Wheeled robots

FIELD AND SERVICE ROBOTICS

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

www.prisma.unina.it

Differential geometry

- Vectors and co-vectors
  - A vector is the element of a vector space $v \in V \quad v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$
  - A co-vector is the element of a co-vector space $w^* \in V^* \quad w^* = [w_1 \quad \cdots \quad w_n]$
- Vector field

$$f(x): x \in U \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$$

  - Smooth vector field if the mapping is smooth
  - It is a vector of (smooth) real-valued functions
- Co-vector field
  - It is a co-vector of (smooth) real-valued functions
- Inner product

$$w^* v = < w^*, v > = \sum_{i=1}^{n} w_i v_i$$

Operators

- Differential (gradient)
  - $\lambda(x): x \in U \subseteq \mathbb{R}^n \to \mathbb{R}$ real-valued funciton

$$d\lambda(x) = \frac{\partial \lambda}{\partial x} = \begin{bmatrix} \frac{\partial \lambda}{\partial x_1} & \cdots & \frac{\partial \lambda}{\partial x_n} \end{bmatrix} \text{ co-vector field}$$

- Lie derivative
  - $f(x): x \in U \subseteq \mathbb{R}^n \to \mathbb{R}^n$ vector field
  - It is the inner product between a gradient and a vector field

$$L_f \lambda = < d\lambda(x), f(x) > = \frac{\partial \lambda}{\partial x} f(x) = \sum_{i=1}^{n} \frac{\partial \lambda_i}{\partial x_i} f_i(x) \text{ real-valued function}$$

  - The opeation can be iterated
- Lie bracket
  - It is defined starting from two vector fields $f(x)$ and $g(x)$

$$[f(x), g(x)] = \frac{\partial g}{\partial x} f(x) - \frac{\partial f}{\partial x} g(x) \text{ vector field}$$

  - The gradient of a vector field is called Jacobian
  - The operation can be iterated

- Distributions
  - Suppose we have d vector fields defined on the same open set $U \subseteq \mathbb{R}^n$
  - Suppose that at any $x \in U \subseteq \mathbb{R}^n$, the vector fields $f_1(x), f_2(x), \dots, f_d(x)$ span the following vector space
$$\Delta(x) = span\{f_1(x), f_2(x), \dots, f_d(x)\}$$
    - If the vector fields are smooth, this is a smooth distribution
  - The dimension of a distribution at a point $x \in U \subseteq \mathbb{R}^n$ is the dimension of the subspace $\Delta(x)$
    - Let consider a matrix having n rows and whose entries are smooth functions of x
    - The columns of such a matrix can be considered as smooth vector fields
    - The dimension of a distribution spanned by the columns of a matrix is the rank of the matrix at the evaluated point
  - A distribution is nonsingular if $\dim(\Delta(x)) = d, \forall x \in U$
  - A singular distribution is a distribution with variable dimension
  - A point $x^o \in U$ is said to be a regular point of the distribution if there exists a neighbourhood U^o of x^o with the property that the distribution is non singular on U^o
    - Each point that is not regular is said to be a point of singularity

- Co-distributions
  - As for distributions but with co-vector fields
  - Any matrix identifies a co-distribution, the one spanned by its rows

- LEMMA
  - Let Δ be a smooth distribution and $x^o \in U \subseteq \mathbb{R}^n$ a regular point of the distribution.
    Suppose $\dim\left(\Delta(x^o)\right) = d$.
    There exists an open neighbourhood U^o of x^o and a set of smooth vector fields, $f_1(x), f_2(x), \dots, f_d(x)$, defined on U^o with the following properties.
    - The vectors $f_1(x), f_2(x), \dots, f_d(x)$ are linearly independent at each $x \in U^o \subseteq \mathbb{R}^n$
    - $\Delta(x) = span\{f_1(x), f_2(x), \dots, f_d(x)\}$ at each x $\in U^o \subseteq \mathbb{R}^n$
    - Every smooth vector field $\tau \in \Delta$ can be expressed on U^o as $\tau = \sum_{i=1}^{d} c_i(x) f_i(x)$ where $c_1(x), \cdots, c_d(x)$ are smooth real-valued functions of $x \in U^o \subseteq \mathbb{R}^n$

Involutive distrubtions

- A distribution is involutive if the Lie bracket $[\tau_1, \tau_2]$ of any pair of vector fields $\tau_1, \tau_2 \in \Delta$ is a vector field belonging to Δ

$$[\tau_1, \tau_2] \in \Delta, \forall \tau_1, \tau_2 \in \Delta$$

- Because of the previous Lemma $\tau_1 = \sum_{i=1}^{d} c_i(x) f_i(x)$ and $\tau_2 = \sum_{i=1}^{d} d_i(x) f_i(x)$
  - Then, Δ is involutive if and only if $[f_i, f_j] \in \Delta, \forall f_i, f_j \in \Delta, 1 \leq i, j \leq d$
- Notice that, by definition, any 1-dimensional distribution is involutivive since $[f, f] = 0$, and the zero vector belongs to any distribution by default

Annihilator

- Sometimes, it is possible to construct co-distributions from given distributions and vice-versa
  - Given a distribution Δ, for each $x \in U \subseteq \mathbb{R}^n$ consider the annihilator of the distribution, that is

$$\Delta^{\perp}(x) = \{w^* \in \mathbb{R}^{n^*} : < w^*, v > = 0, \forall v \in \Delta(x)\}$$ → co-distribution

  - Notice that $\dim(\Delta) + \dim(\Delta^{\perp}) = n$
  - If we see the distribution as spanned by the columns of a matrix $F(x)$, then its annihilator is identified by the set of co-vectors such that $w^* F(x) = 0$
  - Conversely, if a co-distribution is spanned by the rows of a matrix $W(x)$, then its annihilator (that is a distribution) is identified by the vectors such that $W(x)v = 0$
    - The distributions corresponds to the kernel of $W(x)$

- Consider the following control-affine system

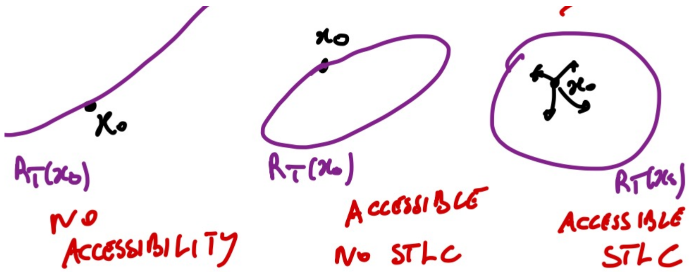
$$\Sigma:\ \dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i$$

  - $f, g_i\ (i = 1, \dots, m)$ smooth vector fields of n components
  - $u \in \mathbb{R}^m$ control input vector
  - $f(x)$ drift vector
    - A system in which $f(x) = 0$ is called driftless
- Definition of controllability *[2]*
  - *The system Σ is said to be controllable if for any two points x_0 and x_f, there exists an admissible control $u(t)$, defined on some time interval $[0, T]$, such that the system Σ with initial conditions x_0 reaches x_f in the finite time T*

[2] A. Isidori, "Nonlinear control systems," Springer-Verlag, 3rd edition, 1995

Reachable set

- Controllability is not easy to prove, in general
- A reachable set is the set of points that may be reached by the system by traveling on trajectories from the initial point in a finite time
- Definitions *[2]*
  - *Given an initial point $x_0 \in U \subseteq \mathbb{R}^n$, we define $R(x_0, t)$ the set of all the points $x \in U$ for which there exists an admissible control input $u(t)$ such that there is a trajectory of Σ with $x(0) = x_0$ and $x(t) = x$*
  - *The reachable set from x_0 at the time T is defined to be $R_T(x_0) = \bigcup_{0 \le t \le T} R(x_0, t)$*
  - *Σ is accessible from $x_0 \in U \subseteq \mathbb{R}^n$ if there exists $T > 0$ such that $R_T(x_0)$ contains a nonempty set*
  - *Σ is small-time locally controllable (STLC) from $x_0 \in U \subseteq \mathbb{R}^n$ if x_0 is an interior point of $R_T(x_0)$ for any $T > 0$*



Accessibility distribution and theorems

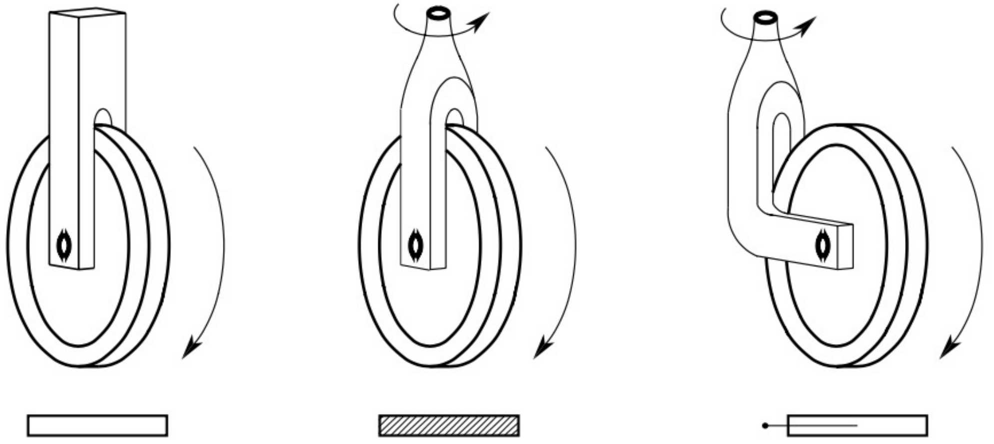
- *Let the accessibility distribution, Δ_A, of Σ be the distribution generated by the vector fields $f, g_1, g_2, \dots, g_m$ and all the Lie brackets that can be generated by these vector fields*

- Theorem *[2]*
  - *Consider the affine control system Σ and assume that the vector fields are C^∞. If $dim\big(\Delta_A(x_0)\big) = n$, then $\forall T > 0$ the set $R_T(x_0)$ has a nonempty interior, that is, the system Σ is accessible from x_0*
    - $dim\big(\Delta_A(x_0)\big) = n$ is the so-called accessibility rank condition

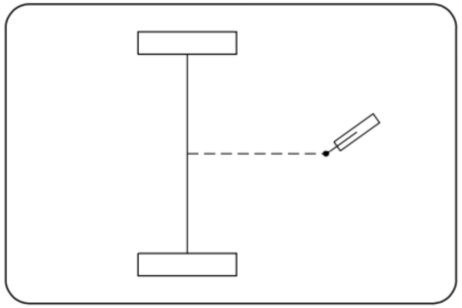
- Theorem *[2]*
  - *Suppose that Σ is a smooth affine control system and driftless. If the accessibility rank condition is satisfied $\forall x \in U$, then the system Σ is controllable*

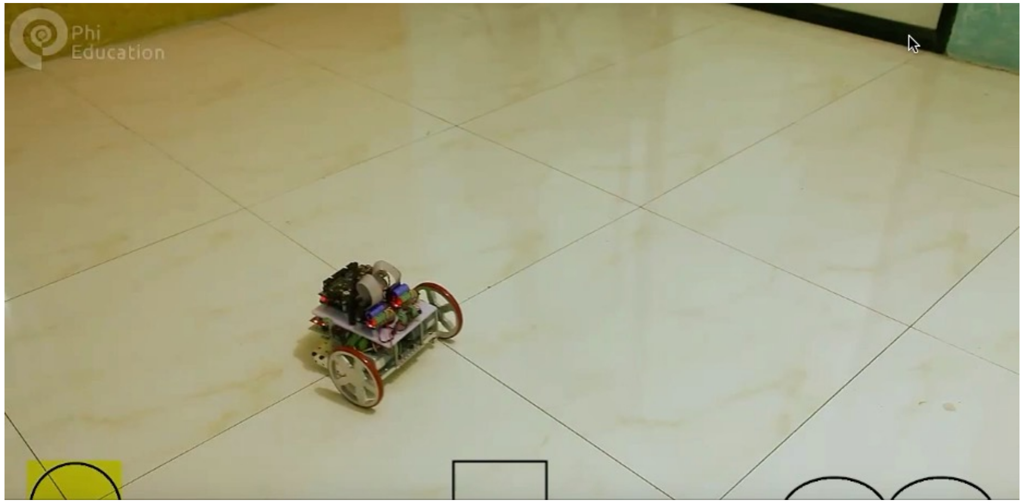
Wheeled robots

- Different type of wheels



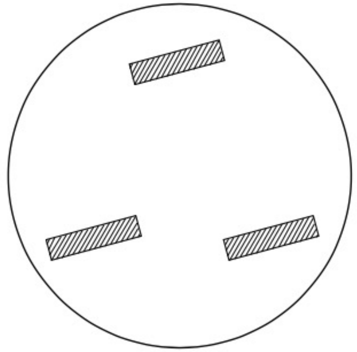
- Differential-drive robot
  - Two fixed wheels + 1 caster wheel (passive)
  - Rotation axes in common (fixed wheels)
  - Two independent motors (fixed wheels)



