Cooperative mobile robots interacting with their environment Part I : path planning

W. PERRUQUETTI
Non-A INRIA - LNE,
LAGIS UMR CNRS 8146,
Ecole Centrale de Lille, 59650 Villeneuve D'Ascq.
wilfrid.perruquetti@inria.fr.

Special thanks to :
Michael Defoort, Thierry Floquet and Anne Marie Kökösy.

30 june 2011

OUTLINE

1. Collaborative robotics
2. Models for mobile robotics
3. Collaborative path planning

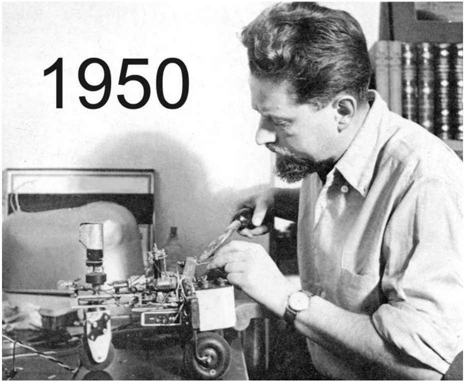
Robotics is a cross fertilizing area which aims at designing and using concrete physical devices with the following capabilities :

- action, (actuators)
- perception, (sensors)
- decision,
- interaction with the environment,

in order to fulfill a task with or without a human.
(The case “not”: human-robot interactions)

Once upon a time . . .

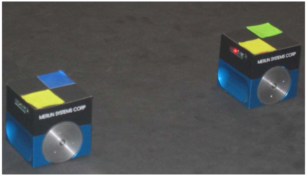
☞ 1950: the turtle created by Grey Walter is probably one of the first autonomous robot.

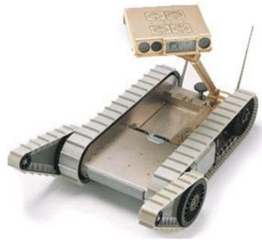


Elsie turtule (tortoise) by Grey Walter

Mobile robotics

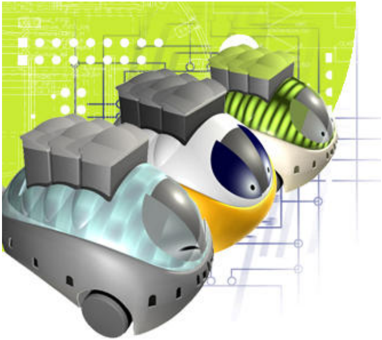
☞ Past: single mobile robots within its environment



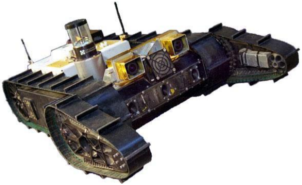


Mobile robotics

Mobile robotics



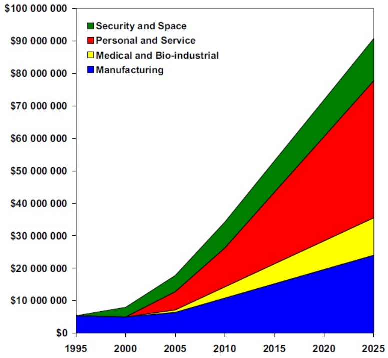
Mobile robotics



Where are we ?

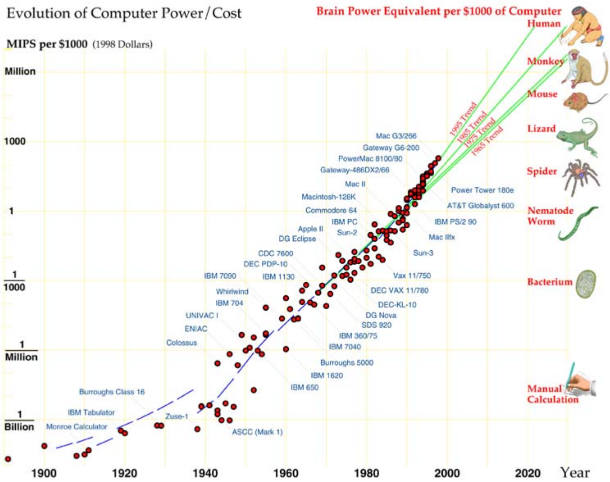
2010: present service robotics.

What are the trends?



2025: service robotics.

What will help ?



2025: Computation capabilities


EXIT
ROBOTO

Réseaux informatiques notamment sans fils ont permis d'entrevoir la séparation de l'ensemble capteurs-commande-actionneurs (CCA).

Conséquences :

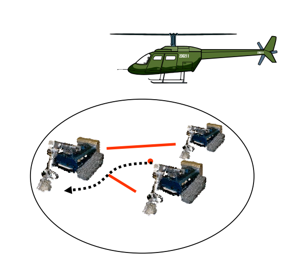
- téléopération de robots, (nouveaux enjeux).
- robots en réseaux : ce sont des dispositifs robotisés (manipulateurs, véhicules mobiles, robots humanoïdes, etc ...) qui sont connectés *via* un réseau de communication tel qu'un réseau local (LAN) ou le réseau internet (WAN) $\rightarrow$ faire coopérer un ensemble de robots.

Nouveaux problèmes : pertes de paquets, retards, QoS etc ...

☞ Robocoop project: http://syner.ec-lille.fr/robocoop

Goals

- Deployment of large scale networks of cooperative mobile robots

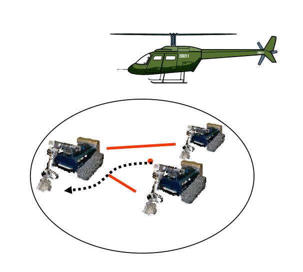


- to get complex behaviors by using simple agent based behaviors

☞ Robocoop project: http://syner.ec-lille.fr/robocoop

Goals

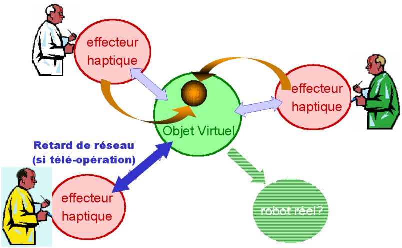
- Deployment of large scale networks of cooperative mobile robots



- to get complex behaviors by using simple agent based behaviors

Applicative fields

- health (tele-robotics, . . .)
- transportation (plane fleet, drones, mobile robots, heterogeneous robots (mobile of different type, planes, underwater robots, . . .)
- security (fire, data collection for “spying”, . . .)
- . . .



Challenges

- local information and decision process,
- constrained communication + delays,
- large scale system,
- uncertain and hostile dynamic environnement,
- ...

Framework: multidisciplinary research

- modeling, path planning and control (constraints, nonlinear models, time delays, hierarchical aspects, hybrid system aspect, quantization . . .)
- graph theory,
- communication protocols,
- logical decision making, scheduling,
- . . .

Framework: multidisciplinary research

- modeling, path planning and control (constraints, nonlinear models, time delays, hierarchical aspects, hybrid system aspect, quantization . . .)
- graph theory,
- communication protocols,
- logical decision making, scheduling,
- . . .

Framework: multidisciplinary research

- modeling, path planning and control (constraints, nonlinear models, time delays, hierarchical aspects, hybrid system aspect, quantization . . .)
- graph theory,
- communication protocols,
- logical decision making, scheduling,
- . . .

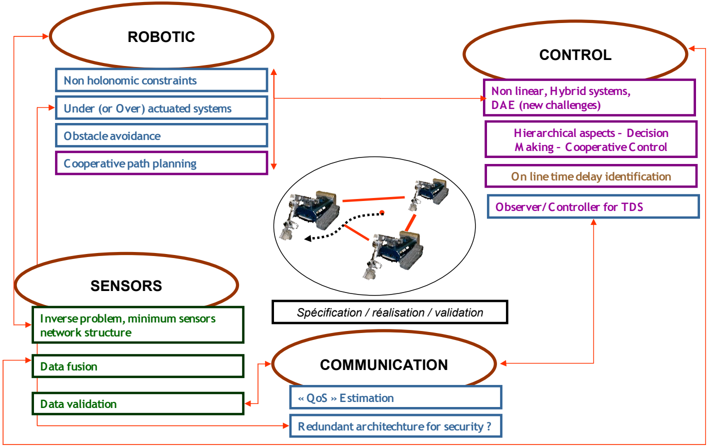
Framework: multidisciplinary research

- modeling, path planning and control (constraints, nonlinear models, time delays, hierarchical aspects, hybrid system aspect, quantization . . .)
- graph theory,
- communication protocols,
- logical decision making, scheduling,
- . . .

Framework: multidisciplinary research

- modeling, path planning and control (constraints, nonlinear models, time delays, hierarchical aspects, hybrid system aspect, quantization ...)
- graph theory,
- communication protocols,
- logical decision making, scheduling,
- ...

