicycle, bicycle, …)
- no generic well-known control approach, different control approaches customized for different kinematics



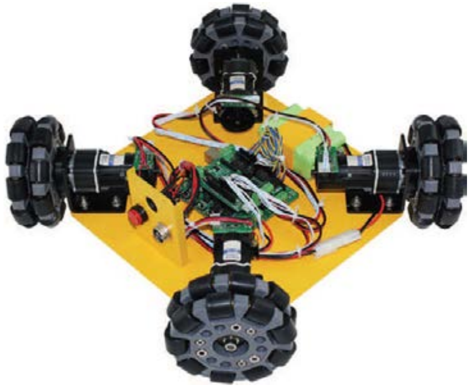
Omnidirectional robots are not subjected to nonholonomic constraints, a generic kinematic model and a generic control approach can be thus derived.

For an omnidirectional robot with more than 3 wheels a mapping between the chassis velocities and the wheel velocity vector can be introduced

$$\omega = H(\theta)\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}$$

As the robot is free to move in any direction, it can be modelled as three independent integrators

$$\dot{x} = v_x \qquad \dot{y} = v_y \qquad \dot{\theta} = v_\theta$$



Given a desired trajectory $x_d(t)$, $y_d(t)$, $\theta_d(t)$ each integrator can be controlled with a PI plus feedforward

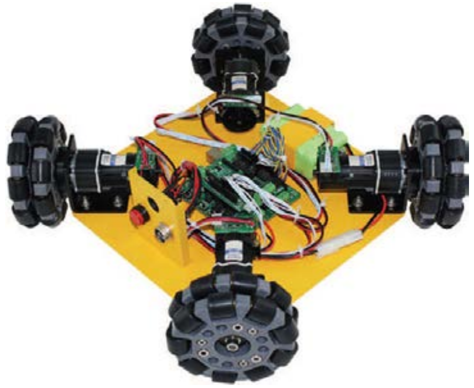
$$v_x = \dot{x}_d(t) + K_{P_x}\left(x_d(t) - x(t)\right) + K_{I_x}\int_0^t \left(x_d(\tau) - x(\tau)\right)\mathrm{d}\tau$$

$$v_y = \dot{y}_d(t) + K_{P_y}\left(y_d(t) - y(t)\right) + K_{I_y}\int_0^t \left(y_d(\tau) - y(\tau)\right)\mathrm{d}\tau$$

$$v_\theta = \dot{\theta}_d(t) + K_{P_\theta}\left(\theta_d(t) - \theta(t)\right) + K_{I_\theta}\int_0^t \left(\theta_d(\tau) - \theta(\tau)\right)\mathrm{d}\tau$$

Then wheel velocities can be computed applying the mapping

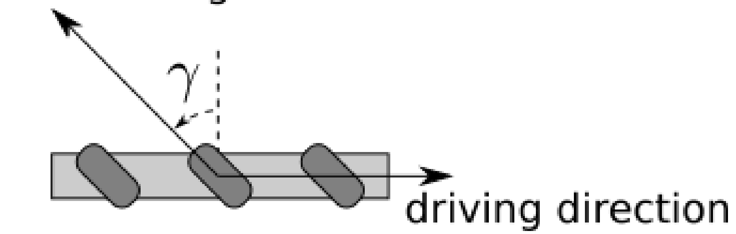
$$\omega = H(\theta)\begin{bmatrix} v_x \\ v_y \\ v_\theta \end{bmatrix}$$



Computing mapping $H(\theta)$ for an omnidirectional robot

Let's derive the mapping $H(\theta)$ for an omnidirectional robot with four mecanum wheels.

A mecanum wheel is characterized by a driving direction and a free sliding direction, usually with $\gamma = \pm 45°$.



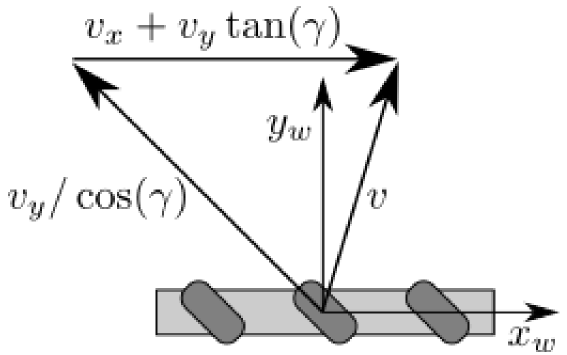
The linear velocity of the center of the wheel with respect to the wheel frame is given by

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_{drive} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v_{slide} \begin{bmatrix} -\sin(\gamma) \\ \cos(\gamma) \end{bmatrix}$$

solving this equation

$$v_{drive} = v_x + v_y \tan(\gamma)$$

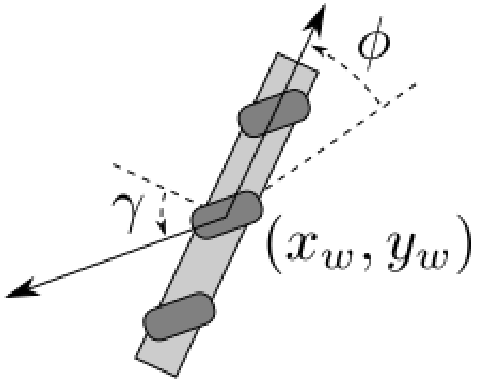
$$v_{slide} = \frac{v_y}{\cos(\gamma)}$$



Computing mapping $H(\theta)$ for an omnidirectional robot

If r is the radius of the wheel, its rotational velocity is given by

$$\omega = \frac{v_{drive}}{r} = \frac{v_x + v_y \tan(\gamma)}{r}$$

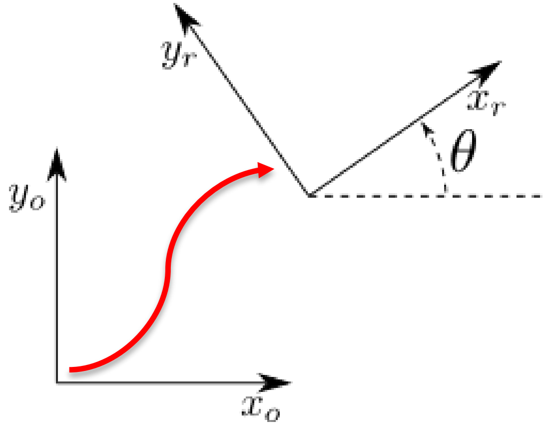


To derive the full mapping $H(\theta)$ we must refer v_x and v_y to an absolute reference frame

$$\omega = \begin{bmatrix} \frac{1}{r} & \frac{tan(\gamma)}{r} \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} -y_w & 1 & 0 \\ x_w & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

Converts the linear velocity to the wheel frame

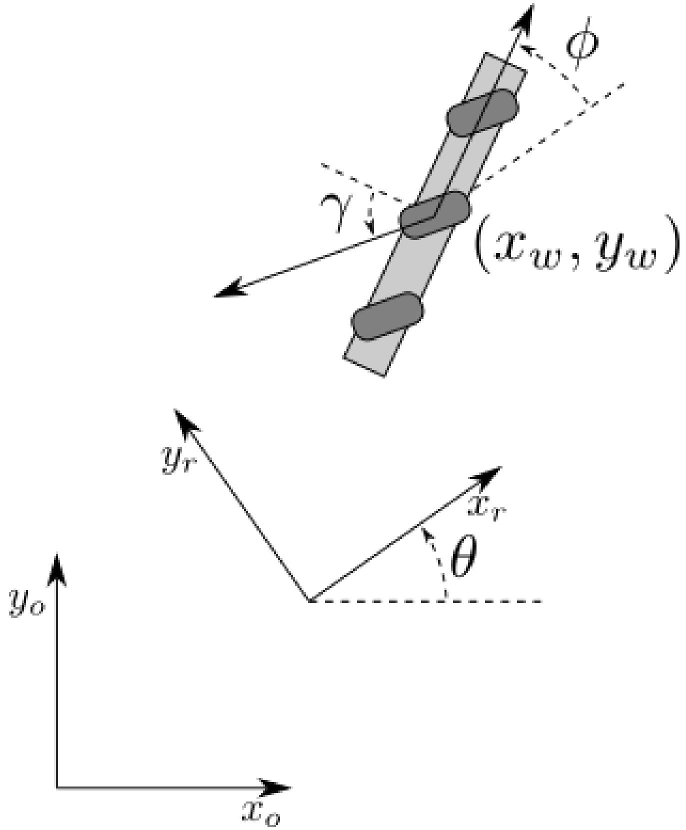
Produces the linear velocity at the wheel wrt frame *r*



Computing mapping $H(\theta)$ for an omnidirectional robot

$$\omega = \begin{bmatrix} \frac{1}{r} & \frac{tan(\gamma)}{r} \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} -y_w & 1 & 0 \\ x_w & 0 & 1 \end{bmatrix} \cdot$$

$$\cdot \begin{bmatrix} 0 & 0 & 1 \\ \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$



and evaluating

$$H(\theta) = \frac{1}{r\cos(\gamma)} \begin{bmatrix} x_w \sin(\phi+\gamma) - y_w \cos(\phi+\gamma) \\ \cos(\phi+\gamma+\theta) \\ \sin(\phi+\gamma+\theta) \end{bmatrix}^T$$

A canonical simplified model for nonholonomic mobile robots

When nonholonomic constraints are introduced, the control problem becomes far more complex, and no generic control approach exists.