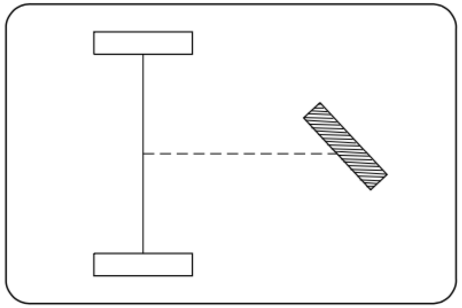


Synchro-drive robot

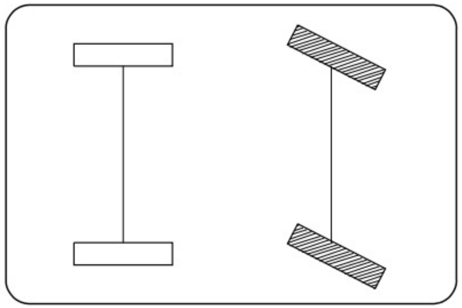
- Synchro-drive robot
  - Three aligned steerable wheels
  - Two motors in total to control orientation and traction of the wheels



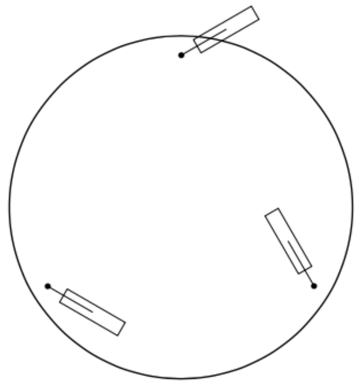
- Tricycle robot
  - Two fixed wheels as the differential drive robot + one steerable wheel
  - Two motors
    - One for the fixed wheels (traction) and one for the steerable wheel (rotation)
    - Two for the steerable wheel
  - Differential is needed



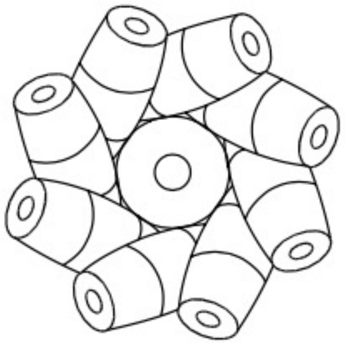
- Car-like robot
  - Two fixed wheels + two steerable wheels
  - Two motors
    - One for the fixed wheels (traction) and one for the steerable wheel (rotation)
    - Two for the steerable wheel

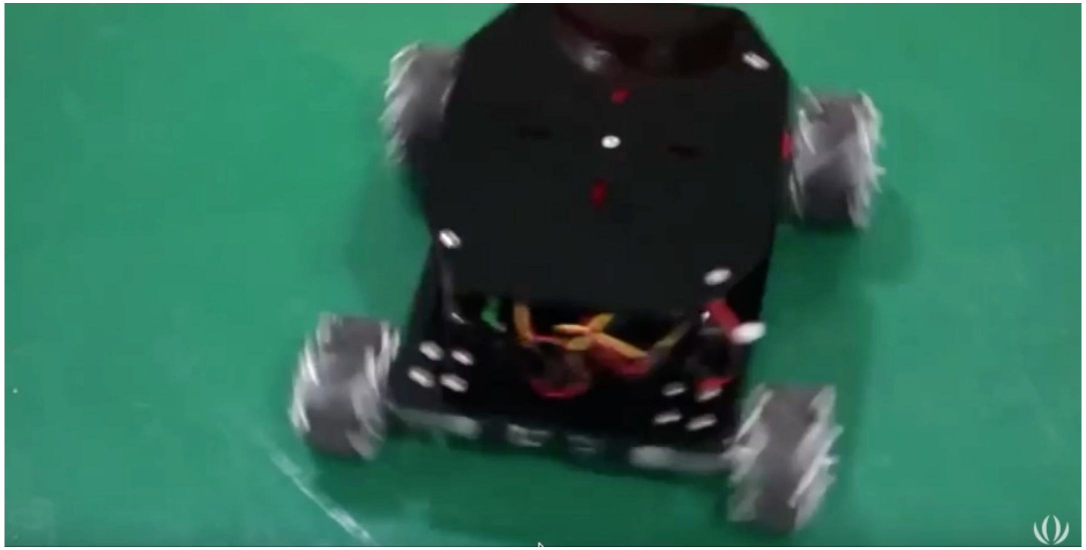


- Omnidirectional robot
  - Three caster wheels
  - One motor for each wheel



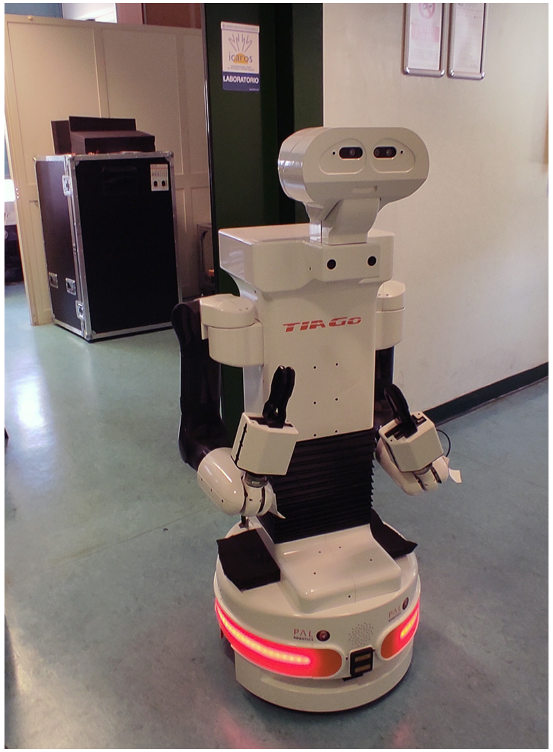
- Swedish wheel (Mecanum)
  - Fixed wheel
  - Car-like robot with four Mecanum wheels and four motors (one for each wheel) is omnidirectional





Mobile manipulation

- Examples



Motion constraints

- Consider a mechanical system whose model-configuration is denoted by $q \in \mathbb{R}^n$
  - q is the vector of generalised coordinates
  - $q(t)$ is the robot motion
- Motion constraints
  - Unilateral constraints
    - Expressed by inequalities
  - Bilateral constraints
    - Expressed by equalities
  - Rheonomic constraints
    - Time dependent
  - Scleronomic constraints
    - Time independent

Holonomic constraints

- Constraints in the form

$$h_i(q) = 0, \qquad i = 1, \dots, k < n$$

are said to be holonomic (integrable) constraints

  - $h_i: \mathbb{R}^n \rightarrow \mathbb{R}$ and $h_i \in C^\infty$ and it is smooth

- The effect of holonomic constraints is to reduce the space of admissible model-configurations to a subset of $\mathbb{R}^n$of dimension $n - k$

- Constraints in the form

$$a_i(q,\dot{q}) = 0, \qquad i = 1,\dots,k < n$$

are said to be kinematic constraints

  - They constraint the instantaneous admissible motion, reducing the set of generalised velocities that can be attained at each model-configuration

- Kinematic constraints are generally expressed in the following so-called Pfaffian form

$$a_i^T(q)\dot{q} = 0, \qquad i = 1,\dots,k < n$$

  - $a_i^T : \mathbb{R}^n \to \mathbb{R}^{n^*}$ are linearly independent smooth co-vector fields
  - In a compact form

$$A^T(q)\dot{q} = 0$$

$$A^T(q) = \begin{bmatrix} a_1^T(q) \\ \vdots \\ a_k^T(q) \end{bmatrix} \text{Pfaffian matrix}$$

- Every holonomic constraint can be seen as a kinematic constraint in the Pfaffian form

$$h_i(q) = 0, \qquad i = 1, \dots, k$$

$$\frac{d}{dt} h_i(q) = \underbrace{\frac{\partial h_i(q)}{\partial q}}_{a_i^T(q)} \dot{q} = 0$$

  - The converse is not true in general!

Nonholonomic constraints

- Definition
  - Given a set of k kinematic constraints in the Pfaffian form $A^T(q)\dot{q} = 0$, if they do not come from holonomic constraints, that means if they are not integrable, then they are called nonholonomic constraints
- A mechanical system with at least one nonholonomic constraint is a nonholonomic system
- Question
  - How can we understand if $A^T(q)\dot{q} = 0$ is integrable or not?
  - To answer this question, we must pass through the controllability analysis of the kinematic model associated to the nonholonomic constraints

Kinematic model associated to nonholonomic constraints

- Suppose to have a mechanical system subject to kinematic constraints in a Pfaffian form

$$A^T(q)\dot{q} = 0$$

  - $A^T \in \mathbb{R}^{k \times n}$
  - $q, \dot{q} \in \mathbb{R}^n$
  - These constraints prevent the mechanical system to move along certain directions
  - We want to understand in which of the $m = n - k$ admissible directions our system can be moved
- Let us construct a distribution from the Pfaffian matrix
  - The distribution is spanned by the vectors $g_j(q), \quad j = 1, \dots, m$ such that

$$A^T(q)g_j(q) = 0$$

$$\Delta = span\{g_j(q) \in \mathbb{R}^n : A^T(q)g_j(q) = 0, j = 1, \dots m\}$$

  - A generic vector belonging to the distribution can be expressed as linear combination of the vector fields spanning the distribution

$$\dot{q} = \sum_{j=1}^{m} g_j(q)u_j = G(q)u \ \in \Delta$$

Kinematic model associated to nonholonomic constraints

- The expression

$$\dot{q} = G(q)u$$

is the kinematic model of a mechanical system subject to $A^T(q)\dot{q} = 0$ and expresses all the admissible trajectories

  - The kinematic model built in this way is not unique
    - The control input vector u may have different interpretations: we seek kinematic models giving a physical meaning to u
  - Notice that the kinematic model is a driftless affine control system

Kinematic model associated to nonholonomic constraints

- The holonomy or nonholonomy of $A^T(q)\dot{q} = 0$ can be discriminated through the associated kinematic model $\dot{q} = G(q)u$
  - $\dot{q} = G(q)u$ is controllable
    - There exists a choice of $u(t)$ bringing the system from a given model-configuration to another
    - The system is subject to nonholonomic constraints only
    - $n = \nu$, with $\nu = \dim(\Delta_A(q))$ and $\Delta = span\{g_1(q), \dots, g_m(q)\}$
    - $A^T(q)\dot{q} = 0$ is not integrable, and it is a set of nonholonomic constraints in the Pfaffian form
  - $\dot{q} = G(q)u$ is not controllable
    - Some model-configurations are prevented
    - The system has some holonomic constraints
    - If $m < \nu < n$, with $\nu = \dim(\Delta_A(q))$ and $\Delta = span\{g_1(q), \dots, g_m(q)\}$, $A^T(q)\dot{q} = 0$ have only partially integrable constraints
    - The system is still nonholonomic
  - $\dot{q} = G(q)u$ is not controllable
    - Some model-configurations are prevented
    - The system has holonomic constraints only
    - If $m = \nu$, with $\nu = \dim(\Delta_A(q))$ and $\Delta = span\{g_1(q), \dots, g_m(q)\}$, $A^T(q)\dot{q} = 0$ have only integrable constraints
    - The system is completely holonomic

$$q = [x \quad y \quad \theta]^T$$

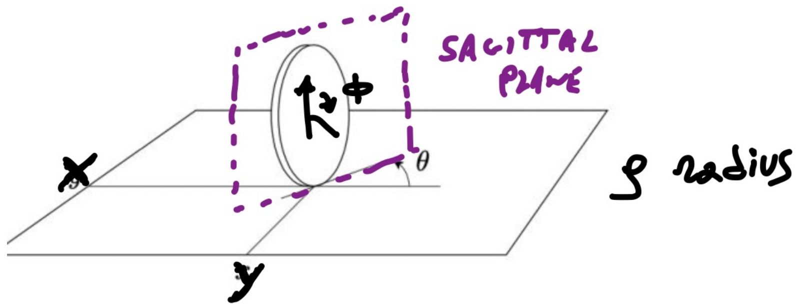
$$\begin{cases} \dot{x} = \rho\dot{\phi}\cos\theta & \textit{multiply both sides by} \sin\theta \\ \dot{y} = \rho\dot{\phi}\sin\theta & \textit{multiply both sides by} \cos\theta \end{cases}$$

$$\begin{cases} \dot{x}\sin\theta = \rho\dot{\phi}\cos\theta\sin\theta \\ \dot{y}\cos\theta = \rho\dot{\phi}\cos\theta\sin\theta \end{cases}$$

$$\Downarrow$$

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

Pure rolling constraint



the velocity perpendicular to this plane is 0

$$\underbrace{[\sin\theta \quad -\cos\theta \quad 0]}_{\boldsymbol{a}_i^T(\boldsymbol{q})} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}}_{\dot{\boldsymbol{q}}} = 0$$

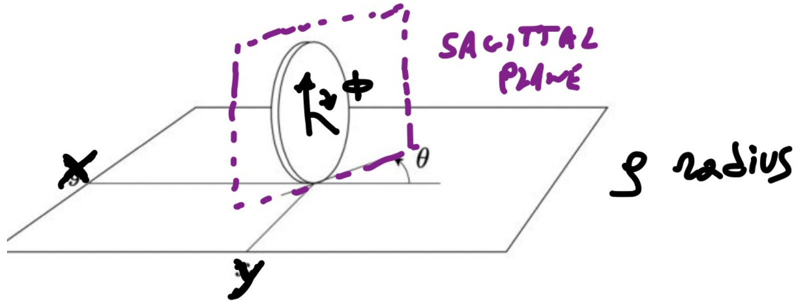
- In the absence of slipping, the velocity of the contact point has zero component in the direction orthogonal to the sagittal plane

Disk rolling on a plane

$$k = 1 \qquad m = n - k = 2$$
$$G(q) = [g_1(q) \quad g_2(q)] \quad u = [u_1 \quad u_2]^T$$
$$\dot{q} = G(q)u = g_1(q)u_1 + g_2(q)u_2$$