Snap shot of Robocoop project / Big picture



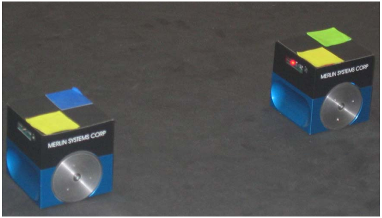
☞ see talk given by Rachid Alami

In a collaborative framework:

- modelling (Part I),
- perception, localization (SLAM: Simultaneous Localization and Mapping, useful for mobile robot),
- decision making (IA),
- task planning: task A $\rightarrow$ task B $\rightarrow$ task C,
- path planning or motion planning (Part I),
- control: trajectory tracking (Part II),

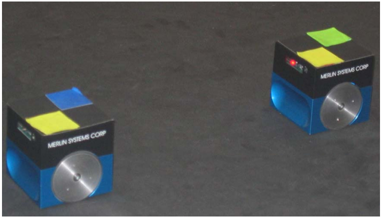
Our Goals

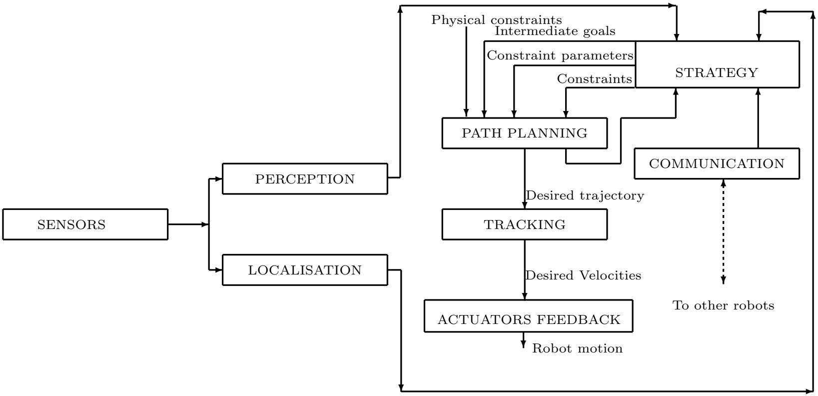
- ✔ path planning and path tracking
- ✔ test on benchmarks



Our Goals

- ✔ path planning and path tracking
- ✔ test on benchmarks




Physical constraints
Intermediate goals
Constraint parameters
Constraints
STRATEGY
PATH PLANNING
COMMUNICATION
PERCEPTION
SENSORS
LOCALISATION
Desired trajectory
TRACKING
Desired Velocities
To other robots
ACTUATORS FEEDBACK
Robot motion

The path planner

- should have real-time capability,
- should be generic (work for all possible WMR),
- should take into account:
  - task (goal),
  - cost (Functional to optimize),
  - **kinematic constraints**: generally a mobile robot can note handle arbitrary displacement.
  - dynamically changing constraints: number of robots, moving obstacles, communication (distance between robots w.r.t a time varying topology), . . .

Leader or not ?

Within a group of mobile robots, some of them may play a particular role: leaders. Distinguish between fleets:

1. with leader: the leader drive the whole fleet or a part of it.
2. without leader: need of a local/global coordination: decision rules must use local informations (most of the time neighbors) or global informations

Questions

☞ How to collect such informations?

☞ What happen if this robot dedicated to data collection is out of order, destroy, or not reliable?

Leader or not ?

Within a group of mobile robots, some of them may play a particular role: leaders. Distinguish between fleets:

1. with leader: the leader drive the whole fleet or a part of it.
2. without leader: need of a local/global coordination: decision rules must use local informations (most of the time neighbors) or global informations

Questions

☞ How to collect such informations?

☞ What happen if this robot dedicated to data collection is out of order, destroy, or not reliable?

Leader or not ?

Within a group of mobile robots, some of them may play a particular role: leaders. Distinguish between fleets:

1. with leader: the leader drive the whole fleet or a part of it.
2. without leader: need of a local/global coordination: decision rules must use local informations (most of the time neighbors) or global informations

Questions

☞ How to collect such informations?
☞ What happen if this robot dedicated to data collection is out of order, destroy, or not reliable?

Questions

☞ How to extract from a graph a minimal representation ensuring some properties (communications, geometric forms of the formation, . . .)?

☞ According to some mission how to choose an initial graph which induces some good properties?

☞ These graphs are time varying (dynamical graphs):

Open questions: analyse, how to control? . . .

Hierarchical structure

To achieve computational tractability:

- "Strategic layer" (higher level): goal planning (for example choose an appropriate functional cost), task scheduling (for example use a petri net for description),
- "Tactical layer" (mid level): guidance, navigation
- "Reflexive layer" (low level): (control) state observation or estimation, trajectory tracking, . . .

Questions

How can we get an "integrated layer" ?

☞ Solve an optimisation problem which integrate some of these facts (gives a path) and then use a good "trajectory tracking"

Hierarchical structure

To achieve computational tractability:

- "Strategic layer" (higher level): goal planning (for example choose an appropriate functional cost), task scheduling (for example use a petri net for description),
- "Tactical layer" (mid level): guidance, navigation
- "Reflexive layer" (low level): (control) state observation or estimation, trajectory tracking, ...

Questions

How can we get an "integrated layer" ?

☞ Solve an optimisation problem which integrate some of these facts (gives a path) and then use a good "trajectory tracking"

Hierarchical structure

To achieve computational tractability:

- "Strategic layer" (higher level): goal planning (for example choose an appropriate functional cost), task scheduling (for example use a petri net for description),
- "Tactical layer" (mid level): guidance, navigation
- "Reflexive layer" (low level): (control) state observation or estimation, trajectory tracking, . . .

Questions

How can we get an "integrated layer" ?
☞ Solve an optimisation problem which integrate some of these facts (gives a path) and then use a good "trajectory tracking"

Hierarchical structure

To achieve computational tractability:

- "Strategic layer" (higher level): goal planning (for example choose an appropriate functional cost), task scheduling (for example use a petri net for description),
- "Tactical layer" (mid level): guidance, navigation
- "Reflexive layer" (low level): (control) state observation or estimation, trajectory tracking, . . .

Questions

How can we get an "integrated layer" ?

☞ Solve an optimisation problem which integrate some of these facts (gives a path) and then use a good "trajectory tracking"

Hierarchical structure

To achieve computational tractability:

- "Strategic layer" (higher level): goal planning (for example choose an appropriate functional cost), task scheduling (for example use a petri net for description),
- "Tactical layer" (mid level): guidance, navigation
- "Reflexive layer" (low level): (control) state observation or estimation, trajectory tracking, . . .

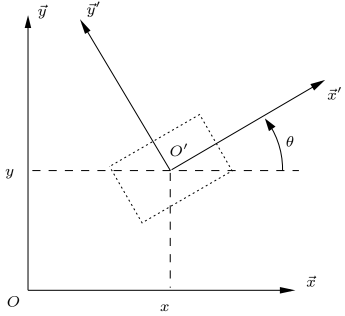
Questions

How can we get an "integrated layer" ?
☞ Solve an optimisation problem which integrate some of these facts (gives a path) and then use a good "trajectory tracking"

Robot = rigid cart equipped with non deformable wheels and moving on a horizontal plan.

The mobile robot position is given in the plane as described in the next picture

$\mathcal{R} = (O, \overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z})$ fixed frame such that (z–axis vertical).
$\mathcal{R}' = (O', \overrightarrow{x}', \overrightarrow{y}', \overrightarrow{z}')$ mobile frame attached to the mobile robot.
$O' \in$ robot: given point (middle of the steering wheels' axis).



Notations

☞ Posture:

$$P = (x, y, \theta)^T$$

belongs to a space $\mathcal{M}$ of dimension $m = 3$ (similar to the work space of a plannar manipulator).

☞ Configuration:

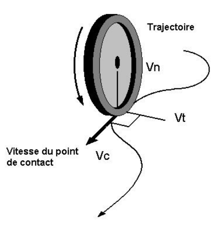
$$q = (q_1, \ldots, q_n)^T$$

n generalised coordinates, belongs to a space $\mathcal{N}$ of dimension n (space configuration).

☞ Mechanical system : position and/or velocity constraint
Integrability ?

- Yes **Holonomic Constraint**: $a\dot{x}_1 = b\dot{x}_2$,
- No **Non Holonomic Constraint**: velocity cannot be removed from these algebraic constraints $\dot{x}_1 \sin(x_3) = \dot{x}_2 \cos(x_3)$.

Example



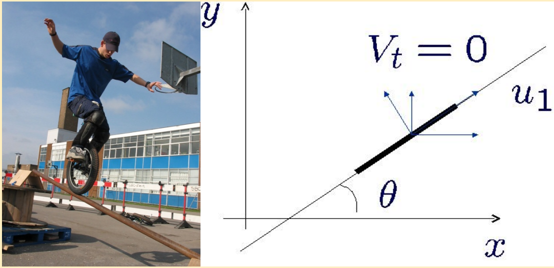
Wheel with single contact point

Hypothesis: pure rolling along the wheel's plan:

- No sliding (dans la direction orthogonale au plan de la roue)
- No skiding (entre la roue et le sol)
- non-deformable wheels of fixed radius r.

Constraints equations: $V_c = 0 \Rightarrow V_t = V_n = 0$.

Example



Wheel with single point contact

Constraints equations:

$$V_t = 0 = \dot{y}\cos(\theta) - \dot{x}\sin(\theta).$$

☞ Non holonomic Constraints: ground/wheel contacts leeds to velocities constraints which are not integrable.

Definition

Non holonomic Constraints in robotics are kinematics conditions which can not be reduced to $g(q,t) = 0$ (only containing the generalized coordinates q and time t).

For WMR these constraints are 1rst ordre non integrable differential equations which can be formulated into a Pfaff form:

$$H(q)\dot{q} = 0, \tag{1}$$

where $\dot{q} = (\dot{q}_1, \ldots, \dot{q}_n)^T$ (generalized velocities)
$H(q) = (h_1(q), h_2(q), \ldots, h_m(q))^T$ an $(m \times n)$–matrix such that all covectors $h_1(q), h_2(q), \ldots, h_m(q)$ are linearly independant and such that $H(q)$ is full rank for all $q \in \mathbb{R}^n$.

☞ The state equation can be rewritten as a driftless system :

$$\dot{x} = G(x)u, x \in \mathbb{R}^n \tag{2}$$

where $G = (g_1(x), g_2(x), \ldots, g_{n-m}(x))^T$ has independant columns $(\text{rang}(G) = n - m)$ and $u = (u_1, u_2, \ldots, u_{n-m})^T$ input vector.

☞ The main assumption is that the Lie algebra rank is n (controllability Lie algebra) which is generated by the Lie brackets of the vector fields $g_1, g_2, \ldots, g_{n-m}$.