We can try, however, to introduce a *canonical simplified model for nonholonomic mobile robots*.

Let's first recap the kinematic models we have already introduced.

The unicycle kinematic model

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \omega \end{cases}$$

The differential drive kinematic model

$$\begin{cases} \dot{x} = \dfrac{\omega_R + \omega_L}{2} r \cos(\theta) \\ \dot{y} = \dfrac{\omega_R + \omega_L}{2} r \sin(\theta) \\ \dot{\theta} = \dfrac{\omega_R - \omega_L}{d} r \end{cases}$$

The rear-wheel drive bicycle

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \tan(\phi)/\ell \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

We already know that the unicycle and the differential drive are equivalent

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \omega \end{cases}$$

$$\begin{cases} \dot{x} = \dfrac{\omega_R + \omega_L}{2} r\cos(\theta) \\ \dot{y} = \dfrac{\omega_R + \omega_L}{2} r\sin(\theta) \\ \dot{\theta} = \dfrac{\omega_R - \omega_L}{d} r \end{cases}$$

setting

$$v = \frac{\omega_R + \omega_L}{2} r \qquad \omega = \frac{\omega_R - \omega_L}{d} r$$

and for the bicycle model?

A canonical simplified model for nonholonomic mobile robots

Assuming that the steering rate limit is so high that the steering angle can be changed instantaneously, we can simplify the bicycle model as

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = v\dfrac{\tan(\phi)}{\ell} \end{cases}$$

and setting

$$v = v \qquad \omega = v\frac{\tan(\phi)}{\ell}$$

the bicycle is equivalent to the unicycle.

A canonical simplified model for nonholonomic mobile robots

Summarizing, the differential drive model can be reduced to the unicycle model applying the following transformation

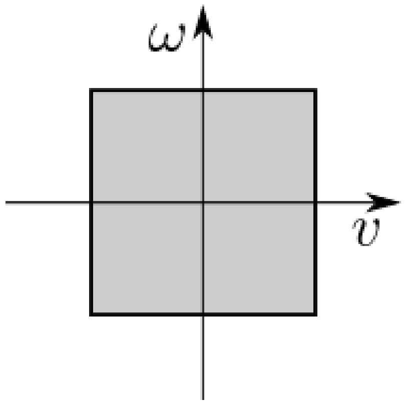
$$\omega_R = \frac{v + \omega\, d/2}{r} \qquad \omega_L = \frac{v - \omega\, d/2}{r}$$

and the bicycle model can be reduced to the unicycle model applying the following transformation

$$v = v \qquad \phi = \arctan\left(\frac{\dot{\theta}\ell}{v}\right)$$

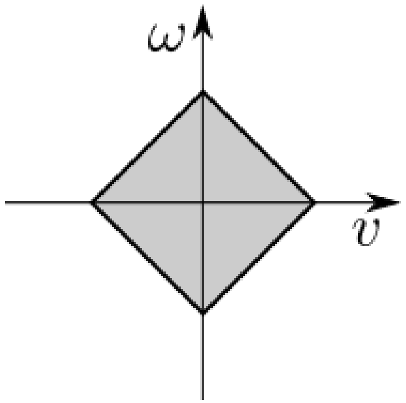
Under these transformations the differential drive and the bicycle are equivalent to the unicycle, the only difference is the mapping of the control limits from the original inputs to v and ω.

A canonical simplified model for nonholonomic mobile robots



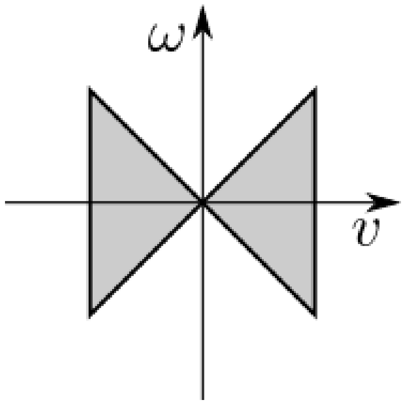
unicycle

$$\omega_m \leq \omega \leq \omega_M$$
$$v_m \leq v \leq v_M$$



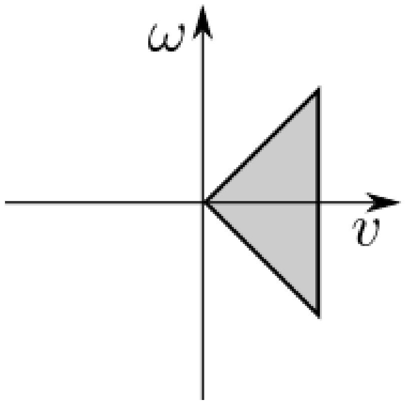
differential drive

$$\omega_{R_m} \leq \omega_R \leq \omega_{R_M}$$
$$\omega_{L_m} \leq \omega_L \leq \omega_{L_M}$$



bicycle with reverse

$$\delta_m \leq \delta \leq \delta_M$$
$$v_m \leq v \leq v_M$$



bicycle without reverse

$$\delta_m \leq \delta \leq \delta_M$$
$$0 \leq v \leq v_M$$

Formalizing a control problem using the canonical simplified model

Consider the design of the trajectory tracking controller for a differential drive robot, using the canonical simplified model. The actuators of the differential drive are characterized by the following limitation

$$|\omega_R| \leq \bar{\omega} \qquad |\omega_L| \leq \bar{\omega}$$

How should we limit the linear and angular velocity of the canonical model, in order to be consistent with the actuator limitations?

Assuming that r is the wheel radius and d the distance between the wheel contact points, the relations between the differential drive model and the canonical simplified model are

$$\omega_R = \frac{v + \omega\, d/2}{r} \qquad \omega_L = \frac{v - \omega\, d/2}{r}$$

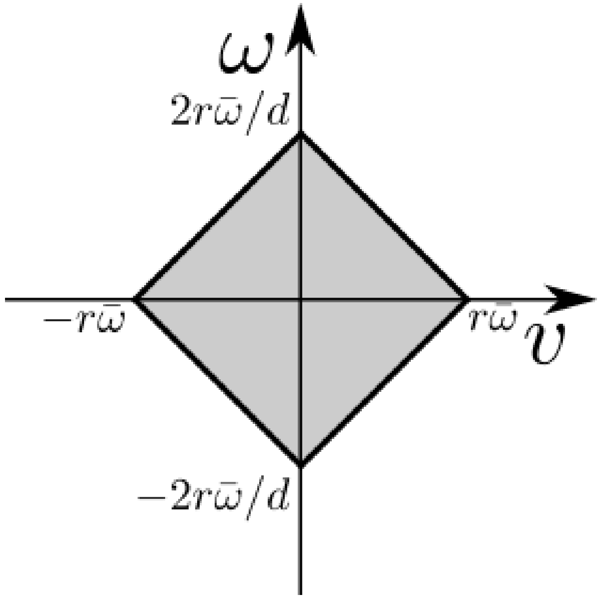
using these relations the two limitations on the actuators can be rewritten as

Formalizing a control problem using the canonical simplified model

$$\left|\frac{1}{r}v + \frac{d}{2r}\omega\right| \le \bar{\omega} \qquad \left|\frac{1}{r}v - \frac{d}{2r}\omega\right| \le \bar{\omega}$$

These two inequalities are equivalent to

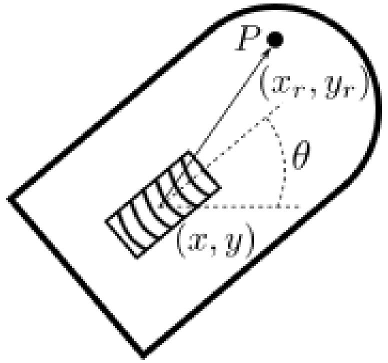
$$\begin{aligned}
\omega &\le -\frac{2}{d}v + 2\frac{r\bar{\omega}}{d} && \text{for} \quad \omega \ge -\frac{2}{d}v \\
\omega &\ge -\frac{2}{d}v - 2\frac{r\bar{\omega}}{d} && \text{for} \quad \omega < -\frac{2}{d}v \\
\omega &\ge \frac{2}{d}v - 2\frac{r\bar{\omega}}{d} && \text{for} \quad \omega \le \frac{2}{d}v \\
\omega &\le \frac{2}{d}v + 2\frac{r\bar{\omega}}{d} && \text{for} \quad \omega > \frac{2}{d}v
\end{aligned}$$



Control of the canonical simplified model

We can now introduce a control law for the canonical model.