- Find the vector fields of the kinematic model

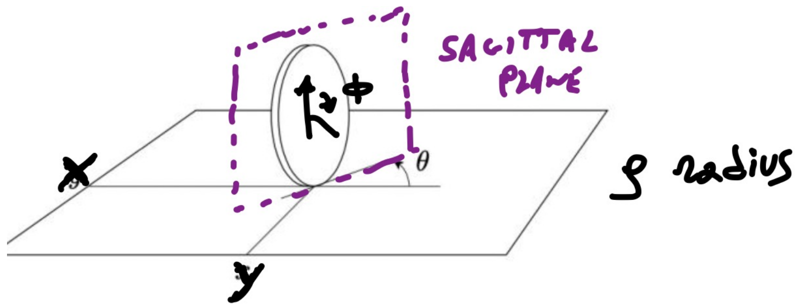
$$[\sin\theta \quad -\cos\theta \quad 0]\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0$$
$$\Downarrow$$
$$v_1 \sin\theta = v_2 \cos\theta$$
$$g_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad g_2(q) = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}$$



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

v heading velocity input
ω angular velocity input

Disk rolling on a plane



- Understand whether the constraint is holonomic or not

$$\Delta_A(q) = span\{g_1, g_2(q), [g_1, g_2(q)]\}$$

$$[g_1, g_2(q)](q) = \begin{bmatrix} 0 & 0 & -\sin\theta \\ 0 & 0 & \cos\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \times \\ \times \\ \times \end{bmatrix} =$$

$$\begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \\ 1 & 0 & 0 \end{bmatrix} \quad \det(F) = 1$$

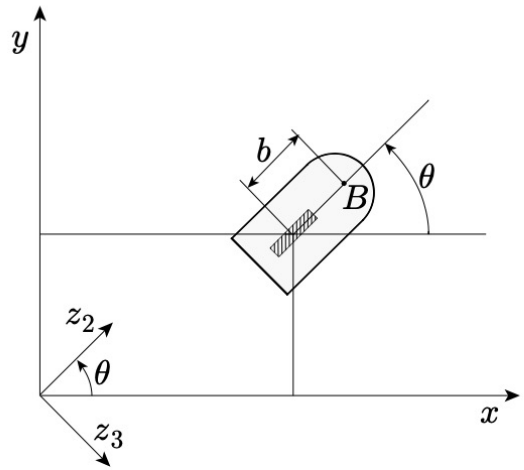
$$rank(F) = 3 \ \forall q \in \mathbb{R}^3$$

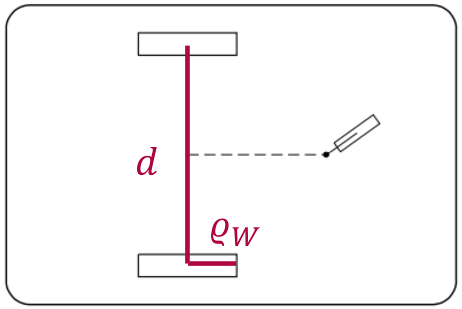
- The system is then controllable. Therefore, the system is completely nonholonomic. The pure rolling constraint is not integrable

- Very similar to the disk rolling on a plane
- Serious problem of balance
  - However, many vehicles are kinematically equivalent to a unicycle
    - An example is the differential-drive robot
    - (x, y) of the unicycle is the midpoint of the segment joining the two differential drive's fixed wheels
    - θ is the common orientation of the fixed wheels
- The unicycle's inputs are
  - v the heading velocity
  - ω the steering velocity
- The differential-drive robot's inputs are
  - ω_L, ω_R the left and right wheel velocity, respectively

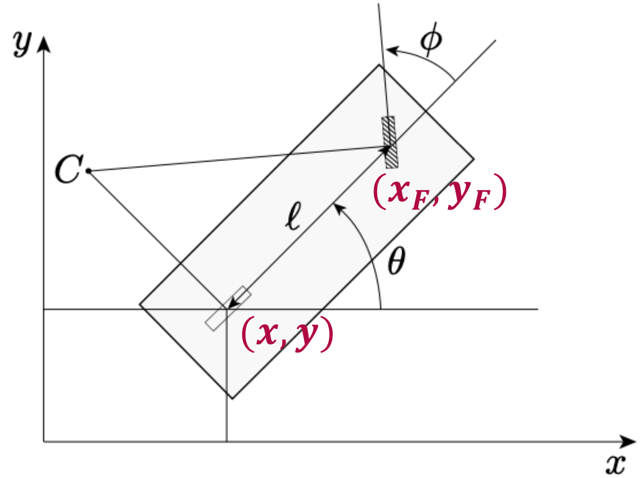
$$v = \frac{\omega_R + \omega_L}{2} \varrho_W, \quad \omega = \frac{\omega_R - \omega_L}{d} \varrho_W$$

$$\omega_R = \frac{2v + \omega d}{2\varrho_W}, \ \omega_L = \frac{2v - \omega d}{2\varrho_W}$$





- Kinematically equivalent to the car-like robot
$$q = [x \quad y \quad \theta \quad \phi]^T$$
- Each wheel as a pure rolling contraint
  - Rear wheel $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$
  - Front wheel $\dot{x}_F \sin(\theta+\phi) - \dot{y}_F\cos(\theta+\phi) = 0$
- The zero motion lines for each wheel intersect at a point C, called instantaneous centre of rotation (that is model-configuration-dependent)
- The following relations bring the pure rolling constraints into the Pfaffian form



$$\begin{cases} x_F = x + l\cos\theta \\ y_F = y + l\sin\theta \end{cases}$$

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

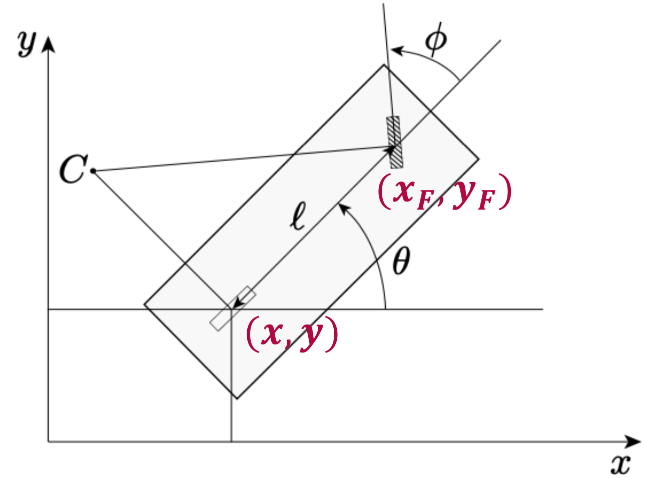
Front wheel $\dot{x}\sin(\theta+\phi) - \dot{y}\cos(\theta+\phi) - l\dot{\theta}\cos\phi = 0$

$$k = 2 \qquad m = n - k = 2$$

$$A^T(q) = \begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 \\ \sin(\theta+\phi) & \cos(\theta+\phi) & -l\cos\phi & 0 \end{bmatrix}$$

$$G(q) = [g_1(q) \quad g_2(q)] \quad u = [u_1 \quad u_2]^T$$

$$\dot{q} = G(q)u = g_1(q)u_1 + g_2(q)u_2$$



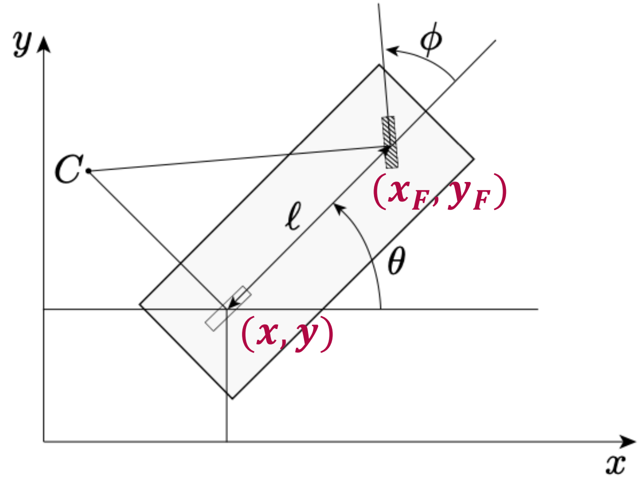
- Find the vector fields of the kinematic model

$$g_1(q) = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi/_l \\ 0 \end{bmatrix} \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- The interpretation of u_1 and u_2 depends on how the vehicle is driven

- Front-wheel drive (fixed wheel is passive)
  - $u_1 = v$ heading velocity
  - $u_2 = \omega$ angular velocity

$$\dot{q} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi/l \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega$$



- Understand whether the constraint is holonomic or not

$$\Delta_A(q) = span\left\{g_1(q), g_2\ \underbrace{[g_1(q), g_2]}_{g_3(q)}, \underbrace{[g_1(q), g_3(q)]}_{g_4(q)}\right\}$$

$$g_3(q) = \begin{bmatrix} \cos\theta\sin\phi \\ \sin\theta\sin\phi \\ -\cos\phi/l \\ 0 \end{bmatrix}, g_4(q) = \begin{bmatrix} -\sin\theta/l \\ \cos\theta/l \\ 0 \\ 0 \end{bmatrix}$$

$$\dim\big(\Delta_A(q)\big) = 4$$

The kinematic model is controllable, hence, the system is completely non-holonomic

- Rear-wheel drive (fixed wheel is active)
  - In this case, the fixed wheel drives the robot: the case is similar to the unicycle
  - To obtain the same unicycle equations in the first two rows, let us write

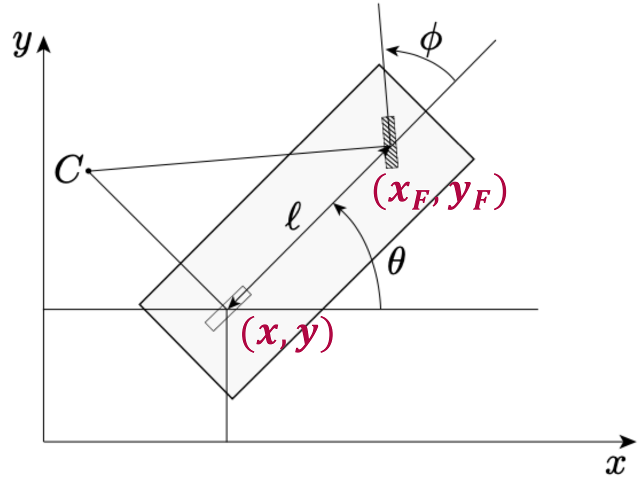
$$u_1 = \frac{v}{\cos\phi}, u_2 = \omega$$

$$\dot{q} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi/l \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

- Understand whether the constraint is holonomic or not

$$\Delta_A(q) = span\left\{g_1(q), g_2\ \underbrace{[g_1(q), g_2]}_{g_3(q)}, \underbrace{[g_1(q), g_3(q)]}_{g_4(q)}\right\}$$

$$g_3(q) = \begin{bmatrix} 0 \\ 0 \\ -1/(l\cos^2\phi) \\ 0 \end{bmatrix}, g_4(q) = \begin{bmatrix} -\sin\theta/(l\cos^2\phi) \\ \cos\theta/(l\cos^2\phi) \\ 0 \\ 0 \end{bmatrix}$$



$$\dim(\Delta_A(q)) = 4$$

The kinematic model is controllable, hence, the system is completely non-holonomic

Equations of motion with nonholonomic constraints

- Notice that the Lagrangian equations of motion $M(q)\ddot{q} + h(q,\dot{q}) = B(q)\tau$ can be seen as $\phi(q,\dot{q},\ddot{q},t) = 0$ which is a holonomic constraint itself
  - If holonomic constraints are present, the constraint force is not included into the equations
- If at least one nonholonomic constraint is present, the Lagrange equations of motion are modified as

$$M(q)\ddot{q} + h(q,\dot{q}) = B(q)\tau + A(q)\lambda$$

$$A^T(q)\dot{q} = 0$$

  - $A(q)$ is the transpose of the Pfaffian matrix
  - $\lambda \in \mathbb{R}^k$ is the vector of Lagrangian multipliers
  - $A(q)\lambda$ represents the reaction forces at the generalised coordinates level
  - $B(q) \in \mathbb{R}^{n\times m}$ maps the $m = n - k$ inputs to the generalised forces performing work on $q \in \mathbb{R}^n$

Calculation of the Lagrange multipliers

- Solve

$$\begin{cases} M(q)\ddot{q} + h(q,\dot{q}) = B(q)\tau + A(q)\lambda & (1) \\ A^T(q)\dot{q} = 0 \end{cases}$$

for the $n + k$ variables q and λ

  - Lagrangian multipliers ensure that there is no motion along the constrained instantaneously prevented directions

- Start differentiating with respect to time the Pfaffian constraints

$$A^T(q)\ddot{q} + \dot{A}(q)^T\dot{q} = 0 \quad (2)$$

- Rewriting (1) as $\ddot{q} = M(q)^{-1}\big(B(q)\tau + A(q)\lambda - h(q,\dot{q})\big)$ and substituting into (2) yields

$$A^T(q)M(q)^{-1}\big(B(q)\tau + A(q)\lambda - h(q,\dot{q})\big) + \dot{A}(q)^T\dot{q} = 0$$

$$A^T(q)M(q)^{-1}A(q)\lambda = -A^T(q)M(q)^{-1}\big(B(q)\tau - h(q,\dot{q})\big) - \dot{A}(q)^T\dot{q}$$