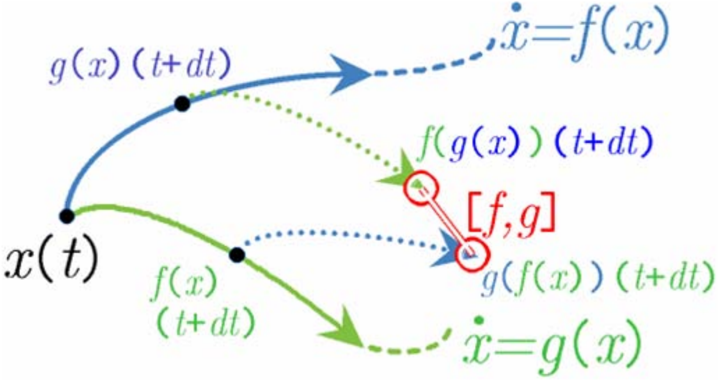
Crochet de Lie

Le **crochet de Lie** (ou commutateur) défini par :

$$[g_1, g_2] = \left(\frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2\right),$$

permet de calculer la condition de commutativité de deux flots $\Phi^t_{g_1}$ et $\Phi^s_{g_2}$.

Crochet de Lie



Chochet de Lie.

Crochet de Lie

Atteignabilité (version locale de la commandabilité) pour

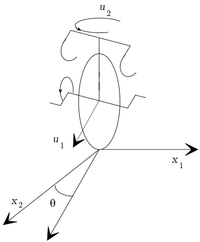
$$\dot{x} = f(x) + \sum_{i=1}^{p} g_i(x)u_i, x \in \mathbb{R}^n$$

pour cela il faut que

$$\mathrm{rang}(\mathcal{A}\{f, g_1, g_2, \ldots, g_p\}) = n,$$

où $\mathcal{A}\{f, g_1, g_2, \ldots, g_p\}$ est l'algèbre de Lie engendrée par les champs de vecteurs $\{f, g_1, g_2, \ldots, g_p\}$.
f est le champs de dérive: il est à noter que les modèles cinématiques que l'on va rencontrer ici sont sans dérive, c'est-à-dire que $f = 0$.

Crochet de Lie



$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \sin\theta \\ \cos\theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2.$$

$$g_1(x) = (\sin(\theta), \cos(\theta), 0)^T, \quad \Phi^t_{g_1} : \begin{pmatrix} x_{10} \\ x_{20} \\ \theta_0 \end{pmatrix} \mapsto \begin{pmatrix} x_{10} + \sin(\theta_0)t \\ x_{20} + \cos(\theta_0)t \\ \theta_0 \end{pmatrix},$$

$$g_2 = (0, 0, 1)^T, \quad \Phi^t_{g_2} : \begin{pmatrix} x_{10} \\ x_{20} \\ \theta_0 \end{pmatrix} \mapsto \begin{pmatrix} x_{10} \\ x_{20} \\ \theta_0 + t \end{pmatrix},$$

$$\Rightarrow \Phi^t_{g_2} \circ \Phi^s_{g_1} \neq \Phi^s_{g_1} \circ \Phi^t_{g_2}$$

Crochet de Lie

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

$$\dim\ (\mathrm{vect}\{[g_1, g_2], g_1, g_2\}) = 3,$$

During the movement, let us assume that:

- le plan de chaque roue reste vertical,
- l'orientation de la roue par rapport au cadre peut être fixe ou variable,
- le contact entre la roue et le sol est réduit à un seul point du plan.

Deux types de contraintes cinématiques doivent être satisfaites en chaque point de la plate-forme mobile, et ceci pour permettre au robot de bouger :

- le long du plan de la roue: la roue roule seulement.
- orthogonal au plan de la roue: non glissement des roues, i.e. la vitesse du robot au long de l'axe orthogonal au plan de la roue est nulle.

☞ Etude pour chaque type de roue donne un système de contraintes.

☞ La non-holonomie des contraintes cinématiques impose des restrictions dans la mobilité du robot.

☞ Parmi toutes les configurations possibles, seulement quelques unes permettent la mobilité du robot en satisfaisant le roulement pur et le non glissement. Pour plus de détails concernant ces restrictions, le lecteur peut se référer à l'article [Cam-97].

☞ Bien évidemment, pour un ensemble de roues donné, toute disposition ne conduit pas à une solution viable. Un mauvais choix peut limiter la mobilité du robot ou occasionner d'éventuels blocages. Par exemple, un robot équipé de deux roues fixes non parallèles ne pourrait pas aller en ligne droite.

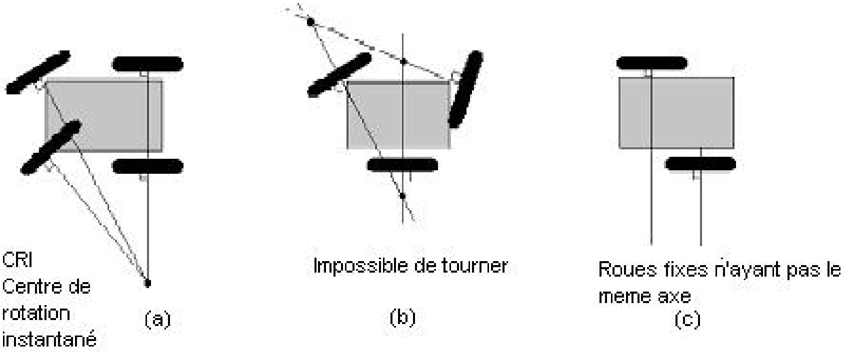
Definition

On appelle centre de rotation instantané (CRI) le point de vitesse nulle liés aux roues autour duquel tourne le robot de façon instantanée.

Proposition

Les points de vitesse nulle liés aux roues se trouvant sur leur axe de rotation, le CRI est le point d'intersection des axes de rotation des roues.

☞ Tous les axes des roues ont pour point d'intersection le Centre de Rotation Instantané CRI ⇒ le vecteur vitesse en chaque point de la structure est orthogonal à la droite liant ce point au CRI.


CRI
Centre de
rotation
instantané
(a)
Impossible de tourner
(b)
Roues fixes n'ayant pas le
meme axe
(c)

CRI.

Proposition

Pour qu'une disposition de roues soit viable et n'entraîne pas de glissement ou dérapage des roues sur le sol, il faut qu'il existe un unique CIR.

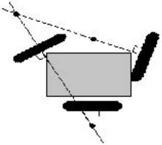
Pour cette raison, il existe en pratique 5 catégories de robots mobiles à roues.

Definition

Le degrès de mobilité δ_m d'un robot est lié au rank le la matrice intervenant dans les contraintes de non holonomies.

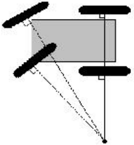
Remark

Le degré de mobilité δ_m est le nombre de degré de liberté du mouvement du robot.



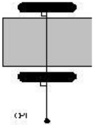
Ne peut se déplacer (pas CRI)

degrè de mobilité 0



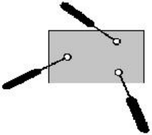
Mouvement en arc fixe (1 seul CRI)

degrè de mobilité 1



Mouvement en arc variable (ligne de CRIs)

degrè de mobilité 2

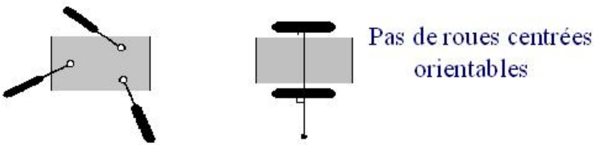


Mobilité totale (CRI se trouve à n'importe quel point)

degrè de mobilité 3

Definition

Let us define the steerability degree δ_s as the number of independant "roues centrées orientables" (en français le degrès de dirigeabilité).



Steerability degree 0

Steerability degree 1 **Steerability degree 2**

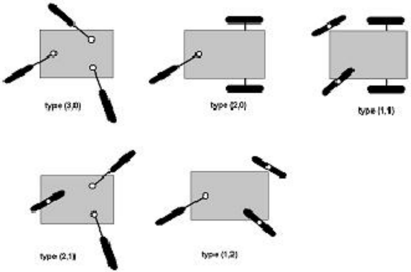
Selon δ_m, deux classes :

- Robots omnidirectionnels : mobilité totale dans le plan $\delta_m = 3$
- Robots à mobilité réduite : degré de mobilté inférieur à 3 $\delta_m < 3$.

Les structures sont désignées par la forme : robot mobile de type (δ_m, δ_s) avec cinq paires des valeurs de δ_m, δ_s vérifiant des inéquations:

$Type$	I	II	III	IV	V
δ_m	3	2	2	1	1
δ_s	0	0	1	1	2

Table: Types de robots mobiles



Examples

Robot de type $(3, 0)$.

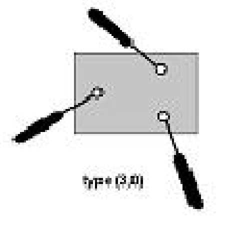
☞ Ces robots n'ont ni de roues conventionnelles fixes ($N_f = 0$), ni de roues centrées orientables ($N_c = 0$).

☞ Le robot est dit "omnidirectionnel" parce qu'il a une totale mobilité dans le plan, i.e. qu'il peut bouger dans n'importe quelle direction sans aucune réorientation à chaque instant. Inversement, les quatre autres types de robots mobiles ont une mobilité réduite.

$$\begin{aligned} \dot{x} &= \cos(\theta)u_1 - \sin(\theta)u_2 \\ \dot{y} &= \sin(\theta)u_1 + \cos(\theta)u_2 \\ \dot{\theta} &= u_3 \end{aligned} \tag{3}$$

Robot de type $(3,0)$.

Example: 3 roues suèdoise



Robot de type $(3,0)$: 3 roues suèdoise

Robot de type $(2,0)$.