Consider a point P on the robot chassis, its absolute coordinates are



$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_r \\ y_r \end{bmatrix}$$

and differentiating, we obtain

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \dot{\theta} \begin{bmatrix} -\sin(\theta) & -\cos(\theta) \\ \cos(\theta) & -\sin(\theta) \end{bmatrix} \begin{bmatrix} x_r \\ y_r \end{bmatrix}$$

The possible istantaneous velocities of P are not restricted by the nonholonomic constraint

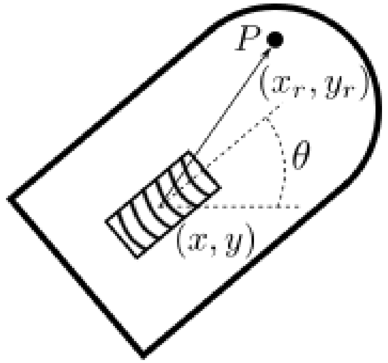
substituting the canonical model equations and solving for v and ω

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{x_r} \begin{bmatrix} x_r \cos(\theta) - y_r \sin(\theta) & x_r \sin(\theta) + y_r \cos(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix}$$

Control of the canonical simplified model

Thanks to the trasformation

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{x_r} \begin{bmatrix} x_r \cos(\theta) - y_r \sin(\theta) & x_r \sin(\theta) + y_r \cos(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix}$$



the canonical model can be reduced to

$$\dot{x}_P = v_{x_P} \qquad \dot{y}_P = v_{y_P}$$

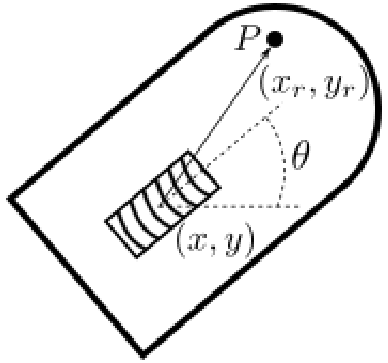
the same linear model we have considered for the omnidirectional robot.

Caveat: this transformation is singular when $x_r = 0$.

Control of the canonical simplified model

The linear model

$$\dot{x}_P = v_{x_P} \qquad \dot{y}_P = v_{y_P}$$



is completely controllable.

It can be thus stabilized by a linear proportional controller: a proportional controller exists

$$v_{x_P} = -K_x x_P \qquad v_{y_P} = -K_y y_P$$

that stabilizes the origin of the system in the (x_P, y_P) coordinates.

But there is no linear controller able to stabilize the chassis configuration in the (x, y, θ) coordinates. This fact is embedded in a well-known theorem:

« A system $\dot{\boldsymbol{q}} = G(\boldsymbol{q})\boldsymbol{u}$ with $rank\big(G(\mathbf{0})\big) < dim(\boldsymbol{q})$ cannot be stabilized to $\boldsymbol{q} = 0$ by a continuous time-invariant feedback control law »

Control of the canonical simplified model

« A system $\dot{\boldsymbol{q}} = G(\boldsymbol{q})\boldsymbol{u}$ with $rank\big(G(\boldsymbol{0})\big) < dim(\boldsymbol{q})$ cannot be stabilized to $\boldsymbol{q} = 0$ by a continuous time-invariant feedback control law »

For the canonical simplified model we have

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

and the rank of $G(\boldsymbol{q})$ is 2 for any value of θ, while the configuration has dimension 3. The theorem applies to the canonical simplified model.

How should we interpret this result? Is the canonical simplified model controllable?

Controllability of the canonical simplified model

To answer these questions we need to introduce a notion of controllability for nonlinear systems.

Let's start from the definitions.

Given a time $T > 0$ and a neighborhood W of an initial configuration $\boldsymbol{q}$, we introduce $R^W(\boldsymbol{q}, T)$ the *reachable set* of configurations from $\boldsymbol{q}$ at time T by feasible trajectories remaining inside W.

The union of reachable sets at time $t \in (0, T]$ is given by

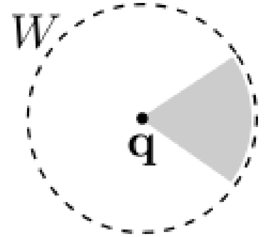
$$R^W(\mathbf{q}, \leq T) = \bigcup_{0 \leq t \leq T} R^W(\mathbf{q}, t)$$

We can now introduce a definition of controllability...

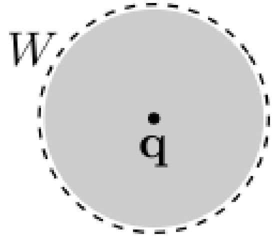
Controllability of the canonical simplified model

A robot is <u>*controllable*</u> from $\boldsymbol{q}$ if, for any $\boldsymbol{q}_{goal}$, there exists a control $u(t)$ that drives the robot from $\boldsymbol{q}$ to $\boldsymbol{q}_{goal}$ in finite time T.

A robot is <u>*small-time locally accessible*</u> (STLA) from $\boldsymbol{q}$ if, for any time $T > 0$ and any neighborhood W, the reachable set $R^W(\boldsymbol{q}, \leq T)$ is a full-dimensional subset of the configuration space.



A robot is <u>*small-time locally controllable*</u> (STLC) from $\boldsymbol{q}$ if, for any time $T > 0$ and any neighborhood W, the reachable set $R^W(\boldsymbol{q}, \leq T)$ is a neighborhood of $\boldsymbol{q}$.

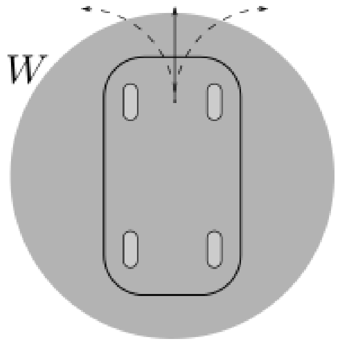


If a system is STLC at all $\boldsymbol{q}$, then it is controllable from any $\boldsymbol{q}$ by the patching together of paths in neighborhoods from $\boldsymbol{q}$ to $\boldsymbol{q}_{goal}$.

Controllability of a car-like robot

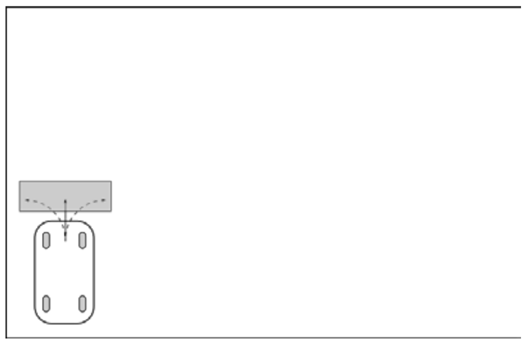
Let's consider some examples:

- a forward-only car is STLA but it is not STLC, if it is confined to a tight space it cannot reach configurations behind its initial configuration
- a car with reverse gear is STLC



In an obstacle-free environment even a forward-only car can drive anywhere.

If there are obstacles in the plane, there may be some free-space configurations that the forward-only car cannot reach.



Now that we know how controllability properties are defined we need a way to test if a system is controllable, STLA or STLC.