- If the k Pfaffian constraints are independent, then $A^T(q)M(q)^{-1}A(q)$ is full rank

$$\lambda = \big(A^T(q)M(q)^{-1}A(q)\big)^{-1}\big(-A^T(q)M(q)^{-1}\big(B(q)\tau + -h(q,\dot{q})\big) - \dot{A}(q)^T\dot{q}\big)$$

Equations of motion getting rid of the Lagrangian multipliers

- However, it is useful and convenient re-derive the equations of motion without explicitly solving for the Lagrangian multipliers
- Pre-multiply both sides of $M(q)\ddot{q} + h(q,\dot{q}) = B(q)\tau + A(q)\lambda$ by $G^T(q)$

$$G^T(q)M(q)\ddot{q} + G^T(q)h(q,\dot{q}) = G^T(q)B(q)\tau + \underbrace{G^T(q)A(q)\lambda}_{0} \quad (1)$$

from the construction of the kinematic matrix

- From the kinematic model

$$\ddot{q} = \dot{G}(q,\dot{q})u + G(q)\dot{u}$$

$$G^T(q)M(q)\ddot{q} = G^T(q)M(q)\dot{G}(q,\dot{q})u + G^T(q)M(q)G(q)\dot{u} \quad (2)$$

- Comparing (1) with (2) yields

$$\underbrace{G^T(q)M(q)G(q)}_{\bar{M}(q)}\,\dot{u} + \underbrace{G^T(q)M(q)\dot{G}(q)u + G^T(q)h(q,\dot{q})}_{m(q,\dot{q},u)} = G^T(q)B(q)\tau$$

positive-definite and symmetric

Equations of motion getting rid of the Lagrangian multipliers

- We obtain $n + m$ differential equations

$$\begin{cases} \dot{u} = -\bar{M}^{-1}(q)m(q,\dot{q},u) + \bar{M}(q)^{-1}G^T(q)B(q)\tau \\ \dot{q} = G(q)u \end{cases}$$

- Performing the partial feedback linearization $\tau = \left(G^T(q)B(q)\right)^{-1}(\bar{M}(q)a + m(q,\dot{q},u))$ yields

  We suppose that $G^T(q)B(q)$ is always invertible, as it happens in the addressed applications

$$\begin{cases} \dot{u} = a & m \text{ integrators on the input channel (dynamic extensions)} \\ \dot{q} = G(q)u & \text{Kinematic model} \end{cases}$$

  - $a \in \mathbb{R}^m$ is a pseudo-acceleration vector

- Defining $\bar{x} = [q^T \quad u^T]^T \in \mathbb{R}^{n+m}$, the previous system can be written as

$$\dot{\bar{x}} = \begin{bmatrix} G(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} a$$

that is an affine control system, known as second-order kinematic model

Equations of motion getting rid of the Lagrangian multipliers

- The implementation of the partial feedback linearization requires the knowledge of $q, \dot{q}, u$
  - u can be difficult to measure, but it can be reconstructed from the kinematic model

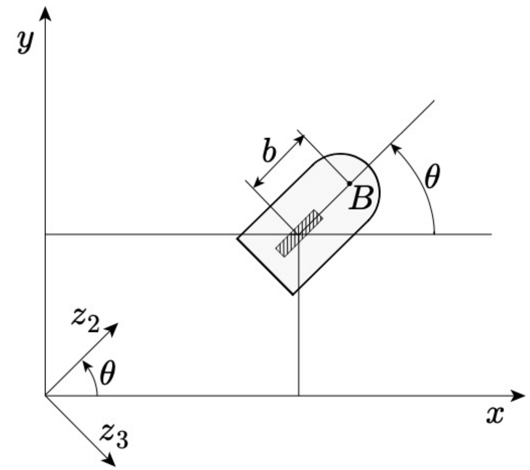
$$\dot{q} = G(q)u \quad \rightarrow \quad u = G^{\dagger}(q)\dot{q} = \left(G^T(q)G(q)\right)^{-1} G^T(q)\dot{q}$$

Dynamic model of the unicycle

- Parameters
  - m mass of the robot
  - J moment of inertia around the vertical axis
  - τ_1 driving torque
  - τ_2 steering torque
- To do all the steps, start writing the following model

$$\begin{cases} M(q)\ddot{q} + h(q,\dot{q}) = B(q)\tau + a(q)\lambda \\ a^T(q)\dot{q} = 0 \end{cases}$$

  - $M(q) = diag\{m, m, J\}$
  - $h(q,\dot{q}) = 0$
  - $B(q) = G(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}$
  - $a^T(q) = [\sin\theta \quad -\cos\theta \quad 0]$

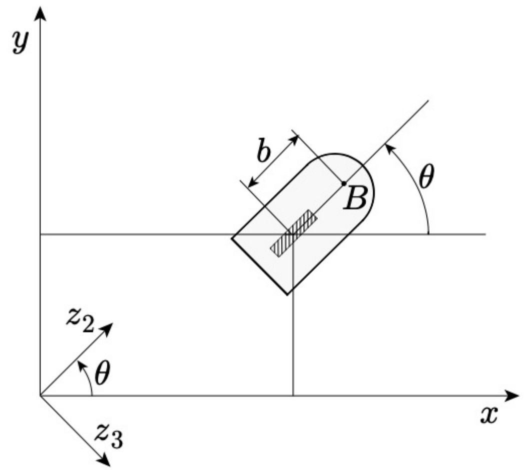


$$\begin{cases} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix}\ddot{q} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}\tau + \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix}\lambda \\ [\sin\theta \quad -\cos\theta \quad 0]\dot{q} = 0 \end{cases}$$

- With the steps seen above, the following model can be achieved

$$\begin{cases} \dot{u} = -\bar{M}^{-1}(q)m(q,\dot{q},u) + \bar{M}(q)^{-1}G^T(q)B(q)\tau \\ \dot{q} = G(q)u \end{cases}$$

  - $\bar{M}(q) = G^T(q)M(q)G(q) = diag(\{m, J\}$
  - $m(q,\dot{q},u) = G^T(q)M(q)\dot{G}(q)u + G^T(q)h(q,\dot{q}) = 0$
  - $G^T(q)B(q) = G^T(q)G(q) = I_2$



$$\begin{cases} \dot{u} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} {}^1/_m & 0 \\ 0 & {}^1/_J \end{bmatrix}\tau \\ \dot{q} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u \end{cases}$$

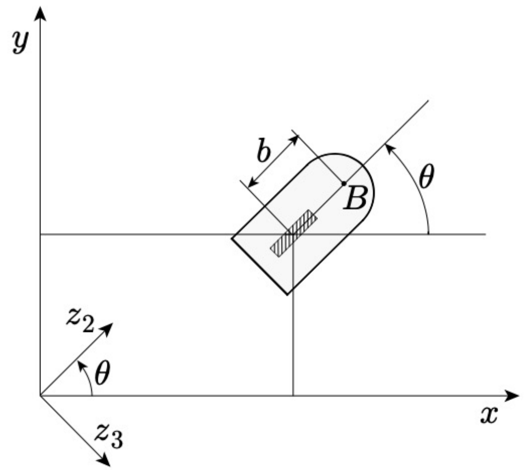
- The following partial feedback linearization is carried out

$$\tau = \left(G^T(q)B(q)\right)^{-1}(\bar{M}(q)a + m(q,\dot{q},u))) = \begin{bmatrix} m & 0 \\ 0 & J \end{bmatrix} a$$

$$\begin{cases} \dot{u} = a \\ \dot{q} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u \end{cases}$$

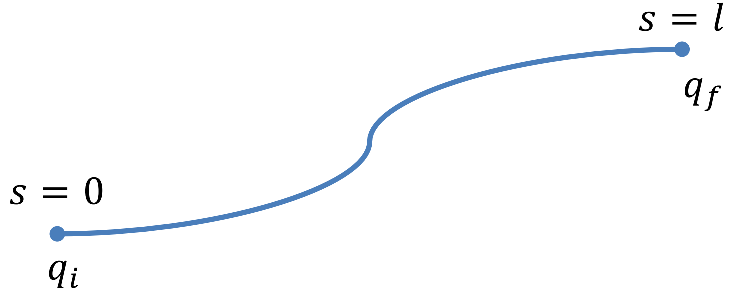
- Defining $\bar{x} = [q^T \quad u^T]^T = [x \quad y \quad \theta \quad v \quad \omega]^T$, the previous system can be written as

$$\dot{\bar{x}} = \begin{bmatrix} G(q)u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} a = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} a$$



Low-level planning problem

- TRAJECTORY = PATH + TIME LAW
  - The path must satisfy the nonholonomic constraints at all points
- Problem
  - *Plan a trajectory* $q(t), \forall t \in [t_i, t_f]$, *leading the robot from* $q(t_i) = q_i$ *to* $q(t_f) = q_f$ *in absence of obstacles and subject to nonholonomic constraints.*
- The trajectory $q(t)$ can be broken down into:
  - Geometric path $q(s)$, with $\frac{dq(s)}{ds} \neq 0, \forall s \in \mathbb{R}$
  - Time law $s = s(t)$, with $s(t_i) = s_i$ and $s(t_f) = s_f$ and monotonic, i.e., $\dot{s}(t) > 0, \forall t \in [t_i, t_f]$
- The variable s represents the path
  - One choice is to chose s as the arclength
  - $l > 0$ is the length of the path



- Having split the trajectory into a geometric path, $q(s)$, plus a time law, $s(t)$, yields

$$\dot{q}(t) = \frac{dq}{dt}\frac{ds}{ds} = \underbrace{\frac{dq}{ds}}_{q'(s)}\frac{ds}{dt} = q'(s)\dot{s}(t)$$

  - $q'(s)$ is a vector tangent to the path into the model-configuration space
  - If s is chosen as the arclength, then $||q'(s)|| = 1$
- The nonholonomic constraints can be written as

$$A^T(q)\dot{q}(t) = A^T(q)q'(s)\dot{s}(t) = 0$$

- Since $\dot{s}(t) > 0$, $\forall t \in [t_i, t_f]$, therefore the only way to satisfy the nonholonomic constraint is

$$A^T(q)q'(s) = 0$$

  - This is the geometric form of a nonholonomic constraint
  - This condition must be verified for all the points on the path $q(s)$
  - In particular, this condition holds for any tangent vector to the path

Geometric admissible paths

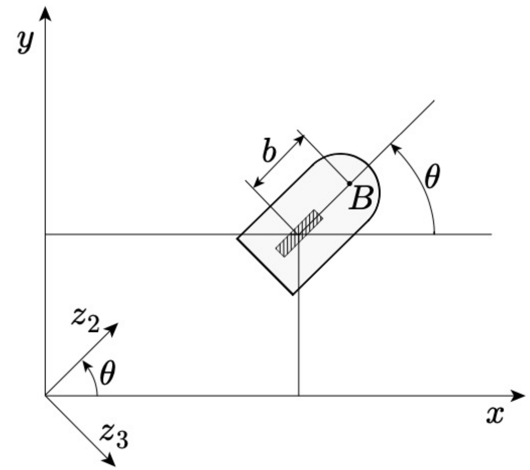
- Geometric admissible paths can be retrieved as we did with the admissible velocities for the derivation of the kinematic model

$$q'(s) = G(q)\tilde{u}(s)$$

  - This is the geometric version of the kinematic model
  - $\tilde{u}(s) \in \mathbb{R}^m$ are geometric inputs
  - $u(t) = \tilde{u}(s)\dot{s}(t)$
- Once the geometric inputs, $\tilde{u}(s)$, are designed for $s \in [s_i, s_f]$, or $s \in [0, l]$ if s is the arclength, the path of the robot in the model-configuration space is uniquely determined
- The choice $s = s(t), t \in [t_i, t_f]$, will identify a particular timing law on the path

Example for the unicycle

- The pure rolling constraint is $\begin{bmatrix}\sin\theta & -\cos\theta & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = 0$
- In its geometric form looks like $\begin{bmatrix}\sin\theta & -\cos\theta & 0\end{bmatrix}\begin{bmatrix}x'\\ y'\\ \theta'\end{bmatrix} = 0$
- The kinematic model for the unicycle is $\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\omega + \begin{bmatrix}\cos\theta\\ \sin\theta\\ 0\end{bmatrix}v$
- The geometric admissible paths are instead $\begin{bmatrix}x'\\ y'\\ \theta'\end{bmatrix} = \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\tilde{\omega} + \begin{bmatrix}\cos\theta\\ \sin\theta\\ 0\end{bmatrix}\tilde{v}$
  - With $\omega(t) = \tilde{\omega}(s)\dot{s}(t)$ and $v(t) = \tilde{v}(s)\dot{s}(t)$



Differential flatness

- Consider an affine control system $(m = 1)$ $\dot{x} = f(x) + g(x)u$ [3]
  - *The system is differentially flat if and only if there exists a so-called flat output* $y = h(x) \in C^n$, *continuous and differentiable function, such that it is possible to express the state and the input as a function of the flat output and its time derivatives*

$$x = \delta\left(y, \dot{y}, \ddot{y}, \dots, y^{(n-1)}\right)$$
$$u = \psi\left(y, \dot{y}, \ddot{y}, \dots, y^{(n)}\right)$$

- Differentially flat systems are useful when explicit trajectory generation is required
  - It is possible to retrieve the full behaviour of the system from the flat output and its time derivatives