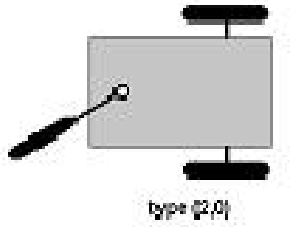
☞ Ces robots n'ont pas de roues conventionnelles centrées orientables ($N_{co} = 0$), mais ils ont une roue conventionnelle fixe ($N_f > 0$), ou même plusieurs mais qui sont montées sur un seul axe commun. Le robot le plus connu et appartenant à cette classe est le robot **Hilare**.

☞ Mobilité réduite.

$$\begin{aligned} \dot{x} &= -\sin(\theta)u_1 \\ \dot{y} &= \cos(\theta)u_1 \\ \dot{\theta} &= u_2 \end{aligned} \tag{4}$$

Robot de type $(2, 0)$.

Example: 2 roues f sur le même axe et 1 roue od



Robot de type $(2, 0)$: 2 roues f sur le même axe et 1 roue od

Robot de type $(2, 1)$.

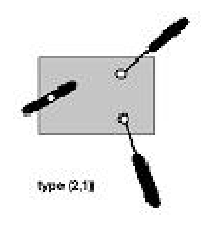
☞ Ces robots n'ont pas de roues conventionnelles fixes. Ils ont une roue conventionnelle orientable centrée et deux roues conventionnelles orientables non-centrées.

☞ Mobilité réduite.

$$\begin{aligned} \dot{x} &= -\sin(\theta + \beta_{c1})u_1 \\ \dot{y} &= \cos(\theta + \beta_{c1})u_1 \\ \dot{\theta} &= u_2 \\ \dot{\beta}_{c1} &= \xi_1 \end{aligned} \tag{5}$$

Robot de type $(2, 1)$.

Example: robot du type $(2, 1)$, avec deux roues conventionnelles orientables décentrées et une roue centrée orientable



Robot de type $(2, 1)$

Robot de type $(1,1)$.

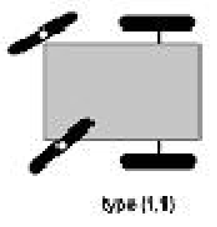
☞ Ces robots ont une ou plusieurs roues conventionnelles avec un seul axe commun et aussi une ou plusieurs roues conventionnelles centrées orientables.

☞ Mobilité réduite.

$$\begin{aligned}
\dot{x} &= -L\sin(\theta)\sin(\beta_{c3})u_1 \\
\dot{y} &= L\cos(\theta)\sin(\beta_{c3})u_1 \\
\dot{\theta} &= \cos(\beta_{c3})u_1 \\
\dot{\beta}_{c3} &= \xi_1
\end{aligned} \tag{6}$$

Robot de type $(1, 1)$.

Example: un robot du type $(1, 1)$, avec deux roues conventionnelles fixe sur le meme axe et une roue centrée orientable, c'est le cas d'un tricycle



Robot de type $(1, 1)$

Robot de type $(1, 2)$.

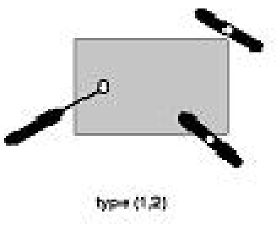
☞ Ces robots n'ont pas de roues conventionnelles fixes. Ils ont au minimum deux roues conventionnelles orientables centrées.

☞ Mobilité réduite.

Complex equations

Robot de type $(1, 2)$.

Example: un robot du type $(1, 2)$, avec deux roues conventionnelles orientables centrées et une roue décentrée orientable



Robot de type $(1, 2)$

Types of models: see [37] "Theory of Robot Control", C. Canudas de Wit, B. Siciliano and G. Bastin (Eds).

Kinematic model (take into account non holonomic constraints)

- Posture Kinematic Model (PKM): the simplest state model which gives a global description of the mobile robot (useful for control). Posture $=(x, y, \theta)$ in most of the case.
- Configuration Kinematic Model (CKM): all the configuration variables (posture + angular position of wheel, ...).

Types of models: see [37] "Theory of Robot Control", C. Canudas de Wit, B. Siciliano and G. Bastin (Eds).

Kinematic model (take into account non holonomic constraints)

- *Posture Kinematic Model (PKM)*: the simplest state model which gives a global description of the mobile robot (useful for control). Posture $=(x, y, \theta)$ in most of the case.
- Configuration Kinematic Model (CKM): all the configuration variables (posture + angular position of wheel, ...).

Types of models: see [37] "Theory of Robot Control", C. Canudas de Wit, B. Siciliano and G. Bastin (Eds).

Kinematic model (take into account non holonomic constraints)

- *Posture Kinematic Model (PKM)*: the simplest state model which gives a global description of the mobile robot (useful for control). Posture $=(x, y, \theta)$ in most of the case.
- Configuration Kinematic Model (CKM): all the configuration variables (posture + angular position of wheel, ...).

Types of models: see [37] “Theory of Robot Control”, C. Canudas de Wit, B. Siciliano and G. Bastin (Eds).

Kinematic model (take into account non holonomic constraints)

- *Posture Kinematic Model (PKM)*: the simplest state model which gives a global description of the mobile robot (useful for control). Posture $=(x, y, \theta)$ in most of the case.
- *Configuration Kinematic Model (CKM)*: all the configuration variables (posture + angular position of wheel, ...).

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- ☞ Configuration Dynamic Model (CDM) : include dynamics of the mobile robots and torques and forces generated by the actuators.
- ☞ Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- Configuration Dynamic Model (CDM) : include dynamics of the mobile robots and torques and forces generated by the actuators.
- Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- ☞ Configuration Dynamic Model (CDM) : include dynamics of the mobile robots and torques and forces generated by the actuators.
- ☞ Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- ☞ Configuration Dynamic Model (CDM) : include dynamics of the mobile robots and torques and forces generated by the actuators.
- ☞ Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- ☞ Configuration Dynamic Model (CDM) : include dynamics of the mobile robots and torques and forces generated by the actuators.
- ☞ Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- ☞ *Configuration Dynamic Model (CDM) :* include dynamics of the mobile robots and torques and forces generated by the actuators.
- ☞ Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- *Configuration Dynamic Model (CDM) :* include dynamics of the mobile robots and torques and forces generated by the actuators.
- Posture Dynamic Model (PDM) : equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Types of models: see [37]

Dynamical model (idem + dynamics induced by actuators (most of the time electrical motors))

- *Configuration Dynamic Model (CDM) :* include dynamics of the mobile robots and torques and forces generated by the actuators.
- *Posture Dynamic Model (PDM) :* equivalent to (CDM) in order to get the EDO (CDM) + 1 $\int$ before each F, τ.

Kinematic model is enough

Generally it is sufficient to use a kinematic model which includes *non holonomic constraints* this is non integrable constraints of the form

$$\boxed{\dot{q} = B(q)u,} \tag{7}$$

where $u \in R^m, q \in R^n$ $(n > m)$.
☞ If not under-actuated (with respect to the mobility degree) then one can perform a feedback linearization :

$$J(q)\dot{u} + C(q,u)u + G(q) = B^T(q)D(q)\Gamma,$$

leads to

$$\dot{u} = v$$

by using $\Gamma = \left(B^T(q)D(q)\right)^{-1}(J(q)v + C(q,u)u + G(q))$.

Kinematic model: flatness is the key point

Flatness (see works from M. Fliess, J.Lévine, Ph.Martin, et P.Rouchon details in [11, 12, 13, 15, 16])

☞ For linear systems

$$\dot{x} = Ax + Bu$$

the following notions are equivalent :

1. controllability,
2. Brunovsky normal form,
3. the parametrization of the state variables and the inputs using m outputs (Brunovsky outputs = flat outputs).

Kinematic model: flatness is the key point

Definition

System

$$\dot{x} = f(x, u), x \in \mathbb{R}^n, u \in \mathbb{R}^m,$$

is **flat** if there exist m functions of the state, the inputs and their derivatives up to order $r \leq n$ (**flat outputs**) such that the state variables and the outputs can be expressed in terms of the flat outputs.

$$\begin{aligned} y &= h(x, u, \dot{u}, \ddot{u}, \ldots, u^{(r)}), \\ x &= \chi(y, \dot{y}, \ldots, y^{(r-1)}), \\ u &= \vartheta(y, \dot{y}, \ldots, y^{(r)}). \end{aligned}$$

Kinematic model: flatness is the key point

Theorem (P. Martin et P. Rouchon [23] (see also [21, 22]))

Any driftless non linear system

$$\dot{x} = B(x)u$$

(which is the case for 7) with m inputs and **at most** *$m+2$ states is flat.*

$\exists$ 3 functions: one defining m *flat outputs* (thus differentially independent) in terms of q, $u, \dot{u}, \ldots, u^{(a)}$) and two other functions one for q the other for u allowing to express them in terms of the output and its time derivatives (in finite number).

Kinematic model: flatness is the key point

- ☞ Thus the PKM and PDM are flat.
- ☞ Thus it implies that they are controllable.
- ☞ But from Brockett's theorem (see [36]) they are not stabilizable by a continuous static time-invariant state feedback.

Kinematic model: flatness is the key point

1. Unicycle mobile robot (type (2,0))

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= w \end{aligned} \tag{8}$$

2. Car-like mobile robot (type (1,1))

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= v\frac{\tan(\phi)}{l} \\ \dot{\phi} &= w \end{aligned} \tag{9}$$

Kinematic model: flatness is the key point

Flat Outputs: (x, y).