Controllability of the canonical simplified model

Given the kinematic model of a mobile robot

$$\dot{\mathbf{q}} = G(\mathbf{q})\,\mathbf{u} = \sum_{i=1}^{m} g_i(\mathbf{q})\,u_i$$

we can compute the accessibility distribution Δ_A, then the system is ***STLA*** from $\boldsymbol{q}$ if

$$\dim(\Delta_A(\mathbf{q})) = \dim(\mathbf{q}) \qquad \text{and} \qquad \operatorname{span}(\mathbf{U}) = \mathbb{R}^m$$

$$\operatorname{span}(\{v_i\}) = \left\{\sum k_i v_i,\ k_i \in \mathbb{R}\right\}$$

if additionally

$$\operatorname{pos}(\mathbf{U}) = \mathbb{R}^m$$

the system is ***STLC*** from $\boldsymbol{q}$.

$$\operatorname{pos}(\{v_i\}) = \left\{\sum k_i v_i,\ k_i \geq 0\right\}$$

Controllability of the canonical simplified model

We can interpret the previous result as follows.

If the accessibility distribution is full rank $\left(\dim\left(\Delta_A(\boldsymbol{q})\right) = \dim(\boldsymbol{q})\right)$, the vector fields locally allow motion in any direction.

Then:

- if $pos(\boldsymbol{U}) = \mathbb{R}^m$, as for a car with reverse gear, all vector fields can be followed forward or backward
- if $span(\boldsymbol{U}) = \mathbb{R}^m$, as for a forward-only car, some vector fields may be followed only forward or only backward

Let’s go back again to the canonical simplified model.

The two vector fields are

$$g_1(\mathbf{q}) = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} \qquad g_2(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and the Lie bracket

$$g_3(\mathbf{q}) = [g_1, g_2](\mathbf{q}) = \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \\ 0 \end{bmatrix}$$

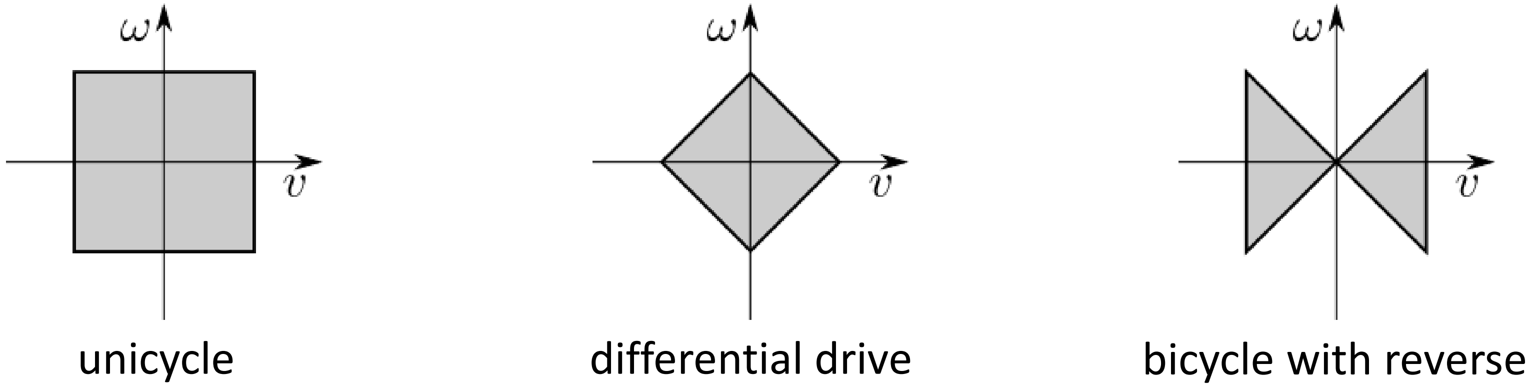
putting all these vectors together in a matrix and computing the determinant we obtain

Controllability of the canonical simplified model

$$\det\left(\left[g_1(\mathbf{q}) \quad g_2(\mathbf{q}) \quad g_3(\mathbf{q})\right]\right) = \det\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ \sin(\theta) & 0 & -\cos(\theta) \\ 0 & 1 & 0 \end{bmatrix} = \cos^2(\theta) + \sin^2(\theta) = 1$$

We can thus conclude that the three vector fields are independent at all $\boldsymbol{q}$, and the accessibility distribution has dimension 3.

Consider now the control sets

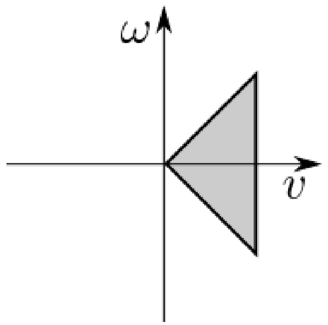


$$\mathrm{pos}(\mathbf{U}) = \mathbb{R}^2$$

STLC

Controllability of the canonical simplified model

and



bicycle without reverse

$$\text{span}\,(\mathbf{U}) = \mathbb{R}^2$$

STLA

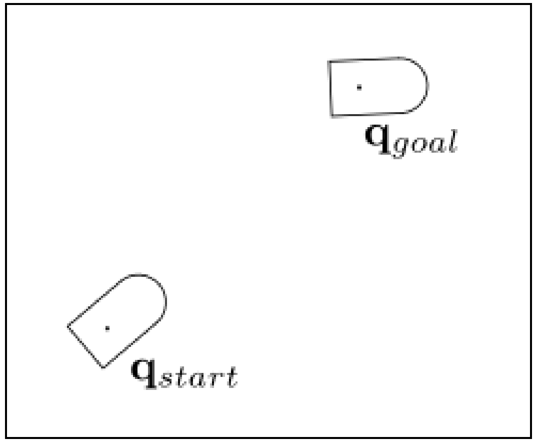
We can thus conclude that all these robots (unicycle, differential drive, bicycle with/without reverse) are controllable in an obstacle free plane.

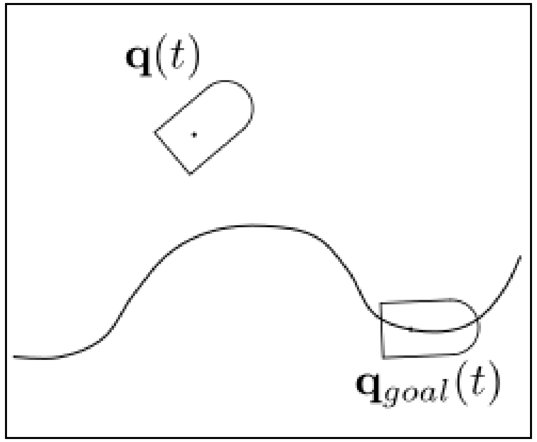
We can introduce three types of feedback control problems:

- regulation (stabilization of a configuration)
- trajectory tracking
- path tracking

A planning step is not required, but the Cartesian trajectory of the robot cannot be predicted

Forcing the robot to move along or close to a trajectory planned in advanced considerably reduces the risk of collisions





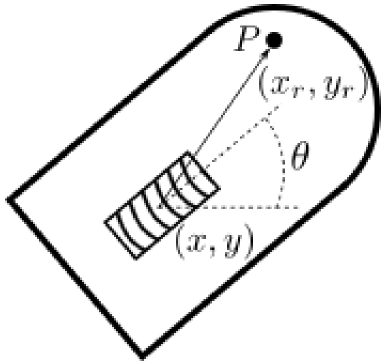
We consider the trajectory tracking problem.

Assume that a reference trajectory $\left(x_d(t), y_d(t), \theta_d(t)\right)$ is available, together with a corresponding nominal control $\left(v_d(t), \omega_d(t)\right)$ for $t \in [0, T]$.

Considering a reference point P on the chassis of the robot (not on the axis of the driving wheels), we can compute its reference trajectory $\left(x_{P_d}(t), y_{P_d}(t)\right)$.

A proportional controller can be introduced to track this reference trajectory

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix} = \begin{bmatrix} k_P\left(x_{P_d} - x_P\right) \\ k_P\left(y_{P_d} - y_P\right) \end{bmatrix}$$



where $k_P > 0$.

Canonical simplified model: trajectory tracking

Finally, applying equation

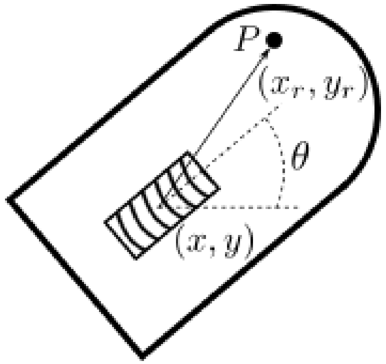
$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{x_r} \begin{bmatrix} x_r \cos(\theta) - y_r \sin(\theta) & x_r \sin(\theta) + y_r \cos(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix}$$

we can convert the velocity of point P into the robot velocity.

How does the robot move under this control law?

As long as the reference point is moving, the robot chassis lines up with the desired orientation θ_d.