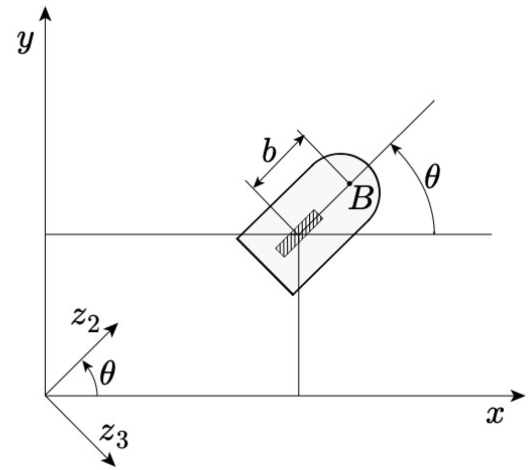
$$u_{ref} = \psi\left(y_{ref}, \dot{y}_{ref}, \ddot{y}_{ref}, \dots, {y_{ref}}^{(n)}\right)$$

- The exact input-state feedback linearizability is a necessary and sufficient condition for flatness
  - Given $\dot{x}(t) = f(x) + g(x)u$, the conditions for the exact input-state feedback linearization are
    - $g, ad_f g, \dots, ad_f^{n-1} g$ are linearly independent vector fields
    - $\Delta = span\{g, ad_f g, \dots, ad_f^{n-2} g\}$ is involutive in a region $\Omega \subseteq \mathbb{R}^n$
  - In this case, the flat output is $y = h(x)$ such that
    - $\frac{\partial h(x)}{\partial x} ad_f^i g = 0, \quad i = 0, \dots, n-2$
    - $\frac{\partial h(x)}{\partial x} ad_f^{n-1} g \neq 0$
  - This can also be extended to control affine systems with m inputs $\dot{x}(t) = f(x) + \sum_{i=1}^{m} g_i(x) u_i$
- Notation for the adjoint operator
  - $ad_f^k g = [f, ad_f^{k-1} g]$
  - $ad_f^0 g = g$

Differential flatness for the unicycle

- The unicycle is a differentially flat system with x and y as flat outputs
- Consider the geometric unicycle model

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tilde{\omega} + \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \tilde{v}$$



- The pair (x, y) are the flat outputs

$$\theta(s) = \mathrm{atan2}\big(y'(s), x'(s)\big) + k\pi\,, k = \{0,1\}$$

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

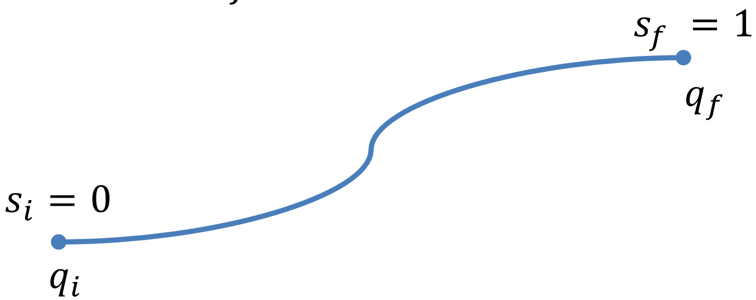
$$\tilde{\omega}(s) = \frac{y''(s)x'(s) - x''(s)y'(s)}{x'(s)^2 + y'(s)^2}$$

  - $k = 0$ forward motion, $k = 1$ backward motion
  - The sign of the geometric heading velocity, $\tilde{v}(s)$, depends on the kind of motion, forward or backward
  - If $x'(\bar{s}) = y'(\bar{s}) = 0$, for some $\bar{s} \in [0, l]$, there is a singularity in $\tilde{\omega}(s)$ since $\tilde{v}(s) = 0$
    - It happens there are cups on the path
    - This also happens if the trajectory shrinks in a point, $l = 0$

- Defining the plan for the flat outputs, x and y, it is possible to reconstruct the other variables algebraically
- The resulting path will automatically embed the nonholonomic constraints since it comes from the geometric version of the pure rolling constraint
- Consider $q_i = [x_i \quad y_i \quad \theta_i]^T$, the initial unicycle's model-configuration, and $q_f = [x_f \quad y_f \quad \theta_f]^T$ the final one: the problem is now to bring the robot from q_i to q_f

Planning via Cartesian polynomials

- Without loss of generality, consider $s_i = 0$ and $s_f = 1$, that is $l = 1$



- Consider cubic polynomials for the flat outputs

$$x(s) = s^3 x_f - (s-1)^3 x_i + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2$$
$$y(s) = s^3 y_f - (s-1)^3 y_i + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2$$

- It is easy to verify

$$x(s_i) = x(0) = x_i$$
$$x(s_f) = x(1) = x_f$$
$$y(s_i) = y(0) = y_i$$
$$y(s_f) = y(1) = y_f$$

- The initial orientation, θ_i, and the final orientation, θ_f, are useful to impose some boundary conditions to retrieve the coefficients $\alpha_x, \alpha_y, \beta_x, \beta_y$

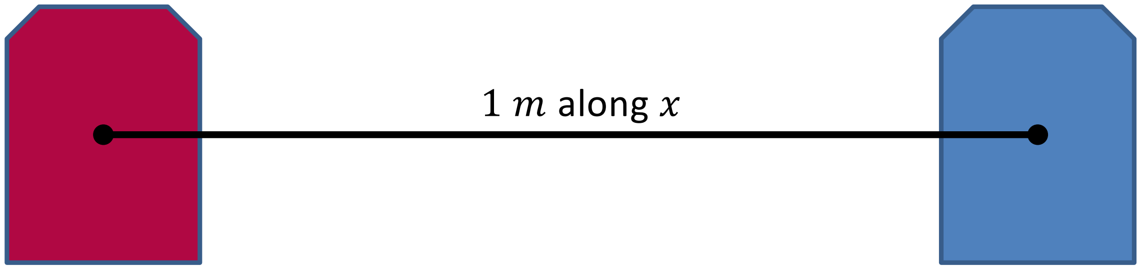
$$x'(s_i) = x'(0) = k_i \cos\theta_i$$
$$y'(s_i) = y'(0) = k_i \sin\theta_i$$
$$x'(s_f) = x'(1) = k_f \cos\theta_f$$
$$y'(s_f) = y'(1) = k_f \sin\theta_f$$

From atan2 expression in $\theta(s)$

  - With $k_i \neq 0$ and $k_f \neq 0$ and $k_i k_f > 0$, depending on the kind of motion chosen on the path (forward or backward)
  - For instance, if $k_i = k_f = k > 0$

$$\alpha_x = k\cos\theta_f - 3x_f$$
$$\alpha_y = k\sin\theta_f - 3y_f$$
$$\beta_x = k\cos\theta_i + 3x_i$$
$$\beta_y = k\sin\theta_i + 3y_i$$

Planning via Cartesian polynomials: Example



$q_i = [x_i \quad y_i \quad \theta_i]^T = \begin{bmatrix}0 & 0 & \frac{\pi}{2}\end{bmatrix}^T$ $\qquad$ $q_f = [x_f \quad y_f \quad \theta_f]^T = \begin{bmatrix}1 & 0 & \frac{\pi}{2}\end{bmatrix}^T$

$$x(s) = s^3 + \alpha_x s^2(s-1) + \beta_x s(s-1)^2$$
$$\mathrm{y}(s) = \alpha_y s^2(s-1) + \beta_y s(s-1)^2$$

$$k = 2$$
$$\alpha_x = -3, \alpha_y = 2, \beta_x = 0, \beta_y = 2$$

$$x'(s) = -6s(s-1)$$
$$y'(s) = 8s(s-1) + 2s^2 + 2(s-1)^2$$
$$\theta(s) = \mathrm{atan2}(y'(s), x'(s))$$

- Once $q(s), s \in [0, l]$, it is now possible to choose a suitable time law, $s = s(t), \mathrm{t} \in [t_i, t_f]$
- Suppose that the unicycle has a maximum velocity for the wheels, $|\omega_L| \leq \omega_{L,max}$ and $|\omega_R| \leq \omega_{R,max}$
  - This is translated into some bounds on the heading and angular velocities $|v(t)| \leq v_{max}$ and $|\omega(t)| \leq \omega_{max}$
- If the chosen time law does not satisfy the previous bounds on the heading and angular velocities, it is necessary to slow down the timing law via uniform scaling
- Rewrite $\tau = {}^{t}/_{T}$ with $T = t_f - t_i$
  - Recall that $\omega(t) = \widetilde{\omega}(s)\dot{s}(t)$ and $v(t) = \tilde{v}(s)\dot{s}(t)$

$$v(t) = \tilde{v}(s)\frac{ds}{dt}\frac{d\tau}{d\tau} = \tilde{v}(s)\frac{ds}{d\tau}\frac{d\tau}{dt} = \tilde{v}(s)\frac{ds}{d\tau}\frac{1}{T}$$

$$\omega(t) = \widetilde{\omega}(s)\frac{ds}{dt}\frac{d\tau}{d\tau} = \widetilde{\omega}(s)\frac{ds}{d\tau}\frac{d\tau}{dt} = \widetilde{\omega}(s)\frac{ds}{d\tau}\frac{1}{T}$$

  - By increasing T, the velocities are reduced uniformly

- Many optimal criteria
  - Curvature → path
  - Energy consumption → path + time law
  - Travel duration → time law
  - Travel length → path
- A simple technique to deal with several optimizations is to over-parameterize the Cartesian polynomials
  - Clearly, the obtained trajectories will be optimal only with respect to the set of trajectories generated by the chosen scheme (i.e., Cartesian polynomials), and it will be a sub-optimal solution of the original problem
- A more systematic approach relies on the optimal control theory

Family of candidate trajectories

- A powerful tool relies on the Pontryagin's minimum principle
  - It is used within the optimal control theory to find the best possible control for taking the system from one state to another in presence of constraints for the state and the control
  - It states that it is necessary for any optimal control along with the optimal state trajectory to solve the so-called Hamiltonian system, which is a two-point boundary value problem, plus a condition on the control Hamiltonian
    - The control Hamiltonian is given by the performance index to be optimized plus a vector of co-state variables, similar to the Lagrangian multipliers but time-variant, multiplied by the system's equations
- Hence, the Pontryagin's minimum principle gives necessary conditions for optimality
  - Using these conditions in a particular problem may lead to a sufficient family of candidate trajectories among which there is certainly the optimal solution (if it exists)

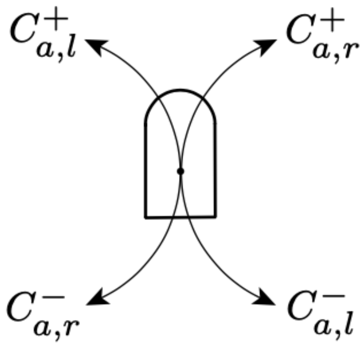
Minimum-time trajectories

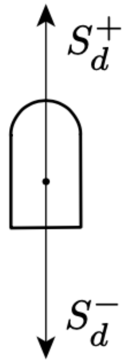
- Consider the problem of moving the unicycle from q_i to q_f in a minimum amount of time
- This can be expressed by the performance index to be minimised

$$J = t_f - t_i = \int_{t_i}^{t_f} dt$$

  - Under the assumption that $\upsilon(t)$ and $\omega(t)$ are bounded
- For this problem, it is possible to find out a sufficient family of candidate trajectories

- This family consists of trajectories obtained by concatenating elementary arcs of two types
  - Arc of circles of variable length covered with velocities $v(t) = \pm v_{max}$ and $\omega(t) = \pm\omega_{max}$
    - The radius of the circle is given by ${}^{v_{max}}/_{\omega_{max}}$
  - Line segments of variable length covered with velocities $v(t) = \pm v_{max}$ and $\omega(t) = 0$





- C_a arc line of duration a
- S_d line segment of duration d
- $+/-$ forward/backward motion
- r/l clockwise/counter-clockwise motion

If $\omega_{max} = 1$ and $v_{max} = 1$, then a and d are also the lengths of the elementary arc/segment, respectively

- The trajectories of the sufficient family solving the minimum-time problem are made through a suitable combination of the elementary arcs/segments
  - They are called Reeds-Shepp curves
  - They are classified into 9 groups

I	$C_a\|C_b\|C_e$	$a \geq 0, b \geq 0, e \geq 0, a+b+e \leq \pi$
II	$C_a\|C_bC_e$	$0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2$
III	$C_aC_b\|C_e$	$0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2$
IV	$C_aC_b\|C_bC_e$	$0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2$
V	$C_a\|C_bC_b\|C_e$	$0 \leq a \leq b, 0 \leq e \leq b, 0 \leq b \leq \pi/2$
VI	$C_a\|C_{\pi/2}S_eC_{\pi/2}\|C_b$	$0 \leq a \leq \pi/2, 0 \leq b \leq \pi/2, e \geq 0$
VII	$C_a\|C_{\pi/2}S_eC_b$	$0 \leq a \leq \pi, 0 \leq b \leq \pi/2, e \geq 0$
VIII	$C_aS_eC_{\pi/2}\|C_b$	$0 \leq a \leq \pi/2, 0 \leq b \leq \pi, e \geq 0$
IX	$C_aS_eC_b$	$0 \leq a \leq \pi/2, 0 \leq b \leq \pi/2, e \geq 0,$