Indeed:

1. for (8): $\theta = \arctan\left(\frac{\dot{y}}{\dot{x}}\right), v = \pm\sqrt{\dot{x}^2 + \dot{y}^2}, w = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2}$
2. for (9): $\theta = \arctan\left(\frac{\dot{y}}{\dot{x}}\right), v = \pm\sqrt{\dot{x}^2 + \dot{y}^2}, \phi = \arctan\left(l\frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}\right), w = \dot{\phi}.$

Example: (2,0)-mobile robot

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = w \qquad (10)$$

Flat Outputs: (x, y). Indeed for (8):

$$\theta = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \qquad (11)$$

$$v = \pm\sqrt{\dot{x}^2 + \dot{y}^2} \qquad (12)$$

$$w = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \qquad (13)$$

Example: (2,0)-mobile robot

Path planning: find $y = f(x)$ such that

$$f(x_i) = y_i, f(x_f) = y_f \quad (14)$$

$$f'(x_i) = \tan(\theta_i), f'(x_f) = 0 \quad (15)$$

Polynomial interpolation:

$$f(x) = a_0 + a_1 d + a_2 d^2 + a_3 d^3, d = \frac{x - x_i}{x_f - x_i}$$

$$f'(x) = d'(a_1 + 2a_2 d + 3a_3 d^2), d'(x) = \frac{1}{x_f - x_i}$$

Example: (2,0)-mobile robot

$$\begin{pmatrix} y_i \\ \tan(\theta_i) \\ y_f \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & d' & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & d' & 2d' & 3d' \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/d' & 0 & 0 \\ -3 & -2/d' & 3 & -1/d' \\ 2 & 1/d' & -2 & 1/d' \end{pmatrix} \begin{pmatrix} y_i \\ \tan(\theta_i) \\ y_f \\ 0 \end{pmatrix}$$

Example: (2,0)-mobile robot

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} y_i \\ \frac{\tan(\theta_i)}{d'} \\ 3(y_f - y_i) - 2\frac{\tan(\theta_i)}{d'} \\ -2(y_f - y_i) + \frac{\tan(\theta_i)}{d'} \end{pmatrix}$$

$$\alpha = \frac{\tan(\theta_i)}{d'} = \tan(\theta_i)(x_f - x_i)$$

$$d = \frac{x - x_i}{x_f - x_i}$$

$$y = f(x) = y_i + \alpha d + \left[3(y_f - y_i) - 2\alpha\right] d^2 + \left[-2(y_f - y_i) + \alpha\right] d^3.$$

Example: (2,0)-mobile robot

Needs to find a time parametrization of the flat outputs:

$$x = x^N(t)$$

satisfying the following conditions

$$x^N(t_i) = x_i, x^N(t_f) = x_f \tag{16}$$

$$\dot{x}^N(t_i) = 0, \dot{x}^N(t_f) = 0 \tag{17}$$

Polynomial interpolation:

$$x^N(\tau) = b_0 + b_1\tau + b_2\tau^2 + b_3\tau^3, \tau = \frac{t - t_i}{t_f - t_i}$$

$$\dot{x}^N(\tau) = \dot{\tau}\left(b_1 + 2b_2\tau + 3b_3\tau^2\right), \dot{\tau} = \frac{1}{t_f - t_i}$$

Example: (2,0)-mobile robot

$$\begin{pmatrix} x_i \\ 0 \\ x_f \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \dot{\tau} & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & \dot{\tau} & 2\dot{\tau} & 3\dot{\tau} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\dot{\tau} & 0 & 0 \\ -3 & -2/\dot{\tau} & 3 & -1/\dot{\tau} \\ 2 & 1/\dot{\tau} & -2 & 1/\dot{\tau} \end{pmatrix} \begin{pmatrix} x_i \\ 0 \\ x_f \\ 0 \end{pmatrix} = \begin{pmatrix} x_i \\ 0 \\ 3(x_f - x_i) \\ -2(x_f - x_i) \end{pmatrix}$$

$$x^N(t) = x_i + (x_f - x_i)\tau^2(3 - 2\tau), \tau = \frac{t - t_i}{t_f - t_i}$$

Example: (2,0)-mobile robot

Open loop control:

$$v = \pm\sqrt{\dot{x}^2 + \dot{y}^2} \quad (18)$$

$$w = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \quad (19)$$

$$y^N(t) = f(x^N(t)), \quad (20)$$

$$\dot{y}^N(t) = \dot{x}^N(t) f'(x^N(t)) \quad (21)$$

$$\ddot{y}^N(t) = \ddot{x}^N(t) f'(x^N(t)) + \dot{x}^{N2}(t) f''(x^N(t)) \quad (22)$$

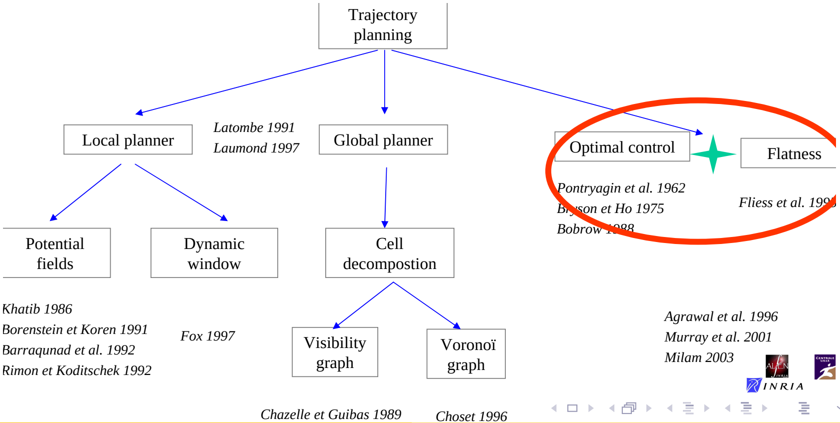
$$v^N(t) = \dot{x}^N(t)\sqrt{1 + f'^2(x^N(t))} \quad (23)$$

$$w^N(t) = \frac{f''(x^N(t))}{1 + f'^2(x^N(t))} \quad (24)$$

Path planning for a single robot

Path planning

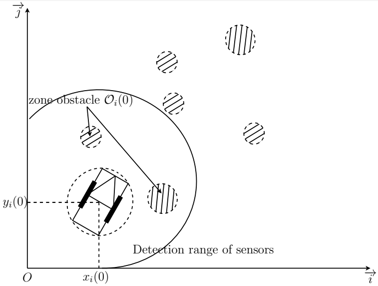
Computation of an executable collision-free trajectory for a robot between an initial given configuration and a final given configuration

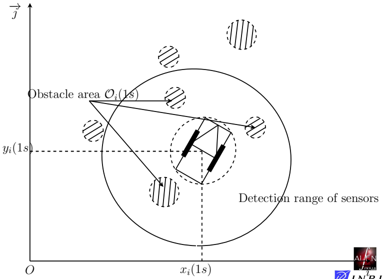


Path planning for a single robot: problem setup

Assumptions and notations

- Geometrical shape of robot i: circle of centre (x_i, y_i) and radius ρ_i
- Geometrical shape of j^{th} obstacle: circle of centre O_j and of radius r_j, denoted $\mathcal{B}_j(O_j, r_j)$
- Set of admissible inputs: $\mathcal{U}_i$
- Obstacle set: subset $\mathcal{O}_i(t_k) \subset \{\mathcal{B}_0(O_0, r_0), \ldots\}$ of M_i obstacles in the range of robot sensors at the time instance t_k





Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:
$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $$\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:
$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:
$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $$\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:
$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $$\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq$ $\rho_i + r_{m_i}$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:
$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $$\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:

$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $$\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

$\rightleftarrows$

$q_i = \varphi_1(z_i, \dot{z}_i, \ddot{z}_i)$

$u_i = \varphi_2(z_i, \dot{z}_i, \ddot{z}_i)$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Criteria:
$J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $$\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

Criteria: $J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i), \varphi_2(z_i, \dot{z}_i, \ddot{z}_i), t)dt$
wrt: $\forall t \in [t_{initial}, t_{final}]$,

- ...
- $$\begin{cases} \varphi_1(z_i(t_{initial}), \ddot{z}_i(t_{initial})) = q_{i,initial} \\ \varphi_1(z_i(t_{final}), \ddot{z}_i(t_{final})) = q_{i,final} \\ \varphi_2(z_i(t_{initial}), \ddot{z}_i(t_{initial})) = u_{i,initial} \\ \varphi_2(z_i(t_{final}), \ddot{z}_i(t_{final})) = u_{i,final} \end{cases}$$
- etc.

$\min J = \int_{t_{initial}}^{t_{final}} L_i(q_i, u_i, t)dt$

$\min J = \int_{t_{initial}}^{t_{final}} L_i(\varphi_1(z_i, \dot{z}_i, \ddot{z}_i),$

Single robot: off-line algorithm

Dynamic optimisation based on flatness

Optimal control + Flatness → Dynamic optimization based on flatness

Resolution of optimal control problems

☞ Transformation into a nonlinear programming problem, using B-spline functions in order to approximate the trajectory of the flat output

☞ Computation of optimal control points using an optimisation procedure (CFSQP)

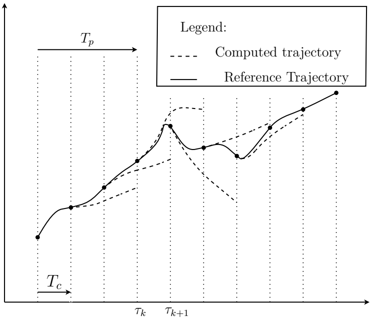
☞ Computation of the corresponding control inputs using the flatness properties of the system

Single robot: on-line algorithm

Main Principle

☞ To relax the constraint that the final point is reached during the planning horizon, allowing the use of an on-line receding horizon path planner

- $T_p (> 0)$: planning horizon
- $T_c (> 0)$: update period
- $\tau_k (k \in \mathbb{N}, \tau_k = t_{initial} + kT_c)$: updates