The controller, however, can select the orientation θ_d or $-\theta_d$, and there is no way to prevent it selecting $-\theta_d$ and a negative velocity.

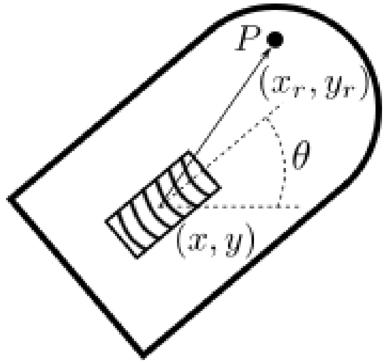


How to fix this? …introducing a nonlinear controller?

…or introducing a more general approach?

We have already seen that thanks to the transformation

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{x_r} \begin{bmatrix} x_r \cos(\theta) - y_r \sin(\theta) & x_r \sin(\theta) + y_r \cos(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix}$$



the canonical model can be reduced to

$$\dot{x}_P = v_{x_P} \qquad \dot{y}_P = v_{y_P}$$

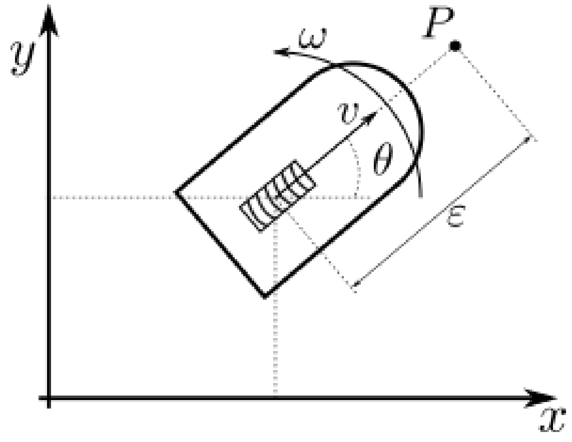
a linear model composed of two independent integrators.

This transformation is an example of a more generic control tool for nonlinear systems, that is called feedback linearization.

Feedback linearization allows to transform a nonlinear model into independent chains of integrators, it is thus a way to exactly linearize a nonlinear model.

We will now introduce examples of feedback linearization for classical robot models...

Exact linearization of a nonlinear system: the unicycle kinematic model



We start considering the unicycle model, making reference to the motion of point P

$$x_P = x + \varepsilon \cos(\theta)$$
$$y_P = y + \varepsilon \sin(\theta)$$

and differentiating with respect to time

$$\dot{x}_P = \dot{x} - \varepsilon \dot{\theta} \sin(\theta) = v \cos(\theta) - \varepsilon \dot{\theta} \sin(\theta) = v_{x_P}$$
$$\dot{y}_P = \dot{y} + \varepsilon \dot{\theta} \cos(\theta) = v \sin(\theta) + \varepsilon \dot{\theta} \cos(\theta) = v_{y_P}$$

We can now apply a trick to derive the change of variable.

Exact linearization of a nonlinear system: the unicycle kinematic model

Multiplying the two equations by $\cos(\theta)$ / $\sin(\theta)$ and summing them together

$$
\begin{array}{rcl}
v\cos^2(\theta) - \varepsilon\omega\sin(\theta)\cos(\theta) & = & v_{xP}\cos(\theta) \\
v\sin^2(\theta) + \varepsilon\omega\cos(\theta)\sin(\theta) & = & v_{yP}\sin(\theta) \\
\hline
v & = & v_{xP}\cos(\theta) + v_{yP}\sin(\theta)
\end{array}
$$

Multiplying the two equations by $\sin(\theta)$ / $\cos(\theta)$ and subtracting them together

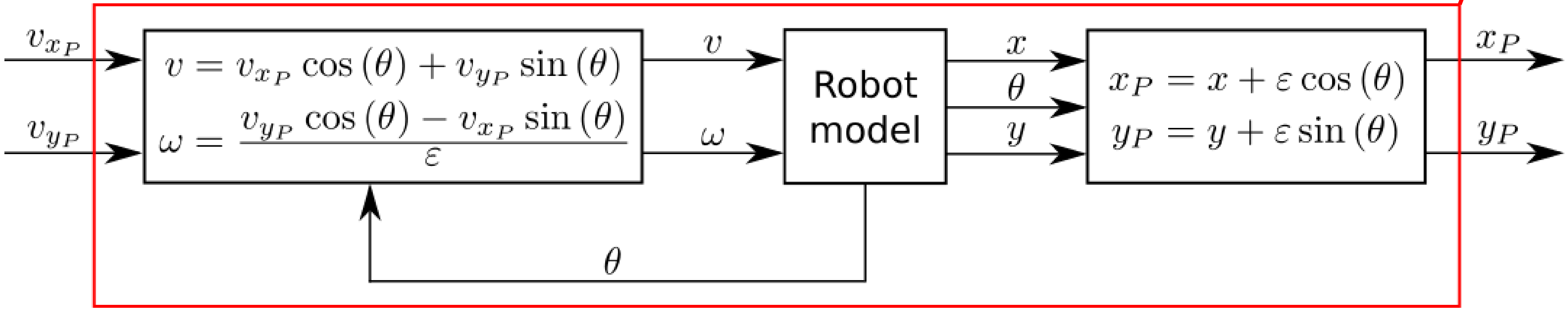
$$
\begin{array}{rcl}
v\cos(\theta)\sin(\theta) - \varepsilon\omega\sin^2(\theta) & = & v_{xP}\sin(\theta) \\
v\sin(\theta)\cos(\theta) + \varepsilon\omega\cos^2(\theta) & = & v_{yP}\cos(\theta) \\
\hline
\varepsilon\omega & = & v_{yP}\cos(\theta) - v_{xP}\sin(\theta)
\end{array}
$$

Summarizing, we found the two change of coordinates

$$v = v_{xp} \cos(\theta) + v_{yp} \sin(\theta)$$

$$\omega = \frac{v_{yp} \cos(\theta) - v_{xp} \sin(\theta)}{\varepsilon}$$

$$\dot{x}_P = v_{xp}$$
$$\dot{y}_P = v_{yp}$$



Some remarks:

- the change of coordinates has no singularities
- the change of coordinates is static
- the closed-loop system is described by two independent integrators
- an outer control loop should regulate the position of point P
- the change of coordinates is equivalent to the one introduced for the canonical model setting $y_r = 0$ and $x_r = \varepsilon$

Can we use this change of coordinates to linearize the bicycle kinematic model?

Exact linearization of a nonlinear system: the bicycle kinematic model

Consider a simplified version of the rear-wheel drive bicycle model

$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = \frac{v}{\ell}\tan(\phi)$$

We can apply a similar change of coordinates, considering that $\omega = \frac{v}{\ell}\tan\phi$.

Considering the previous linearizing law, we have

$$v = v_{xP}\cos(\theta) + v_{yP}\sin(\theta)$$
$$\omega\varepsilon = v_{yP}\cos(\theta) - v_{xP}\sin(\theta)$$

Let's concentrate on the second equation

$$\tan(\phi) = \frac{\ell}{v\varepsilon}v_{yP}\left(\cos(\theta) - v_{xP}\sin(\theta)\right) = \frac{\ell}{\varepsilon}\frac{v_{yP}\cos(\theta) - v_{xP}\sin(\theta)}{v_{xP}\cos(\theta) + v_{yP}\sin(\theta)}$$

Exact linearization of a nonlinear system: the bicycle kinematic model

Summarizing, the change of coordinates is

$$v = v_{x_P} \cos(\theta) + v_{y_P} \sin(\theta)$$

$$\phi = \arctan\left(\frac{\ell}{\varepsilon} \frac{v_{y_P} \cos(\theta) - v_{x_P} \sin(\theta)}{v_{x_P} \cos(\theta) + v_{y_P} \sin(\theta)}\right)$$

and the closed-loop system is again characterized by two independent integrators

$$\dot{x}_P = v_{x_P}$$

$$\dot{y}_P = v_{y_P}$$

But the change of coordinates is singular when $v = 0$.

Exact linearization of the unicycle and bicycle models: an interpretation

Before studying the linearization of a dynamic model, we would like to investigate more the exact linearization tool.

In all the examples we started from a 3D configuration space, representing robot pose, and we end up with a reduced configuration space, representing robot position.

From a physical point of view, the model in the new coordinates describes the motion of the robot as the motion of a particle.

This particle can move in the 2D space and it is not subjected to any constraint.

From a system theory point of view, the change of coordinates transforms a 3rd order system into a 2nd order system, it is thus a feedback that induces a loss of observability.