  - The symbol | represents a cusp, that is, a motion inversion on the path

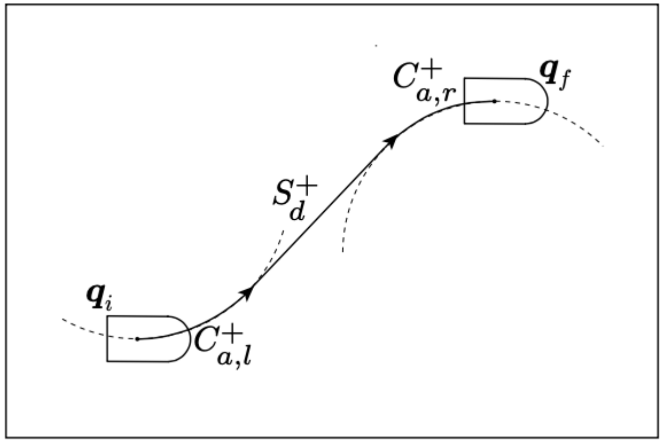
- Each group contains trajectories consisting of a sequence of no more than 5 elementary arcs/segments
- Each group produces a finite number of sequences
  - For example, group IX generate 8 sequences

$$\text{IX} \qquad C_a S_e C_b \qquad\qquad 0 \leq a \leq \pi/2, 0 \leq b \leq \pi/2, e \geq 0,$$

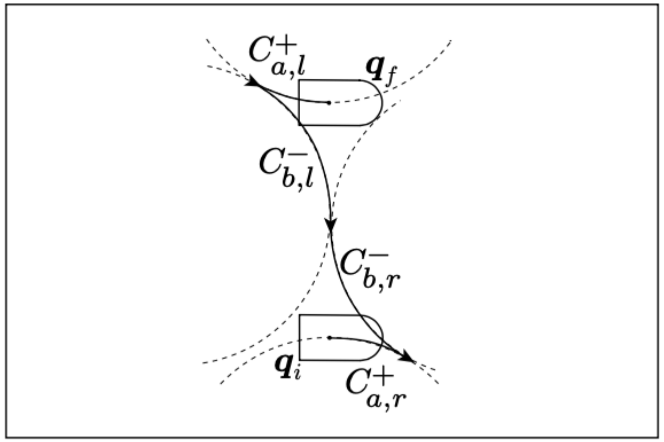
$$C^+_{a,r} S^+_e C^+_{a,r},\ C^+_{a,r} S^+_e C^+_{a,l},\ C^+_{a,l} S^+_e C^+_{a,r},\ C^+_{a,l} S^+_e C^+_{a,l}$$
$$C^-_{a,r} S^-_e C^-_{a,r},\ C^-_{a,r} S^-_e C^-_{a,l},\ C^-_{a,l} S^-_e C^-_{a,r},\ C^-_{a,l} S^-_e C^-_{a,l}.$$

- Making the same for all the 9 groups, there are 48 different sequences
- One may use an exhaustive algorithm to identify the minimum-time trajectory from q_i to q_f
  - 1) Determine all the trajectories belonging to a sufficient family connecting q_i to q_f
  - 2) Compute the value of $J = t_f - t_i$ along these trajectories and choose the one to which the minimum value is associated

- Example of paths



IX group $C^+_{a,l} S^+_d C^+_{a,r}$



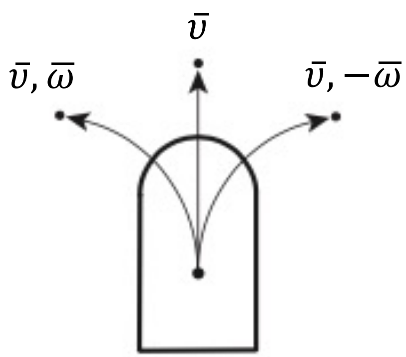
V group $C^+_{a,r}|C^-_{b,r}\ C^-_{b,l}|C^+_{a,l}$

High-level planning problem

- We discovered that a wheeled robot like a differential-drive one cannot move instantaneously along some directions because of the kinematic/non-holonomic constraints
- We have to relax the following assumption seen during the high-level planning problem
  - Suppose also that $\mathfrak{B}$ can instantaneously move everywhere

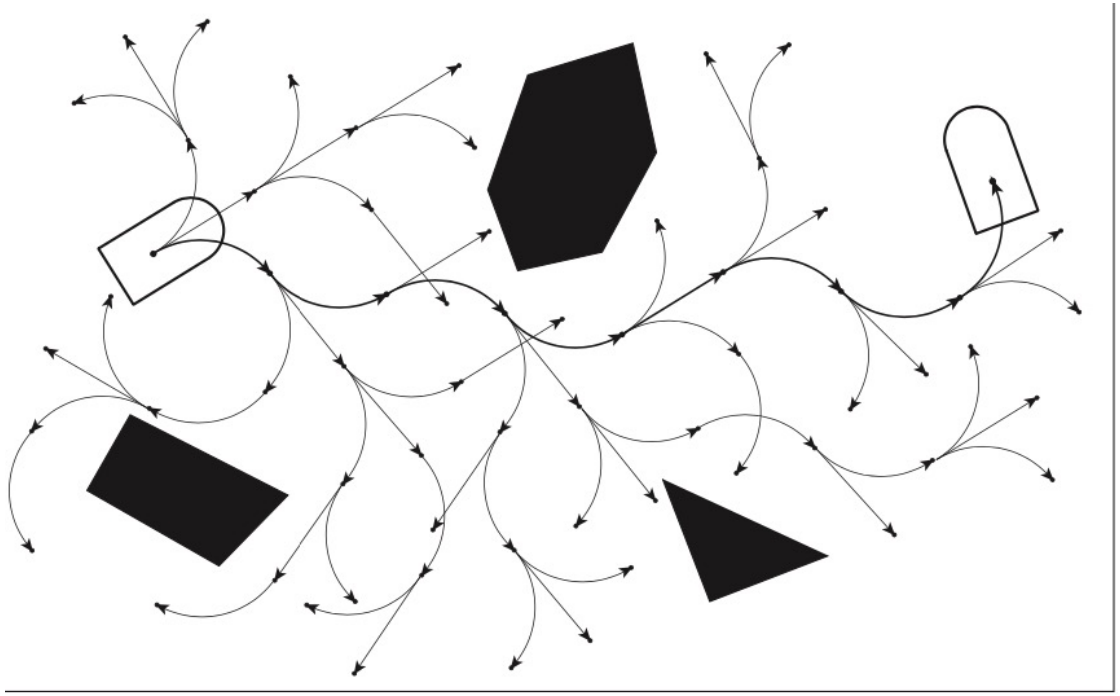
Extension to nonholonomic robots

- PRM and RRT can be applied, as described, to robots that are not subject to any kinematic constraints
  - The rectilinear path can be followed by the unicycle if and only if at the end of each segment they rotate on the spot
- A more general approach consists in the use of motion primitives that are a finite set of admissible local paths in $\mathcal{C}$, each one produced by a given choice of the velocity inputs of the kinematic model
  - Admissible local paths are connections of motion primitives



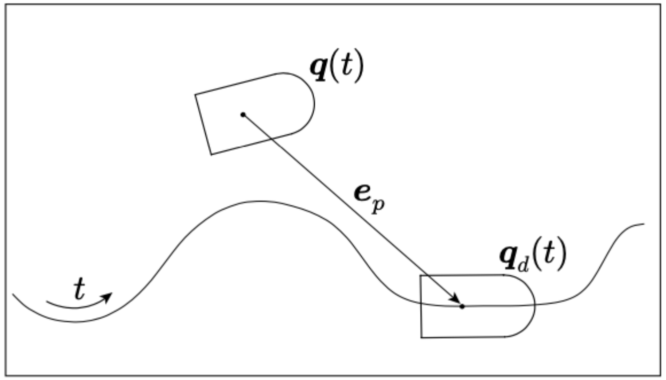
Extension to nonholonomic robots

- In the RRT methodology, once identified q_{near}, q_{new} is generated by applying the motion primitives
  - Other motion primitives as the Reed-Shepp curves can be employed

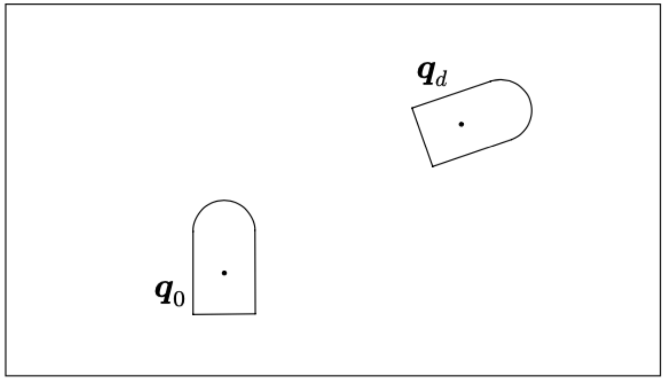


Definition of the problems within the motion control

- The kinematic model is considered for the model-based control problem
  - The dynamic model can be seen as a second-order kinematic model
  - The majority of off-the-shelf commercial robots can be commanded through the wheels' velocities only
- The unicycle is the only robot addressed here
- Tracking and regulation problems



Tracking



Regulation

- The trajectory planning is carried out through the flat outputs
  - It is then known $x_d(t)$ and $y_d(t)$ since the unicycle is a differential flat system
    - The orientation and the kinematic inputs can be reconstructed from the flat outputs

$$\theta_d(t) = \operatorname{atan2}\big(\dot{y}(t), \dot{x}(t)\big) + k\pi, k = \{0,1\}$$

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

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

    - The sign of k and $v_d(t)$ are supposed to be determined a-priori
- The error can be constructed from the current model-configuration at time t, $q(t) = [x(t) \quad y(t) \quad \theta(t)]^T$, and the desired model-configuration at the same time, $q_d(t) = [x_d(t) \quad y_d(t) \quad \theta_d(t)]^T$

$$e(t) = q_d(t) - q(t)$$

  - However, it is better considering the error rotated in the current heading direction, expressed by $\theta(t)$

$$e(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d(t) - x(t) \\ y_d(t) - y(t) \\ \theta_d(t) - \theta(t) \end{bmatrix}$$

- It is possible to take the time derivative of such an error

$$\dot{e}(t) = \dot{\theta}\begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ -\cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_d(t) - x(t) \\ y_d(t) - y(t) \\ \theta_d(t) - \theta(t) \end{bmatrix} + \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{x}_d(t) - \dot{x}(t) \\ \dot{y}_d(t) - \dot{y}(t) \\ \dot{\theta}_d(t) - \dot{\theta}(t) \end{bmatrix} \quad (1)$$

- Recall the kinematic model of the unicycle along the desired trajectory

$$(2) \quad \begin{cases} \dot{x}_d(t) = v_d(t)\cos\theta_d(t) \\ \dot{y}_d(t) = v_d(t)\sin\theta_d(t) \\ \dot{\theta}_d(t) = \omega_d(t) \end{cases}$$

- Substituting (2) into (1) yields

$$\begin{cases} \dot{e}_1(t) = v_d(t)\cos e_3(t) - v(t) + e_2(t)\omega(t) \\ \dot{e}_2(t) = v_d(t)\sin e_3(t) - e_1(t)\omega(t) \\ \dot{e}_3(t) = \omega_d(t) - \omega(t) \end{cases}$$

- It is possible to design the following transformations

$$v(t) = v_d(t)\cos e_3(t) - u_1(t)$$
$$\omega(t) = \omega_d(t) - u_2(t)$$

  - Where $u_1(t)$ and $u_2(t)$ are two virtual control inputs

- We then obtain

$$\begin{cases} \dot{e}_1(t) = \omega_d(t)e_2(t) + u_1(t) - u_2(t)e_2(t) \\ \dot{e}_2(t) = -\omega_d(t)e_1(t) + v_d(t)\sin e_3(t) + u_2(t) \\ \dot{e}_3(t) = u_2(t) \end{cases}$$

$$\dot{e}(t) = \begin{bmatrix} 0 & \omega_d(t) & 0 \\ -\omega_d(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e(t) + \begin{bmatrix} 0 \\ \sin e_3(t) \\ 0 \end{bmatrix} v_d(t) + \begin{bmatrix} 1 & -e_2(t) \\ 0 & e_1(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Linear term

Nonlinear terms

Time-varying terms through the desired kinematic inputs, that are not states or inputs

Control-based on approximate linearization

- A first technique is based on making an approximation around $e(t) = 0$
  - $\sin e_3(t) \approx e_3(t)$
  - $-e_2(t)u_1(t) = 0$
  - $e_1(t)u_2(t) = 0$
- Given the approximations above, we have

$$\dot{e}(t) = \begin{bmatrix} 0 & \omega_d(t) & 0 \\ -\omega_d(t) & 0 & v_d(t) \\ 0 & 0 & 0 \end{bmatrix} e(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

  - This is a time-varying linear system
- The following feedback control is designed

$$u_1(t) = -k_1 e_1(t)$$
$$u_2(t) = -k_2(t) e_2(t) - k_3 e_3(t)$$

  - $k_1, k_3 > 0$

- The closed-loop system is

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

  - The characteristic polynomial of $A(t)$ is $p(\lambda) = \lambda(\lambda + k_1)(\lambda + k_3) + \omega_d(t)^2(\lambda + k_3) + v_d(t)k_2(t)(\lambda + k_1)$
  - Choose

$$k_1 = k_3 = 2\zeta a, \qquad 0 < \zeta < 1, a > 0$$

$$k_2(t) = \frac{a^2 - \omega_d(t)}{v_d(t)}$$

    - Notice that $k_2(t) \to \infty$ as $v_d(t) \to 0$
  - With the choices above, the characteristic polynomial of $A(t)$ is $p(\lambda) = (\lambda + 2\zeta a)(\lambda^2 + 2\zeta a\lambda + a^2)$
    - This has three eigenvalues

$$\lambda_1 = -2\zeta a < 0$$

$\lambda_{2,3}$ are complex conjugates roots with $\mathrm{Re}\{\lambda_{2,3}\} < 0, \omega_n = a$ the natural frequency, and ζ the damping