Single robot: on-line algorithm

Implementation

- initialisation step: computations before the movement of the robot
- step of iterative computations: computations over any interval $[\tau_{k-1}, \tau_k)$

$$
\begin{array}{rcl}
\min J_{\tau_k} & = & c \left\| q_i(\tau_k + T_p, \tau_k) - q_{i,final} \right\|^2 \\
 & & + \int_{\tau_k}^{\tau_k + T_p} L_i(q_i(t, \tau_k), u_i(t, \tau_k), t) dt \\
 & & (c > 0) \text{ slc } \forall t \in [\tau_k, \tau_k + T_p] : \\
 & & \left\{ \begin{array}{lcl}
 \dot{q}_i(t, \tau_k) & = & f_i(q_i(t, \tau_k), u_i(t, \tau_k)) \\
 q_i(\tau_k, \tau_k) & = & q_{i,ref}(\tau_k, \tau_{k-1}) \\
 u_i(\tau_k, \tau_k) & = & u_{i,ref}(\tau_k, \tau_{k-1}) \\
 u_i(t, \tau_k) & \in & \mathcal{U}_i \\
 d(q_i(t, \tau_k), O_{m_i}) & \geq & \rho_i + r_{m_i}, \ \forall O_{m_i} \in \mathcal{O}_i(\tau_k)
 \end{array} \right.
\end{array}
$$

Single robot: on-line algorithm

Implementation

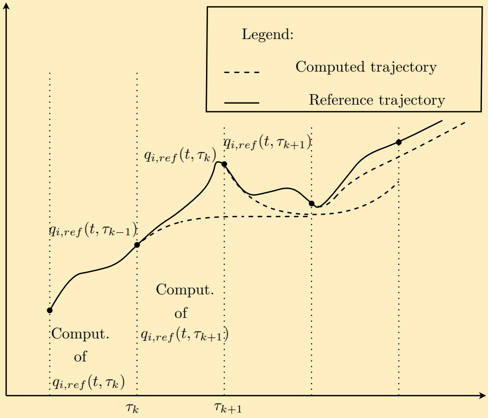
☞ initialisation step: computations before the movement of the robot

☞ step of iterative computations: computations over any interval $[\tau_{k-1}, \tau_k)$

$$
\begin{array}{rcl}
\min J_{\tau_k} & = & c\left\|q_i(\tau_k+T_p,\tau_k)-q_{i,final}\right\|^2 \\
& & +\int_{\tau_k}^{\tau_k+T_p} L_i(q_i(t,\tau_k),u_i(t,\tau_k),t)dt \\
& & (c>0) \text{ slc } \forall t\in[\tau_k,\tau_k+T_p]: \\
& & \left\{\begin{array}{lcl}
\dot q_i(t,\tau_k) & = & f_i(q_i(t,\tau_k),u_i(t,\tau_k)) \\
q_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\
u_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\
u_i(t,\tau_k) & \in & \mathcal{U}_i \\
d(q_i(t,\tau_k),O_{m_i}) & \geq & \rho_i + r_{m_i},\ \forall O_{m_i}\in\mathcal{O}_i(\tau_k)
\end{array}\right.
\end{array}
$$

Single robot: on-line algorithm

Implementation



Single robot: numerical examples

Criteria

Minimisation of travelling time

Off-line

☞ Global knowledge

Video

Comput. Time	Optimum
4min	12.6s

Single robot: numerical examples

Criteria

Minimisation of travelling time

Off-line

☞ Global knowledge

Video

Comput. Time	Optimum
4min	12.6s

On-line

☞ Local knowledge

Video

Comput. Time	Optimum
50ms	15s

Multi-robots coordination

Objective

☞ To generate a (sub) optimal trajectory for each robot which satisfy:

- terminal constraints
- physical constraints (nonholonomic, maximum velocities, ...)
- obstacle avoidance
- minimum distances between robots (collision avoidance)
- maximum distances between robots (respect of the broadcasting range)

Communication graph $(\mathcal{N}, \mathcal{A}, \mathcal{S})$

- Robots $\mathcal{N} = \{1, \ldots, N_a\}$
- Edges $\mathcal{A} \subset \mathcal{N} \times \mathcal{N} \leftrightarrows$ communication links
- Constraints of the edges

$d_{i,com} \in \mathbb{R}^+$: broadcasting range of robot i

Multi-robots coordination

Objective

☞ To generate a (sub) optimal trajectory for each robot which satisfy:

- terminal constraints
- physical constraints (nonholonomic, maximum velocities, . . .)
- obstacle avoidance
- minimum distances between robots (collision avoidance)
- maximum distances between robots (respect of the broadcasting range)

Communication graph $(\mathcal{N}, \mathcal{A}, \mathcal{S})$

- Robots $\mathcal{N} = \{1, \ldots, N_a\}$
- Edges $\mathcal{A} \subset \mathcal{N} \times \mathcal{N} \leftrightarrows$ communication links
- Constraints of the edges
  $d_{i,com} \in \mathbb{R}^+$: broadcasting range of robot i

Multi-robots coordination

Objective

☞ To generate a (sub) optimal trajectory for each robot which satisfy:

- terminal constraints
- physical constraints (nonholonomic, maximum velocities, . . .)
- obstacle avoidance
- minimum distances between robots (collision avoidance)
- maximum distances between robots (respect of the broadcasting range)

Communication graph $(\mathcal{N}, \mathcal{A}, \mathcal{S})$

- Robots $\mathcal{N} = \{1, \ldots, N_a\}$
- Edges $\mathcal{A} \subset \mathcal{N} \times \mathcal{N} \leftrightarrows$ communication links
- Constraints of the edges
  $d_{i,com} \in \mathbb{R}^+$: broadcasting range of robot i

Multi-robots coordination

Objective

☞ To generate a (sub) optimal trajectory for each robot which satisfy:

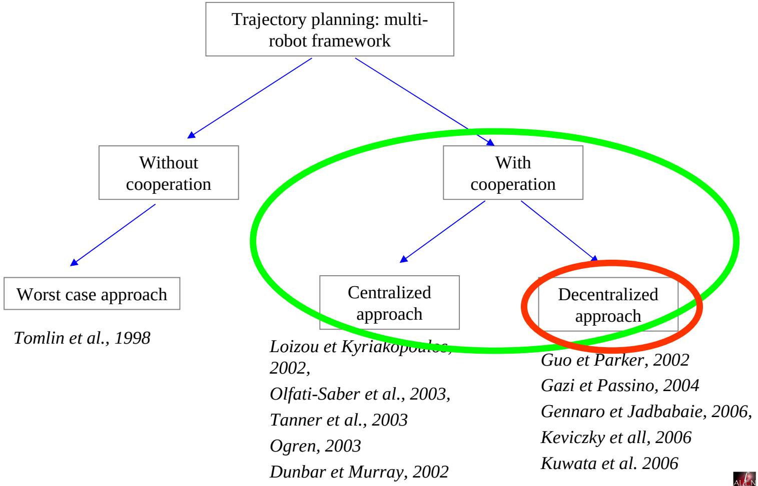
- terminal constraints
- physical constraints (nonholonomic, maximum velocities, ...)
- obstacle avoidance
- minimum distances between robots (collision avoidance)
- maximum distances between robots (respect of the broadcasting range)

Communication graph $(\mathcal{N}, \mathcal{A}, \mathcal{S})$

- Robots $\mathcal{N} = \{1, \ldots, N_a\}$
- Edges $\mathcal{A} \subset \mathcal{N} \times \mathcal{N} \leftrightarrows$ communication links
- Constraints of the edges

  $d_{i,com} \in \mathbb{R}^+$: broadcasting range of robot i

Multi-robots coordination



Multi-robots coordination: centralized approach

Extension of the on-line algorithm proposed for a single robot

☞ Resolution via a supervisor (independent unit or a single robot of the formation)

- initialisation step
- step of iterative computations:

$$\min J_{\tau_k} = \left\| q_i(\tau_k + T_p, \tau_k) - q_{i,final} \right\|^2 + c \int_{\tau_k}^{\tau_k + T_p} L_i(q_i(t,\tau_k), u_i(t,\tau_k), t) dt$$

$(c \geq 0)$ wrt: $\forall t \in [\tau_k, \tau_k + T_p]$,

$$\begin{cases} \dot{q}_i(t,\tau_k) & = & f_i(q_i(t,\tau_k), u_i(t,\tau_k)) \\ q_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(t,\tau_k) & \in & \mathcal{U}_i \\ d(q_i(t,\tau_k), O_{m_i}) & \geq & \rho_i + r_{m_i}, \quad \forall O_{m_i} \in \mathcal{O}_i(\tau_k) \\ d(q_i(t,\tau_k), q_p(t,\tau_k)) & \leq & \min(d_{i,com}, d_{p,com}), \quad \forall p \in \mathcal{N} \text{ tel que } (i,p) \in \mathcal{A} \\ d(q_i(t,\tau_k), q_{p'}(t,\tau_k)) & > & \rho_i + \rho_{p'}, \quad \forall p' \in \mathcal{N} \text{ tel que } p' \neq i \end{cases}$$

Resolution of optimal control problems

☞ Dynamic optimisation based on flatness

Multi-robots coordination: centralized approach

Extension of the on-line algorithm proposed for a single robot

☞ Resolution via a supervisor (independent unit or a single robot of the formation)

- initialisation step
- step of iterative computations:

$$\min J_{\tau_k} = \left\| q_i(\tau_k + T_p, \tau_k) - q_{i,final} \right\|^2 + c \int_{\tau_k}^{\tau_k + T_p} L_i(q_i(t,\tau_k), u_i(t,\tau_k), t)dt$$

$(c \geq 0)$ wrt: $\forall t \in [\tau_k, \tau_k + T_p]$,

$$\begin{cases} \dot{q}_i(t,\tau_k) & = & f_i(q_i(t,\tau_k), u_i(t,\tau_k)) \\ q_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(t,\tau_k) & \in & \mathcal{U}_i \\ d(q_i(t,\tau_k), O_{m_i}) & \geq & \rho_i + r_{m_i}, \quad \forall O_{m_i} \in \mathcal{O}_i(\tau_k) \\ d(q_i(t,\tau_k), q_p(t,\tau_k)) & \leq & \min(d_{i,com}, d_{p,com}), \quad \forall p \in \mathcal{N} \text{ tel que } (i,p) \in \mathcal{A} \\ d(q_i(t,\tau_k), q_{p'}(t,\tau_k)) & > & \rho_i + \rho_{p'}, \quad \forall p' \in \mathcal{N} \text{ tel que } p' \neq i \end{cases}$$