For a robot model the heading is no more observable from the output.

What are the main consequences?

- The linearizing feedback induces a hidden dynamics that can/cannot be asymptotically stable… if it is not asymptotically stable the change of coordinates cannot be applied
- The heading is no more observable, an outer controller cannot control the heading of the robot

Let’s consider an example…

Consider the unicycle model linearized with respect to a point P and the change of coordinates

$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = \omega$$

$$v = v_{xP}\cos(\theta) + v_{yP}\sin(\theta)$$
$$\omega = \frac{v_{yP}\cos(\theta) - v_{xP}\sin(\theta)}{\varepsilon}$$

The closed-loop system is described by the following dynamical system

$$\dot{x} = v_{xP}\cos^2(\theta) + v_{yP}\sin(\theta)\cos(\theta)$$
$$\dot{y} = v_{xP}\cos(\theta)\sin(\theta) + v_{yP}\sin^2(\theta)$$
$$\dot{\theta} = \frac{v_{yP}\cos(\theta) - v_{xP}\sin(\theta)}{\varepsilon}$$

We assume that P moves at constant velocity along a straight line

$$v_{x_P}(t) = \bar{v}_P \cos\left(\bar{\theta}_P\right)$$

$$v_{y_P}(t) = \bar{v}_P \sin\left(\bar{\theta}_P\right)$$

with $\bar{v}_P$, $\bar{\theta}_P$ constant and $\bar{v}_P > 0$.

The equations of the closed-loop system become

$$\dot{x} = \left[\bar{v}_P \cos\left(\bar{\theta}_P\right)\cos(\theta) + \bar{v}_P \sin\left(\bar{\theta}_P\right)\sin(\theta)\right]\cos(\theta) = \bar{v}_P \cos\left(\theta - \bar{\theta}_P\right)\cos(\theta)$$

$$\dot{y} = \left[\bar{v}_P \cos\left(\bar{\theta}_P\right)\cos(\theta) + \bar{v}_P \sin\left(\bar{\theta}_P\right)\sin(\theta)\right]\sin(\theta) = \bar{v}_P \cos\left(\theta - \bar{\theta}_P\right)\sin(\theta)$$

$$\dot{\theta} = \frac{\bar{v}_P \sin\left(\bar{\theta}_P\right)\cos(\theta) - \bar{v}_P \cos\left(\bar{\theta}_P\right)\sin(\theta)}{\varepsilon} = -\frac{\bar{v}_P}{\varepsilon}\sin\left(\theta - \bar{\theta}_P\right)$$

Focusing on the unobservable state, we define $\Delta\theta = \theta - \bar{\theta}_P$.

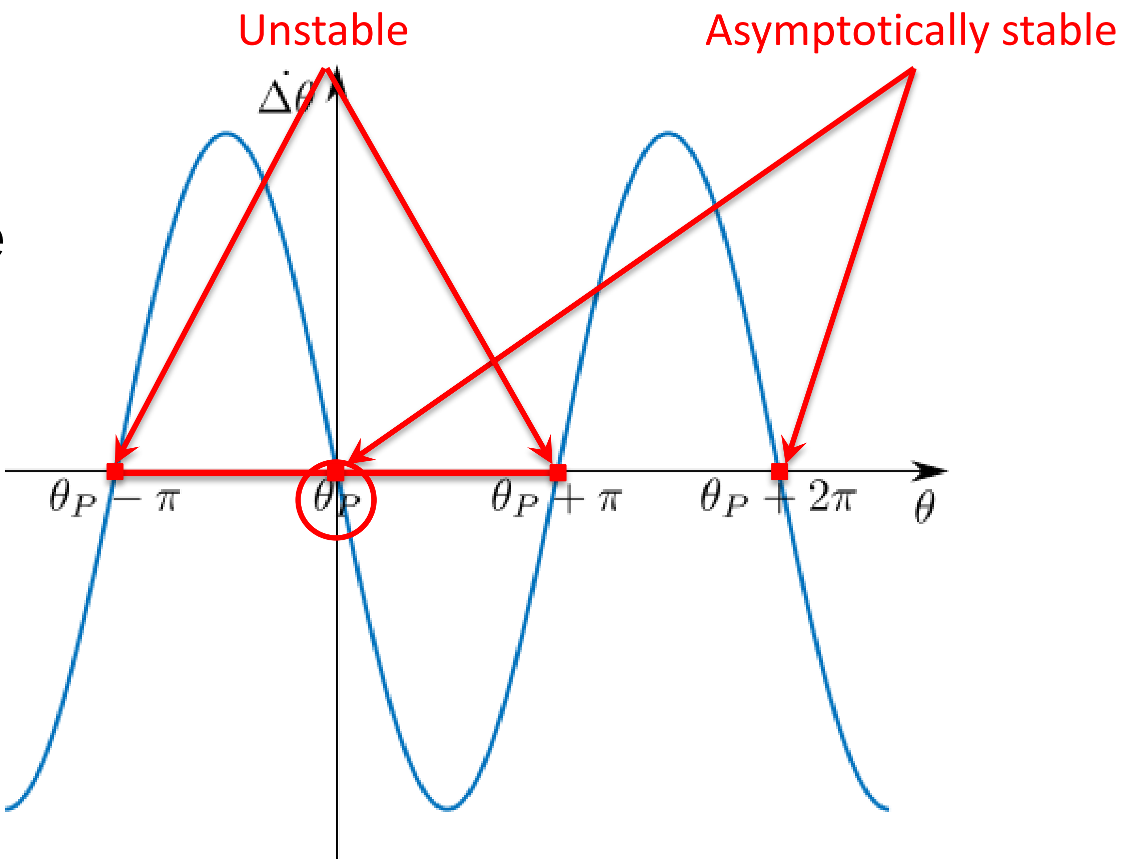
Exact linearization of the unicycle and bicycle models: an interpretation

The heading dynamics can be written as

$$\Delta\dot{\theta} = -\frac{\bar{v}_P}{\varepsilon}\sin(\Delta\theta)$$

and the equilibria of this nonlinear system are

- $\theta = \bar{\theta}_P + 2k\pi$, $k \in \mathbb{Z}$, asymptotically stable with a basin of attraction $(\bar{\theta}_P + 2(k -$



Exact linearization of a nonlinear system: the bicycle dynamic model

What about linearizing a dynamic model?

We consider the example of the bicycle model

$$\ddot{\psi} = \frac{bC_r\alpha_r - aC_f\alpha_f}{I_z}$$

$$\dot{\beta} = -\frac{C_f\alpha_f + C_r\alpha_r}{MV} - \dot{\psi}$$

$$\dot{x} = V\cos(\psi + \beta)$$

$$\dot{y} = V\sin(\psi + \beta)$$

$$\alpha_f = \beta + \frac{a\dot{\psi}}{v} - \delta$$

$$\alpha_r = \beta - \frac{b\dot{\psi}}{v}$$

As for the unicycle kinematic model we can linearize with respect to a point P that does not belong to the vehicle.