Control-based on approximate linearization

- However, since this system is time-varying, there is no guarantee that the closed loop is asymptotically stable
- The closed-loop system is indeed asymptotically stable when the desired kinematic inputs, $v_d(t)$ and $\omega_d(t)$, are constant
  - This happens, for instance, along the Reeds-Sheep's curves
- The result is anyway only a local one due to the linearization
  - For very small initial errors, the unicycle will converge asymptotically to the desired path
- Due to the singularity of $k_2(t)$, this method can be applied to persistent trajectories only
  - Persistent trajectories are such that $|v_d(t)| > 0$
  - Motion inversion is thus not allowed

- Recall the following tracking error system

$$\dot{e}(t) = \begin{bmatrix} 0 & \omega_d(t) & 0 \\ -\omega_d(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e(t) + \begin{bmatrix} 0 \\ \sin e_3(t) \\ 0 \end{bmatrix} v_d(t) + \begin{bmatrix} 1 & -e_2(t) \\ 0 & e_1(t) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

- Take the following assumptions around $e(t) = 0$
  - $\sin e_3(t)$ is kept as it is
  - $-e_2(t)u_1(t) = 0$
  - $e_1(t)u_2(t) = 0$
- Given the approximations above, we have

$$\dot{e}(t) = \begin{bmatrix} 0 & \omega_d(t) & 0 \\ -\omega_d(t) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e(t) + \begin{bmatrix} 0 \\ \sin e_3(t) \\ 0 \end{bmatrix} v_d(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

- Component-wise, this is equivalent to

$$
(1) \quad \begin{cases} \dot{e}_1(t) = \omega_d(t)e_2(t) + u_1(t) \\ \dot{e}_2(t) = -\omega_d(t)e_1(t) + v_d(t)\sin e_3(t) \\ \dot{e}_3(t) = u_2(t) \end{cases}
$$

- Consider the following control laws

$$\boxed{\operatorname{sinc}(x) = \frac{\sin x}{x}, \operatorname{sinc}(0) = 1}$$

$$
(2) \quad \begin{aligned} u_1(t) &= -k_1\big(v_d(t), \omega_d(t)\big)e_1(t) \\ u_2(t) &= -k_2 v_d(t)\frac{\sin e_3(t)}{e_3(t)}e_2(t) - k_3\big(v_d(t), \omega_d(t)\big)e_3(t) \end{aligned}
$$

  - $k_1\big(v_d(t), \omega_d(t)\big) > 0$ and $k_3\big(v_d(t), \omega_d(t)\big) > 0$ bounded positive gain functions, and $k_2 > 0$
- Theorem
  - *If $v_d(t)$ and $\omega_d(t)$, with their time derivatives, are bounded and persistent (do not converge to zero), the origin of (1) is asymptotically stable when the controller (2) is applied.*

- Sketch of the proof
  - The closed-loop system is

$$
\begin{cases}
\dot{e}_1(t) = \omega_d(t)e_2(t) - k_1\big(v_d(t),\omega_d(t)\big)e_1(t) \\
\dot{e}_2(t) = -\omega_d(t)e_1(t) + v_d(t)\sin e_3(t) \\
\dot{e}_3(t) = -k_2 v_d(t)\dfrac{\sin e_3(t)}{e_3(t)}e_2(t) - k_3\big(v_d(t),\omega_d(t)\big)e_3(t)
\end{cases}
$$

  - Choose the following Lyapunov function $V = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2} > 0$
  - Its time-derivative is

$$
\begin{aligned}
\dot{V} &= k_2\dot{e}_1e_1 + k_2\dot{e}_2e_2 + \dot{e}_3e_3 \\
&= k_2\cancel{\omega_d e_2}e_1 - k_2k_1(v_d,\omega_d)e_1^2 + k_2\cancel{v_d e_2}\sin e_3 - k_2\cancel{\omega_d e_2}e_1 - k_2\cancel{v_d e_2}\frac{\sin e_3}{\cancel{e_3}}\cancel{e_3} - k_3(v_d,\omega_d)e_3^2 \\
&= -k_2k_1(v_d,\omega_d)e_1^2 - k_3(v_d,\omega_d)e_3^2 \le 0
\end{aligned}
$$

  - $\dot{V}$ is negative semi-definite because e_2 is not missing
  - Since the system is time-varying, Lasalle cannot be used: we can use the Barbalat's lemma

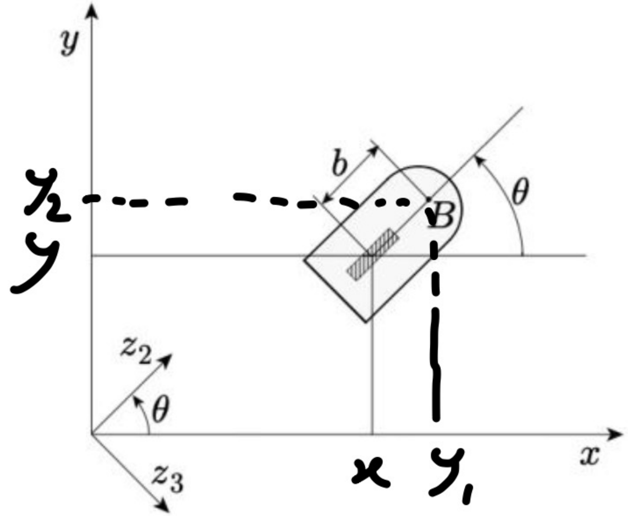
- Barbalat's lemma
  - If we have $\dot{e}(t) = f(e(t), t)$, given a scalar function $V(e(t), t)$ such that
    - $V(e(t), t)$ is lower-bounded
    - $\dot{V}(e(t), t) \leq 0$
    - $\dot{V}(e(t), t)$ is uniformly continuous or $\ddot{V}(e(t), t)$ is bounded
  - Then, $\lim_{t \longrightarrow +\infty} \dot{V}(e(t), t) = 0$

(Almost a) Nonlinear control – cont'd

- Sketch of the proof (cont'd)
  - It can be shown that $\ddot{V}$ is bounded for the addressed closed-loop system
  - Therefore $\lim_{t\longrightarrow+\infty} \dot{V}(e(t),t) = 0$, this means that $e_1(t) \rightarrow 0$ and $e_3(t) \rightarrow 0$
  - It is then possible to show that , since $e_1(t)$ and $e_3(t)$ go to zero, the following relation holds
$$\lim_{t\longrightarrow+\infty} \left(v_d^2(t) + \omega_d^2(t)\right) e_2^2(t) = 0$$
  - This means that $e_2(t) \rightarrow 0$ given the assumption on non-persistent trajectories

- In this case, we linearize the system through feedback
  - We do not start from the error system as the other two techniques
- Let's choose a point B at a distance $|b|$ from the center of the wheel along the sagittal axis
  - $b \neq 0$
    - $b > 0$, the point is ahead
    - $b < 0$, the point is behind
  - (x, y) coordinates of the center of the wheel
  - (y_1, y_2) coordinates of the point B
- Choose as outputs

$$y_1 = x + b\cos\theta$$
$$y_2 = y + b\sin\theta$$



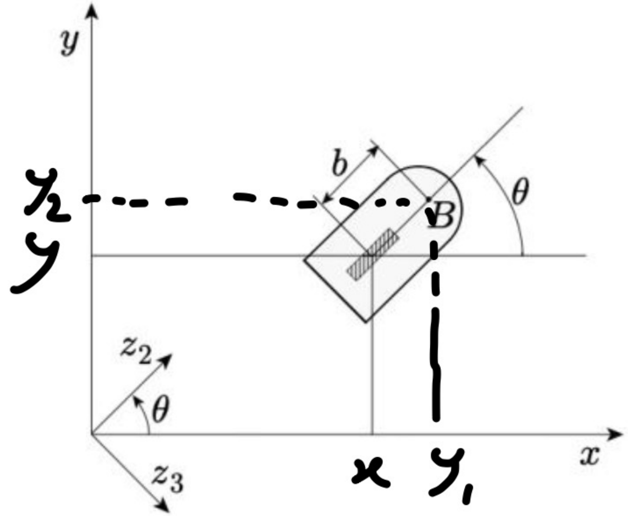
- The time derivatives of these outputs are (substituting the kinematic model of the unicycle)

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{bmatrix}}_{T(\theta)} \begin{bmatrix} v \\ \omega \end{bmatrix} = T(\theta) \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad (1)$$

  - Notice that $\det(T(\theta)) \neq 0 \Leftrightarrow b \neq 0$
  - $T(\theta)$ is invertible only if the distance $|b| \neq 0$
- Given such a condition, we can design

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = T(\theta)^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (2)$$

  - u_1, u_2 two virtual control inputs
- Substituting (2) into (1), we get $\begin{cases} \dot{y}_1 = u_1 \\ \dot{y}_2 = u_2 \end{cases}$



- The following simple controller can be designed

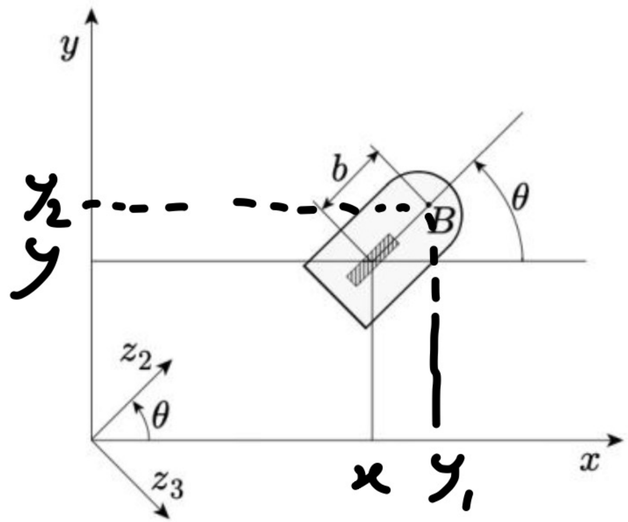
$$u_1 = \dot{y}_{1,d} + k_1\left(y_{1,d} - y_1\right)$$
$$u_2 = \dot{y}_{2,d} + k_2\left(y_{2,d} - y_2\right)$$

  - $k_1, k_2 > 0$
  - This controller guarantees exponential convergence to the desired $y_{1,d}$ and $y_{2,d}$
- Unfortunately, this approach controls the position of the point B only, leaving the orientation uncontrolled

$$\dot{\theta} = \frac{1}{b}\left(u_2 \cos\theta - u_1 \sin\theta\right)$$

- The closed loop equations are

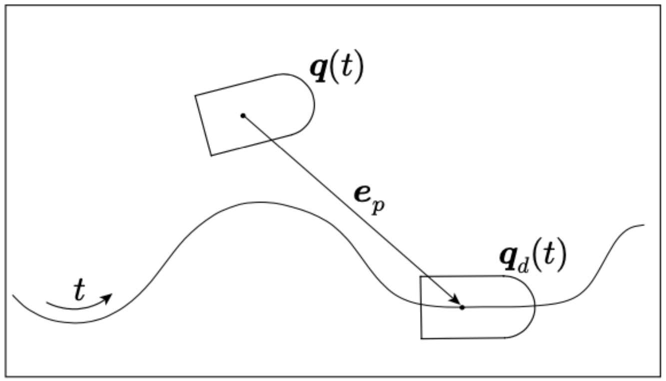
$$\begin{cases} \left(\dot{y}_{1,d} - \dot{y}_1\right) + k_1\left(y_{1,d} - y_1\right) = 0 \\ \left(\dot{y}_{2,d} - \dot{y}_2\right) + k_2\left(y_{2,d} - y_2\right) = 0 \\ \dot{\theta} = \dfrac{\left(\left(\dot{y}_{2,d} + k_2\left(y_{2,d} - y_2\right)\right)\right)\cos\theta - \left(\dot{y}_{1,d} + k_1\left(y_{1,d} - y_1\right)\right)\sin\theta}{b} \end{cases}$$

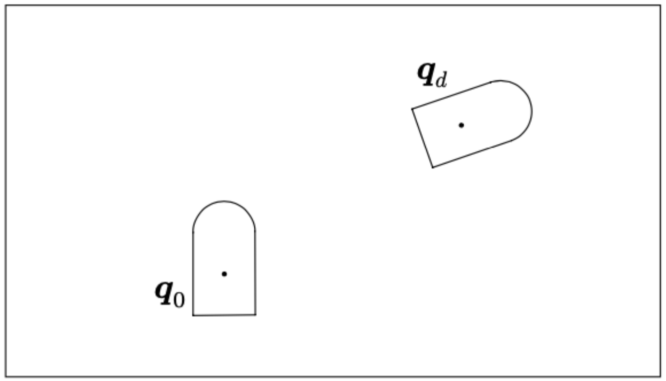


Discussion regarding the trajectory tracking

- The controller based on approximate linearization and the (almost) nonlinear control both require a persistent trajectory
- The input-output linearization control approach does not require a persistent trajectory, but the orientation is uncontrolled
  - Besides, the set-point is given by a point B at a distance $|b| \neq 0$ along the sagittal axis from the vehicle representative point (i.e., the wheel's centre)
- The problem of tracking the full pose (position + orientation) of non-persistent trajectories is structural
  - It is possible to prove that, because of the nonholonomic constraint, the unicycle does not admit ANY universal controller that can asymptotically stabilize arbitrary state trajectories, either persistent or not