Resolution of optimal control problems

☞ Dynamic optimisation based on flatness

Multi-robots coordination: centralized approach

Limitation 1

- Prohibitive computation time

Solution 1

Step of simplification of the initial problem:
☞ Motion planning of a virtual robot which is located at the centre of gravity of the formation

Limitation 2

- Problems due to the supervisor

Multi-robots coordination: centralized approach

Limitation 1

- Prohibitive computation time

Solution 1

Step of simplification of the initial problem:
☞ Motion planning of a virtual robot which is located at the centre of gravity of the formation

Limitation 2

- Problems due to the supervisor

Multi-robots coordination: centralized approach

Limitation 1

- Prohibitive computation time

Solution 1

Step of simplification of the initial problem:
☞ Motion planning of a virtual robot which is located at the centre of gravity of the formation

Limitation 2

- Problems due to the supervisor

Multi-robots coordination: decentralized approach

Desired objectives

- low computation time
- high performances
- use of available local information
- no supervisor

Solution

Distributed optimisation based on local information

✎ Each vehicle i only takes into account the intentions of the robots belonging to the conflict set $\mathcal{C}_i(\tau_k)$ (may produce a collision $\mathcal{C}_{i,collision}(\tau_k)$ or may lost the communication $\mathcal{C}_{i,com}(\tau_k)$)

Multi-robots coordination: decentralized approach

Desired objectives

- low computation time
- high performances
- use of available local information
- no supervisor

Solution

Distributed optimisation based on local information

✎ Each vehicle i only takes into account the intentions of the robots belonging to the conflict set $\mathcal{C}_i(\tau_k)$ (may produce a collision $\mathcal{C}_{i,collision}(\tau_k)$ or may lost the communication $\mathcal{C}_{i,com}(\tau_k)$)

Multi-robots coordination: decentralized approach

Desired objectives

- low computation time
- high performances
- use of available local information
- no supervisor

Solution

Distributed optimisation based on local information
✎ Each vehicle i only takes into account the intentions of the robots belonging to the conflict set $\mathcal{C}_i(\tau_k)$ (may produce a collision $\mathcal{C}_{i,collision}(\tau_k)$ or may lost the communication $\mathcal{C}_{i,com}(\tau_k)$)

Multi-robots coordination: decentralized approach

Desired objectives

- low computation time
- high performances
- use of available local information
- no supervisor

Solution

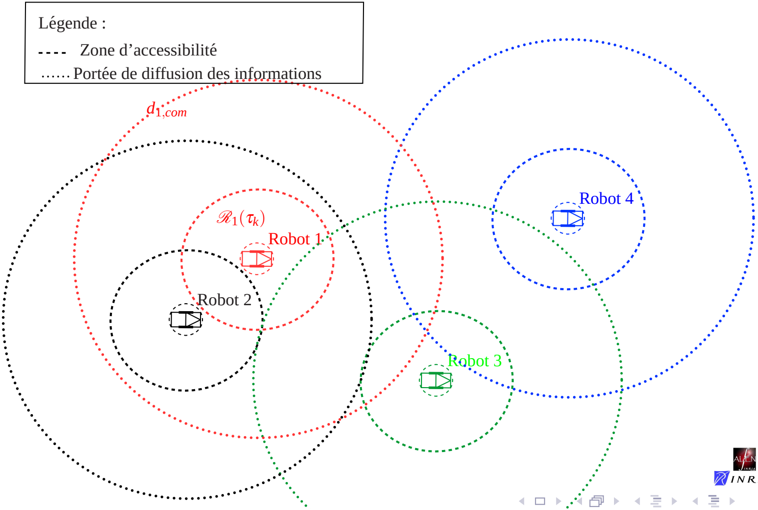
Distributed optimisation based on local information

✎ Each vehicle i only takes into account the intentions of the robots belonging to the conflict set $\mathcal{C}_i(\tau_k)$ (may produce a collision $\mathcal{C}_{i,collision}(\tau_k)$ or may lost the communication $\mathcal{C}_{i,com}(\tau_k)$)

Multi-robots coordination: decentralized approach

Conflicts with robot 1:

$\mathcal{C}_{1,collision}(\tau_k) = \{2\} \qquad \mathcal{C}_{1,com}(\tau_k) = \{4\}$



Multi-robots coordination: decentralized approach

Difficulties

Knowledge of the intentions of robots $p \in \mathcal{C}_i(\tau_k)$

- uniqueness of the presumed trajectory
- coherence between the presumed trajectory and the optimal planned trajectory

Solution

☞ Decomposition of the algorithm into 2 steps:

- ⋆ determination of the presumed trajectory (which only satisfy the individual constraints)
- ⋆ determination of the optimal planned trajectory from the exchanged information between robots belonging to the subset $\mathcal{C}_i(\tau_k)$

Multi-robots coordination: decentralized approach

Difficulties

Knowledge of the intentions of robots $p \in \mathcal{C}_i(\tau_k)$

- uniqueness of the presumed trajectory
- coherence between the presumed trajectory and the optimal planned trajectory

Solution

☞ Decomposition of the algorithm into 2 steps:

- ⋆ determination of the presumed trajectory (which only satisfy the individual constraints)
- ⋆ determination of the optimal planned trajectory from the exchanged information between robots belonging to the subset $\mathcal{C}_i(\tau_k)$

Multi-robots coordination: decentralized approach

Difficulties

Knowledge of the intentions of robots $p \in \mathcal{C}_i(\tau_k)$

- uniqueness of the presumed trajectory
- coherence between the presumed trajectory and the optimal planned trajectory

Solution

☞ Decomposition of the algorithm into 2 steps:

⋆ determination of the presumed trajectory (which only satisfy the individual constraints)

⋆ determination of the optimal planned trajectory from the exchanged information between robots belonging to the subset $\mathcal{C}_i(\tau_k)$

Multi-robots coordination: decentralized approach

Difficulties

Knowledge of the intentions of robots $p \in \mathcal{C}_i(\tau_k)$

- uniqueness of the presumed trajectory
- coherence between the presumed trajectory and the optimal planned trajectory

Solution

☞ Decomposition of the algorithm into 2 steps:

- ⋆ determination of the presumed trajectory (which only satisfy the individual constraints)
- ⋆ determination of the optimal planned trajectory from the exchanged information between robots belonging to the subset $\mathcal{C}_i(\tau_k)$

Multi-robots coordination: decentralized approach

Difficulties

Knowledge of the intentions of robots $p \in \mathcal{C}_i(\tau_k)$

- uniqueness of the presumed trajectory
- coherence between the presumed trajectory and the optimal planned trajectory

Solution

☞ Decomposition of the algorithm into 2 steps:

- ⋆ determination of the presumed trajectory (which only satisfy the individual constraints)
- ⋆ determination of the optimal planned trajectory from the exchanged information between robots belonging to the subset $\mathcal{C}_i(\tau_k)$

Multi-robots coordination: decentralized approach

Difficulties

Knowledge of the intentions of robots $p \in \mathcal{C}_i(\tau_k)$

- uniqueness of the presumed trajectory
- coherence between the presumed trajectory and the optimal planned trajectory

Solution

☞ Decomposition of the algorithm into 2 steps:

- ⋆ determination of the presumed trajectory (which only satisfy the individual constraints)
- ⋆ determination of the optimal planned trajectory from the exchanged information between robots belonging to the subset $\mathcal{C}_i(\tau_k)$

Multi-robots coordination: decentralized approach

Few notations of robot i

- Intuitive horizon $T_d \in \mathbb{R}^+$
- Planning horizon $T_p \in \mathbb{R}^+$ $(T_p \leq T_d)$
- Update horizon $T_c \in \mathbb{R}^+$ $(T_c \leq T_p)$
- $\widehat{q}_i(t, \tau_k)$, $\widehat{u}_i(t, \tau_k)$: presumed trajectory of robot i beginning at τ_k with $t \in [\tau_k, \tau_k + T_d]$ and corresponding control inputs
- $q_{i,ref}(t, \tau_k)$, $u_{i,ref}(t, \tau_k)$: optimal planned trajectory of robot i beginning at τ_k with $t \in [\tau_k, \tau_k + T_p]$ and corresponding control inputs

Multi-robots coordination: decentralized approach

Few notations of robot i

- Intuitive horizon $T_d \in \mathbb{R}^+$
- Planning horizon $T_p \in \mathbb{R}^+$ $(T_p \leq T_d)$
- Update horizon $T_c \in \mathbb{R}^+$ $(T_c \leq T_p)$
- $\widehat{q}_i(t, \tau_k)$, $\widehat{u}_i(t, \tau_k)$: presumed trajectory of robot i beginning at τ_k with $t \in [\tau_k, \tau_k + T_d]$ and corresponding control inputs
- $q_{i,ref}(t, \tau_k)$, $u_{i,ref}(t, \tau_k)$: optimal planned trajectory of robot i beginning at τ_k with $t \in [\tau_k, \tau_k + T_p]$ and corresponding control inputs

Multi-robots coordination: decentralized approach

First sub-problem $\widehat{P}_i(\tau_k)$:

☞ Computation of the presumed trajectory over the intuitive horizon T_d

$$\min_{\widehat{q}_i(t,\tau_k),\widehat{u}_i(t,\tau_k)} \left\|\widehat{q}_i(\tau_k+T_d,\tau_k)-q_{i,final}\right\|^2 + c\int_{\tau_k}^{\tau_k+T_d} L_i(\widehat{q}_i(t,\tau_k),\widehat{u}_i(t,\tau_k),t)dt$$

$$\text{wrt } \forall t\in[\tau_k,\tau_k+T_d]: \begin{cases} \dot{\widehat{q}}_i(t,\tau_k) & = & f_i(\widehat{q}_i(t,\tau_k),\widehat{u}_i(t,\tau_k)) \\ \widehat{q}_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\ \widehat{u}_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\ \widehat{u}_i(t,\tau_k) & \in & \mathcal{U}_i \\ d(\widehat{q}_i(t,\tau_k),O_{m_i}) & \geq & \rho_i+r_{m_i},\ \forall O_{m_i}\in\mathcal{O}_i(\tau_k) \end{cases}$$