Exact linearization of a nonlinear system: the bicycle dynamic model

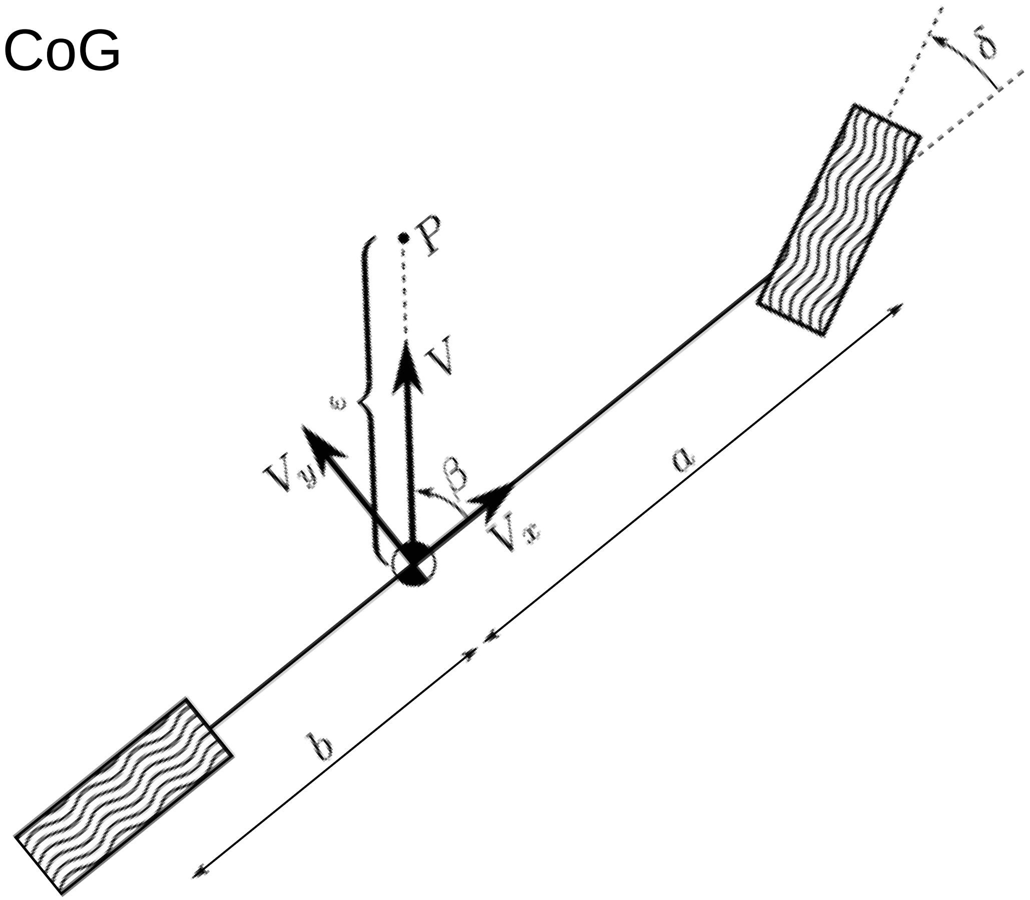
Consider a point P at a distance ε from the vehicle CoG along the velocity vector

$$x_P = x + \varepsilon \cos(\beta + \psi)$$

$$y_P = y + \varepsilon \sin(\beta + \psi)$$

and differentiating with respect to time

$$\begin{aligned} \dot{x}_P &= \dot{x} - \varepsilon \left(\dot{\beta} + \dot{\psi}\right) \sin(\beta + \psi) \\ &= V \cos(\beta + \psi) - \varepsilon \left(\dot{\beta} + \dot{\psi}\right) \sin(\beta + \psi) \\ \dot{y}_P &= \dot{y} + \varepsilon \left(\dot{\beta} + \dot{\psi}\right) \cos(\beta + \psi) \\ &= V \sin(\beta + \psi) + \varepsilon \left(\dot{\beta} + \dot{\psi}\right) \cos(\beta + \psi) \end{aligned}$$



Exact linearization of a nonlinear system: the bicycle dynamic model

Introducing the sideslip dynamics

$$\omega = \dot{\beta} + \dot{\psi} = -\frac{C_f \alpha_f + C_r \alpha_r}{MV} = -\frac{C_f + C_r}{MV}\beta + \frac{bC_r - aC_f}{MV}\frac{\dot{\psi}}{V} + \frac{C_f}{MV}\delta$$

in the previous equations

$$\dot{x}_P = V\cos(\beta+\psi) - \varepsilon\left(\dot{\beta}+\dot{\psi}\right)\sin(\beta+\psi) = V\cos(\beta+\psi) - \varepsilon\omega\sin(\beta+\psi) = V_{P_x}$$

$$\dot{y}_P = V\sin(\beta+\psi) + \varepsilon\left(\dot{\beta}+\dot{\psi}\right)\cos(\beta+\psi) = V\sin(\beta+\psi) + \varepsilon\omega\cos(\beta+\psi) = V_{P_y}$$

we obtain the change of coordinates

$$\begin{bmatrix} V_{P_x} \\ V_{P_y} \end{bmatrix} = \begin{bmatrix} \cos(\beta+\psi) & -\varepsilon\sin(\beta+\psi) \\ \sin(\beta+\psi) & \varepsilon\cos(\beta+\psi) \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\beta+\psi) & \sin(\beta+\psi) \\ -\frac{\sin(\beta+\psi)}{\varepsilon} & \frac{\cos(\beta+\psi)}{\varepsilon} \end{bmatrix} \begin{bmatrix} V_{P_x} \\ V_{P_y} \end{bmatrix}$$

Exact linearization of a nonlinear system: the bicycle dynamic model

Finally, from the definition of ω we obtain

$$\delta = \frac{MV}{C_f}\omega + \frac{C_f + C_r}{C_f}\beta - \frac{bC_r - aC_f}{C_f}\frac{\dot{\psi}}{V}$$

Summarizing, if we apply the change of coordinates

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\beta+\psi) & \sin(\beta+\psi) \\ -\frac{\sin(\beta+\psi)}{\varepsilon} & \frac{\cos(\beta+\psi)}{\varepsilon} \end{bmatrix} \begin{bmatrix} V_{P_x} \\ V_{P_y} \end{bmatrix}$$

$$\delta = \frac{MV}{C_f}\omega + \frac{C_f + C_r}{C_f}\beta - \frac{bC_r - aC_f}{C_f}\frac{\dot{\psi}}{V}$$

the bicycle dynamic model is transformed into two independent integrators

$$\dot{x}_P = V_{xP}$$

$$\dot{y}_P = V_{yP}$$

Some remarks:

- as for the kinematic model, the linearized system represent the motion of a particle
- sideslip and yaw dynamics are hidden by the change of coordinates
- the change of coordinates is static but it is singular when $V = 0$
- the change of coordinates is now a function of the dynamic model parameters (M, C_f, C_r, a, b), there can be robustness issues

Exact linearization of a nonlinear system: a general approach

We can now generalize the approach we have adopted for the unicycle and bicycle kinematic model, and for the bicycle dynamic model:

1. Select a point P, not belonging to the null velocity lines, whose position represents the controlled variable
2. Express the position of point P as a function of the configuration variables
3. Differentiate the previous expression with respect to time until the inputs of the original model appears
4. Derive a relation between the new inputs (velocity, acceleration, jerk, snap,… of point P) and the inputs of the original model

The generic expression of the dynamic model of a mobile robot we have introduced

$$\dot{\mathbf{q}} = G(\mathbf{q})\,\mathbf{v}$$
$$\dot{\mathbf{v}} = -M^{-1}(\mathbf{q})\,\mathbf{m}(\mathbf{q},\mathbf{v}) + M^{-1}(\mathbf{q})\,G^T(\mathbf{q})\,S(\mathbf{q})\,\tau$$

allows to devise a general but partial feedback linearization.

Assuming

$$det\left(G^T(\mathbf{q})\,S(\mathbf{q})\right) \neq 0$$

a "control availability" assumption that is often satisfied, we can select

$$\tau = \left(G^T(\mathbf{q})\,S(\mathbf{q})\right)^{-1}\left(M(\mathbf{q})\,\mathbf{a} + \mathbf{m}(\mathbf{q},\mathbf{v})\right)$$

where $\boldsymbol{a}$ is the new control variable. The closed-loop system is reduced to

$$\dot{\mathbf{q}} = G(\mathbf{q})\,\mathbf{v}$$
$$\dot{\mathbf{v}} = \mathbf{a}$$

m independent integrators

Kinematic model

Some remarks:

- it is a general approach similar to the one used to linearize the model of a manipulator
- as for the previous linearization of dynamic models, the linearizing law entails model parameters, a robustness issue can arise
- if the system is unconstrained and fully actuated

$$G(\mathbf{q}) = S(\mathbf{q}) = I_n$$

the linearizing law reduces to

$$\tau = M(\mathbf{q})\,\mathbf{a} + \mathbf{m}(\mathbf{q}, \mathbf{v})$$

and it is thus equivalent to an *inverse dynamics control*. The linearized system is equivalent to n decoupled double integrators

$$\dot{\mathbf{q}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = \mathbf{a}$$

The motion control problem for a mobile robot can be formulated with respect to

- kinematic model
- dynamic model

Dynamic effects are handled by low level control systems or can be neglected (low velocities/accelerations)

Dynamic effects are more important in autonomous vehicles than in mobile robotics

At least two reasons allow to go for the first option:

- dynamics can be cancelled out with a dynamic state feedback
- in the majority of the robots wheel torques cannot be accessed, as there are low level control loops integrated in the hardware architecture, and generalized velocities are usually the only accessible commands