- Now, we want to bring the robot from q_0 to q_d without caring of the path or the trajectory





- Through this approach, we will regulate the position of the unicycle only
  - The orientation is free
- Without loss of generality, $q_d = [0 \quad 0 \quad \theta_d]^T$, but θ_d is uncontrollable
- The position error is

$$e_p = [-x \quad -y]^T$$

- Recall the kinematic model of the unicycle, the following regulation controller is designed

$$\begin{cases} v = -k_1(x\cos\theta + y\sin\theta) \\ \omega = k_2(atan2(y,x) + \pi - \theta) \end{cases}$$

  - $k_1, k_2 > 0$
- Interpretation
  - The heading velocity, v, is proportional to the projection of e_p along the sagittal axis
  - The angular velocity, ω, is proportional to the difference between the current unicycle's orientation and the orientation of the vector e_p

- Consider the following scalar function

$$V = \frac{1}{2}(x^2 + y^2) \geq 0$$

  - Positive semi-definite because the orientation is missing
- The time derivative is

$$\dot{V} = x\dot{x} + y\dot{y} = -k_1(x\cos\theta + y\sin\theta)^2 \leq 0$$

  - It is negative semi-definite because the orientation is missing
- It is possible to verify that $\ddot{V}$ is bounded
- Thanks to the Barbalat's lemma, we have

$$\lim_{t\to+\infty} -k_1(x\cos\theta + y\sin\theta)^2 = 0$$

  - This implies $x\cos\theta + y\sin\theta \to 0$, meaning that the projection of e_p along the sagittal axis goes to zero
  - The only point in which the projection of e_p along the sagittal axis is zero can be the origin only

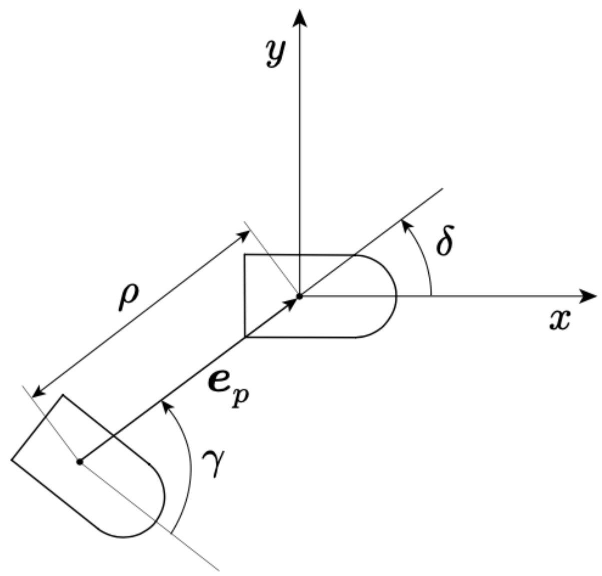
- In fact, a steering velocity, ω, would force the unicycle to rotate on the spot so as to align itself with e_p
- This will cause a change in the sagittal axis and then a change in the heading velocity, v
- The only condition such that the robot stops is when $x\cos\theta + y\sin\theta = 0 \wedge e_p = [0 \quad 0]^T$
- This is true for any initial model-configuration

- Without loss of generality, $q_d = [0 \quad 0 \quad 0]^T$
- Now, we express the problem in polar coordinates
  - $\rho = |e_p|$ is the distance between the unicycle and the origin
  - γ is the angle between e_p and the sagittal axis
  - δ is the angle between e_p and the x −axis
- Then, we have

$$\rho = \sqrt{x^2 + y^2}$$
$$\gamma = atan2(y, x) - \theta + \pi$$
$$\delta = \gamma + \theta$$

- The unicycle model in polar coordinates is

$$\begin{cases} \dot{\rho} = -v \cos \gamma \\ \dot{\gamma} = \dfrac{\sin \gamma}{\rho} v - \omega \\ \dot{\delta} = \dfrac{\sin \gamma}{\rho} v \end{cases}$$



Notice the singularity for $\rho = 0$, the system in polar coordinates is not defined at the target point (!)
One-to-one mapping between Cartesian and polar coordinates is lost in the origin

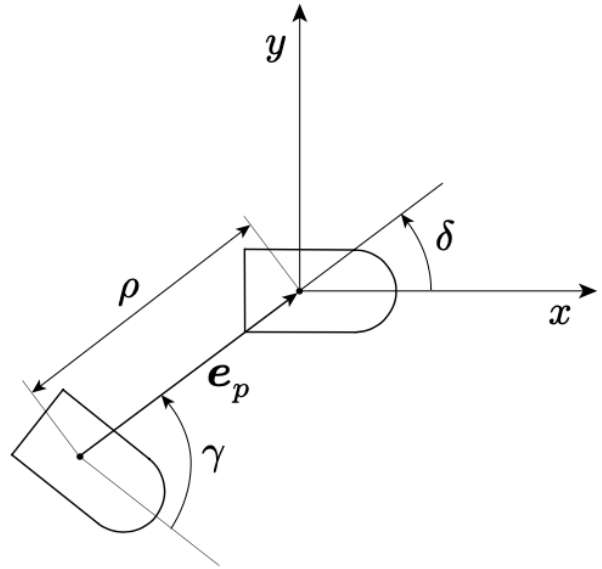
- We can design the following feedback controller

$$v = k_1 \rho \cos\gamma$$

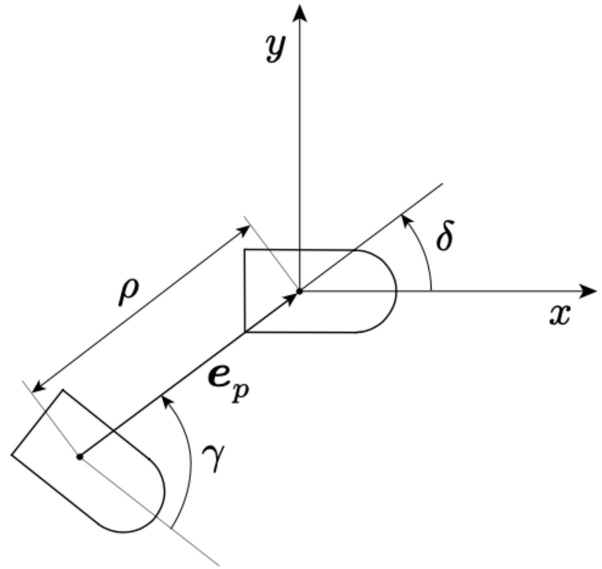
$$\omega = k_2\gamma + k_1 \sin\gamma \cos\gamma \left(1 + k_3 \frac{\delta}{\gamma}\right)$$

  - $k_1, k_2, k_3 > 0$
- This controller makes the system converging to $[\rho \quad \gamma \quad \delta]^T = [0 \quad 0 \quad 0]^T$
- The closed-loop system is

$$\begin{cases} \dot{\rho} = -k_1 \rho \cos^2\gamma \\ \dot{\gamma} = -k_2\gamma - k_1 k_3 \delta \cos\gamma \,\mathrm{sinc}\,\gamma \\ \dot{\delta} = k_1 \sin\gamma \cos\gamma \end{cases}$$



- The sketch of the proof starts from
$$V = \frac{1}{2}(\rho^2 + \gamma^2 + k_3\delta^2) > 0$$
- The time derivative along the closed-loop system's trajectories is
$$\dot{V} = -k_1\rho^2\cos^2\gamma - k_2\gamma^2 \leq 0$$
  - It is negative semi-definite because δ is missing
- It is possible to prove that $\ddot{V}$ is bounded
- Thanks to the Barbalat's leema, $\dot{V} \to 0$ as $t \to +\infty$
- Therefore
$$\lim_{t\to+\infty} -k_1\rho^2\cos^2\gamma - k_2\gamma^2 = 0$$
  - This implies that $\rho \to 0$ and $\gamma \to 0$
  - Further analysis shows that this implies that also $\delta \to 0$



Discussion regarding the regulation problem

- It is possible to bring the unicycle to the desired model-configuration in the full-state through polar coordinates
  - However, the kinematic model is not defined at the target point and the controller (back in the original coordinates) is discontinuous at the origin
- Indeed, it can be proved that ANY feedback control law that can regulate the full pose (position + orientation) of the unicycle must be necessarily discontinuous and/or time-variant
  - Again, this is due to the nonholonomy of the system
- This is due to the Brockett's necessary condition
  - In the particular case of underactuated driftless systems and linearly independent vector fields, the necessary condition is violated, meaning that it does not exist any continuous feedback controller that can asymptotically stabilize an equilibrium point

Problem

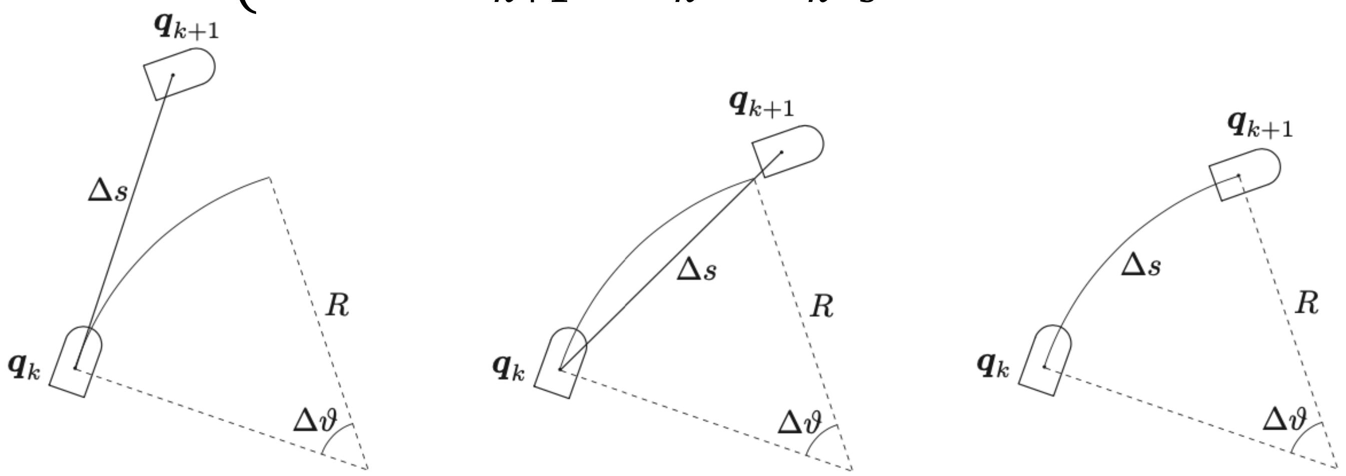
- The implementation of any feedback controller requires the availability of the robot model-configuration at each instant
- However, as said, for mobile robots in general, there is not the possibility to measure directly the position and the orientation with respect to a fixed world frame accurately
- A localization procedure estimating the robot's state must be implemented

- Consider a unicycle with constant heading velocity, v_k, and angular velocity, ω_k, within a time interval $[t_k, t_k + T_s]$
  - $T_s > 0$ is the sampling time
  - If the unicycle is doing an arc of circle, its radius is $r = {}^{v_k}/_{\omega_k}$
  - If the unicycle is doing a line segment, then $\omega_k = 0$
- Let us define $t_{k+1} = t_k + T_s$
- The value of the model-configuration $q(t_{k+1}) = q_{k+1}$ can be retrieved from the unicycle's kinematic model where, as first possibility, we use the Euler's approximation

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

- Another approximation is given by the 2^{nd} order Runge-Kutta approximation

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



- The previous method relies upon the knowledge of v_k and ω_k
  - They are available as command inputs
  - However, the real value of these variables exhibited by the unicycle can be different
- It is thus useful reconstruct v_k and ω_k from the proprioceptive sensors (e.g., wheels' encoders)
- Note that

$$v_k T_s = \Delta s \qquad \omega_k T_s = \Delta\theta$$

$$\frac{v_k}{\omega_k} = \frac{\Delta s}{\Delta\theta} = r$$

- One can measure the displacement of each unicycle's wheel through the left encoder, $\Delta\phi_l$, and the right encoder, $\Delta\phi_r$

$$\Delta\phi_r = \omega_r T_s \qquad \Delta\phi_l = \omega_l T_s$$

- Recalling

$$v_k = \frac{\rho}{2}(\omega_r + \omega_l) \qquad \omega_k = \frac{\rho}{d}(\omega_r - \omega_l)$$

- Once can multiply both sides by T_s

$$v_k T_s = \frac{\rho}{2}(\omega_r + \omega_l)T_s \qquad \omega_k T_s = \frac{\rho}{d}(\omega_r - \omega_l)T_s$$

- Therefore

$$v_k T_s = \frac{\rho}{2}(\Delta\phi_r + \Delta\phi_l) \qquad \omega_k T_s = \frac{\rho}{d}(\Delta\phi_r - \Delta\phi_l)$$

$$v_k = \frac{\rho}{2T_s}(\Delta\phi_r + \Delta\phi_l) \qquad \omega_k = \frac{\rho}{dT_s}(\Delta\phi_r - \Delta\phi_l)$$

Final discussion

- Odometric localization is anyway affected by great errors over big period of time
  - This is caused by slippage, inaccuracy in the kinematic parameters (radius of the wheels, …)
- More robust methods are represented by active localization techniques
  - They rely upon both proprioceptive and exteroceptive sensors (cameras, sonars, LIDAR,…)
  - These techniques make use of Bayesian estimation theory, extended Kalman filters, particle filters, and so on
- The one presented here is a passive localization technique
  - It relies on proprioceptive sensors only