Remember that, making reference to the motion of point P

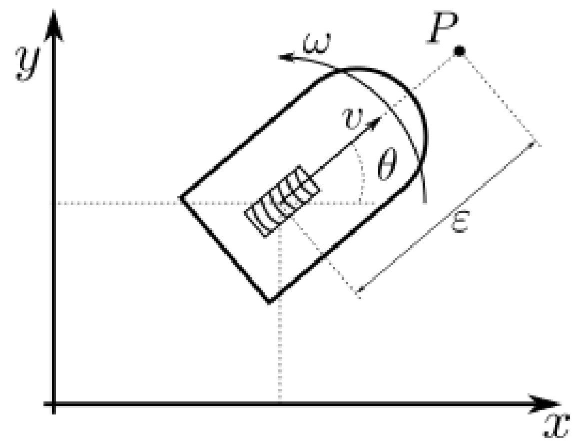
$$x_P = x + \varepsilon \cos(\theta)$$
$$y_P = y + \varepsilon \sin(\theta)$$

and introducing the feedback linearizing law

$$v = v_{xP} \cos(\theta) + v_{yP} \sin(\theta)$$
$$\omega = \frac{v_{yP} \cos(\theta) - v_{xP} \sin(\theta)}{\varepsilon}$$

we obtain

$$\dot{x}_P = v_{xP}$$
$$\dot{y}_P = v_{yP}$$
$$\dot{\theta} = \frac{v_{yP} \cos(\theta) - v_{xP} \sin(\theta)}{\varepsilon}$$



The dynamics of point P is now linear and can be controlled with a linear regulator

A feedback linearization based trajectory tracking controller for the canonical model

A simple linear controller

$$v_{xP} = \dot{x}_{P_d} + k_1 \left(x_{P_d} - x_P \right)$$
$$v_{yP} = \dot{y}_{P_d} + k_2 \left(y_{P_d} - y_P \right)$$

with $k_1, k_2 > 0$, guarantees exponential convergence to zero of the Cartesian tracking error with decoupled dynamics

$$\left(\dot{x}_{P_d} - \dot{x}_P \right) + k_1 \left(x_{P_d} - x_P \right) = \dot{e}_{P_x} + k_1 e_{P_x} = 0$$
$$\left(\dot{y}_{P_d} - \dot{y}_P \right) + k_2 \left(y_{P_d} - y_P \right) = \dot{e}_{P_y} + k_2 e_{P_y} = 0$$

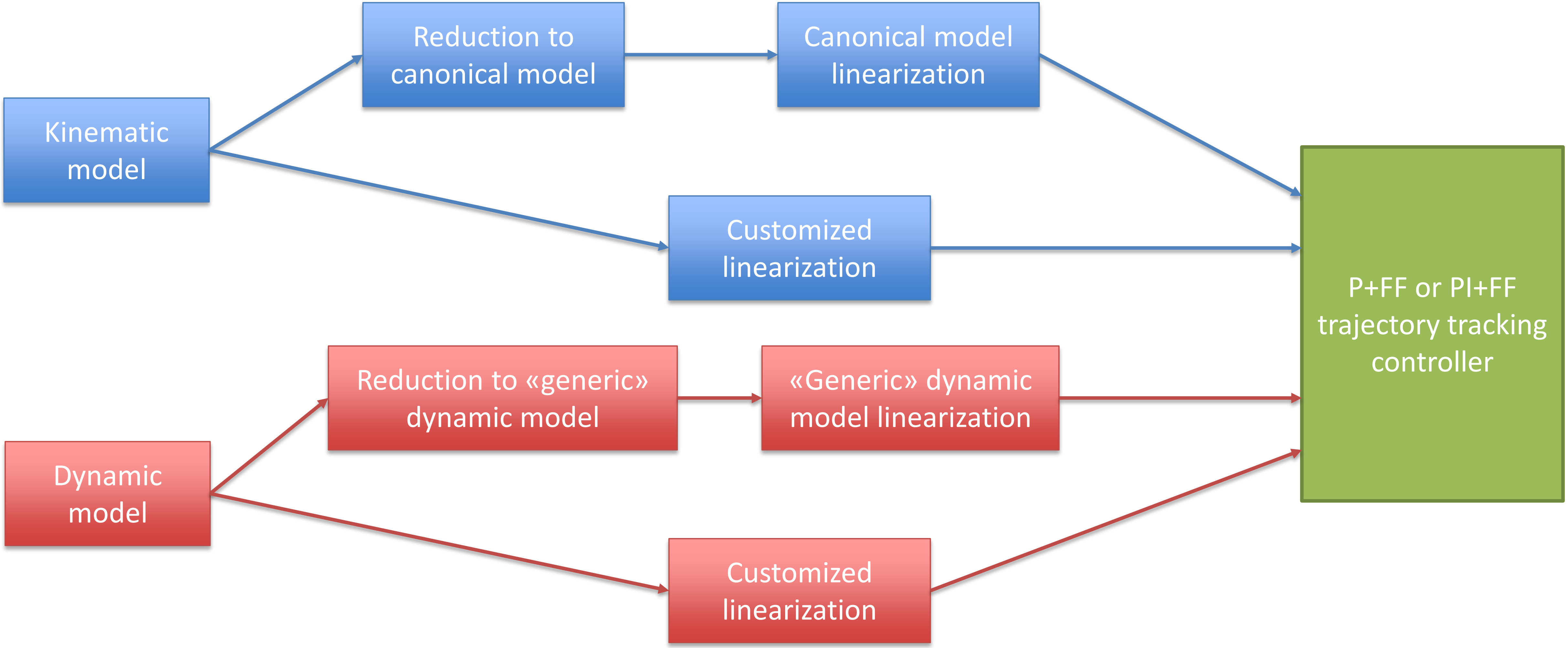
The orientation of the robot is not controlled and evolves according to

$$\dot{\theta} = \frac{\left[\dot{y}_{P_d} + k_2 \left(y_{P_d} - y_P \right) \right] \cos(\theta) - \left[\dot{x}_{P_d} + k_1 \left(x_{P_d} - x_P \right) \right] \sin(\theta)}{\varepsilon}$$

but at least at steady state the equilibrium of this system is

$$\dot{y}_{P_d} \cos(\theta) = \dot{x}_{P_d} \sin(\theta) \quad \Rightarrow \quad \tan(\theta) = \frac{\dot{y}_{P_d}}{\dot{x}_{P_d}} = \tan(\theta_d)$$

Trajectory tracking control of a mobile robot



As we already know, to control a mobile robot we need to estimate in real-time the robot configuration at each time instant. This is called *localization problem*.

Consider a unicycle robot moving under the action of velocity commands v and ω, constant within each sampling interval.

In the sampling interval the robot moves along:

- an arc of circle of radius $R = v_k/\omega_k$, if $\omega_k \neq 0$
- a line segment, if $\omega_k = 0$

Assume the robot configuration at time t_k is known, $\boldsymbol{q}(t_k) = \boldsymbol{q}_k$, together with the inputs v_k and ω_k in the interval $[t_k, t_{k+1})$.

Using forward integration of the kinematic model with the Euler method, we can derive the configuration $\boldsymbol{q}_{k+1}$ at time t_{k+1}

$$x_{k+1} = x_k + v_k T_s \cos(\theta_k)$$
$$y_{k+1} = y_k + v_k T_s \sin(\theta_k)$$
$$\theta_{k+1} = \theta_k + \omega_k T_s$$

These relations are approximated as they assume θ_k constant in the integration period

This relation is exact

where $T_s = t_{k+1} - t_k$.

A more accurate estimate can be achieved adopting the second-order Runge-Kutta integration method

$$x_{k+1} = x_k + v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right)$$

$$y_{k+1} = y_k + v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right)$$

$$\theta_{k+1} = \theta_k + \omega_k T_s$$

The average value of the orientation in the integration period

More complex and accurate methods can be devised.

We have now to relate our estimators to a set of available measurements, wheel encoder measurements.

Consider, for example, the case of a differential drive robot.

If $\Delta\phi_{R_k}$ and $\Delta\phi_{L_k}$ are the rotations of the wheels measured by the incremental encoders during the *k*-th sampling interval

$$\Delta s_k = \frac{r}{2}\left(\Delta\phi_{R_k} + \Delta\phi_{L_k}\right) \qquad \Delta\theta_k = \frac{r}{d}\left(\Delta\phi_{R_k} - \Delta\phi_{L_k}\right)$$

The linear and angular velocities can be estimated as

$$v_k = \frac{\Delta s_k}{T_s} \qquad \omega_k = \frac{\Delta\theta_k}{T_s}$$

The method we have introduced is called *odometric localization* or *passive localization* or *dead reckoning*.

This method is subject to an error that grows over time (*drift*) that quickly becomes significant over sufficiently long paths. This is due to several causes, including wheel slippage, inaccuracy in kinematic parameters, numerical errors in the integration,…