Multi-robots coordination: decentralized approach

Second sub-problem $P_i^*(\tau_k)$:

☞ Computation of the optimal trajectory over the planning horizon T_p

$$\min_{q_i(t,\tau_k),u_i(t,\tau_k)} \left\| q_i(\tau_k + T_p, \tau_k) - q_{i,final} \right\|^2 + c \int_{\tau_k}^{\tau_k+T_p} L_i(q_i(t,\tau_k), u_i(t,\tau_k), t) dt$$

$$\text{wrt } \forall t \in [\tau_k, \tau_k + T_p] : \begin{cases} \dot{q}_i(t,\tau_k) & = & f_i(q_i(t,\tau_k), u_i(t,\tau_k)) \\ q_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(t,\tau_k) & \in & \mathcal{U}_i \\ d(q_i(t,\tau_k), O_{m_i}) & \geq & \rho_i + r_{m_i}, \quad \forall O_{m_i} \in \mathcal{O}_i(\tau_k) \\ d(q_i(t,\tau_k), \widehat{q}_p(t,\tau_k)) & \leq & \min(d_{i,com}, d_{p,com}) - \xi, \quad \forall p \in \mathcal{C}_{i,con} \\ d(q_i(t,\tau_k), \widehat{q}_{p'}(t,\tau_k)) & > & \rho_i + \rho_{p'} + \xi, \quad \forall p' \in \mathcal{C}_{i,collision}(\tau_k) \\ d(q_i(t,\tau_k), \widehat{q}_i(t,\tau_k)) & \leq & \xi \end{cases}$$

- coherence between the presumed and the optimal trajectories
- collision avoidance and maintain of the communication links

Multi-robots coordination: decentralized approach

Second sub-problem $P_i^*(\tau_k)$:

☞ Computation of the optimal trajectory over the planning horizon T_p

$$\min_{q_i(t,\tau_k),u_i(t,\tau_k)} \left\| q_i(\tau_k + T_p, \tau_k) - q_{i,final} \right\|^2 + c \int_{\tau_k}^{\tau_k+T_p} L_i(q_i(t,\tau_k), u_i(t,\tau_k), t)dt$$

$$\text{wrt } \forall t \in [\tau_k, \tau_k + T_p] : \begin{cases} \dot{q}_i(t,\tau_k) & = & f_i(q_i(t,\tau_k), u_i(t,\tau_k)) \\ q_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(t,\tau_k) & \in & \mathcal{U}_i \\ d(q_i(t,\tau_k), O_{m_i}) & \geq & \rho_i + r_{m_i}, \quad \forall O_{m_i} \in \mathcal{O}_i(\tau_k) \\ d(q_i(t,\tau_k), \widehat{q}_p(t,\tau_k)) & \leq & \min(d_{i,com}, d_{p,com}) - \xi, \quad \forall p \in \mathcal{C}_{i,com} \\ d(q_i(t,\tau_k), \widehat{q}_{p'}(t,\tau_k)) & > & \rho_i + \rho_{p'} + \xi, \quad \forall p' \in \mathcal{C}_{i,collision}(\tau_k) \\ d(q_i(t,\tau_k), \widehat{q}_i(t,\tau_k)) & \leq & \xi \end{cases}$$

- coherence between the presumed and the optimal trajectories
- collision avoidance and maintain of the communication links

Multi-robots coordination: decentralized approach

Second sub-problem $P_i^*(\tau_k)$:

☞ Computation of the optimal trajectory over the planning horizon T_p

$$\min_{q_i(t,\tau_k),u_i(t,\tau_k)} \left\| q_i(\tau_k + T_p, \tau_k) - q_{i,final} \right\|^2 + c \int_{\tau_k}^{\tau_k+T_p} L_i(q_i(t,\tau_k), u_i(t,\tau_k), t)dt$$

$$\text{wrt } \forall t \in [\tau_k, \tau_k + T_p] : \begin{cases} \dot{q}_i(t,\tau_k) & = & f_i(q_i(t,\tau_k), u_i(t,\tau_k)) \\ q_i(\tau_k,\tau_k) & = & q_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(\tau_k,\tau_k) & = & u_{i,ref}(\tau_k,\tau_{k-1}) \\ u_i(t,\tau_k) & \in & \mathcal{U}_i \\ d(q_i(t,\tau_k), O_{m_i}) & \geq & \rho_i + r_{m_i}, \quad \forall O_{m_i} \in \mathcal{O}_i(\tau_k) \\ d(q_i(t,\tau_k), \widehat{q}_p(t,\tau_k)) & \leq & \min(d_{i,com}, d_{p,com}) - \xi, \quad \forall p \in \mathcal{C}_{i,com} \\ d(q_i(t,\tau_k), \widehat{q}_{p'}(t,\tau_k)) & > & \rho_i + \rho_{p'} + \xi, \quad \forall p' \in \mathcal{C}_{i,collision}(\tau_k) \\ d(q_i(t,\tau_k), \widehat{q}_i(t,\tau_k)) & \leq & \xi \end{cases}$$

- coherence between the presumed and the optimal trajectories
- collision avoidance and maintain of the communication links

Multi-robots coordination: comparative results (1)

Algorithms

- Centralized approach
- "Leader/Follower" [Kuwata et al., J. of Field Robotics, 2006]
- Weakly decentralized approach [Keviczky et al., Automatica, 2006]
- Strongly decentralized approach

Number of robots N_a	2
Maximum linear velocity $v_{i,max}$	0.5m/s
Maximum angular velocity $v_{i,max}$	5rad/s
Radius of robot ρ_i	0.2m
Planning horizon T_p	2s
Update horizon T_c	0.5s
Intuitive horizon T_d	2s
Maximum deformation ξ	0.25

Multi-robots coordination: comparative results (1)

Algorithms

- Centralized approach
- "Leader/Follower" [Kuwata et al., J. of Field Robotics, 2006]
- Weakly decentralized approach [Keviczky et al., Automatica, 2006]
- Strongly decentralized approach

Number of robots N_a	2
Maximum linear velocity $v_{i,max}$	0.5m/s
Maximum angular velocity $v_{i,max}$	5rad/s
Radius of robot ρ_i	0.2m
Planning horizon T_p	2s
Update horizon T_c	0.5s
Intuitive horizon T_d	2s
Maximum deformation ξ	0.25

Multi-robots coordination: comparative results (1)

Algorithms

- Centralized approach
- "Leader/Follower" [Kuwata et al., J. of Field Robotics, 2006]
- Weakly decentralized approach [Keviczky et al., Automatica, 2006]
- Strongly decentralized approach

Number of robots N_a	2
Maximum linear velocity $v_{i,max}$	0.5m/s
Maximum angular velocity $v_{i,max}$	5rad/s
Radius of robot ρ_i	0.2m
Planning horizon T_p	2s
Update horizon T_c	0.5s
Intuitive horizon T_d	2s
Maximum deformation ξ	0.25

Multi-robots coordination: comparative results (1)

Algorithms

- Centralized approach
- "Leader/Follower" [Kuwata et al., J. of Field Robotics, 2006]
- Weakly decentralized approach [Keviczky et al., Automatica, 2006]
- Strongly decentralized approach

Number of robots N_a	2
Maximum linear velocity $v_{i,max}$	0.5m/s
Maximum angular velocity $v_{i,max}$	5rad/s
Radius of robot ρ_i	0.2m
Planning horizon T_p	2s
Update horizon T_c	0.5s
Intuitive horizon T_d	2s
Maximum deformation ξ	0.25

Multi-robots coordination: comparative results (1)

Algorithms

- Centralized approach
- "Leader/Follower" [Kuwata et al., J. of Field Robotics, 2006]
- Weakly decentralized approach [Keviczky et al., Automatica, 2006]
- Strongly decentralized approach

Number of robots N_a	2
Maximum linear velocity $v_{i,max}$	0.5m/s
Maximum angular velocity $v_{i,max}$	5rad/s
Radius of robot ρ_i	0.2m
Planning horizon T_p	2s
Update horizon T_c	0.5s
Intuitive horizon T_d	2s
Maximum deformation ξ	0.25

Multi-robots coordination: comparative results (1)

Cent. and weakly decent.

Video

"Leader / follower"

Video

Multi-robots coordination: comparative results (1)

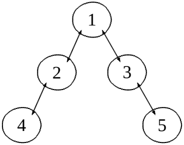
Strongly decentralized

Video

Multi-robots coordination: comparative results (1)

Approach	Cent.	Leader/Follower	Weakly decent.	Strongly decent.
Maxi. time of conflict resolution	$172ms$	$40ms$	$172ms$	$94ms$
Time reaching goal	$16.2s$	$16.5s$	$16.2s$	$16.4s$

Multi-robots coordination: comparative results (2)



Number of robots N_a	5
Maximum linear velocity $v_{i,max}$	0.5m/s
Maximum angular velocity $v_{i,max}$	5rad/s
Radius of robot ρ_i	0.2m
Broadcasting range $d_{i,com}$	2.5m
Planning horizon T_p	2s
Update horizon T_c	0.5s
Intuitive horizon T_d	2.5s
Maximum deformation ξ	0.25

Multi-robots coordination: comparative results (2)

Strongly decentralized

Video

Multi-robots coordination: comparative results (2)

Approach	Cent.	Leader/Follower	Weakly decent.	Strongly decent.
Maxi time of conflict resolution	$2050ms$	$313ms$	$703ms$	$121ms$
Exchanged Info.	global	local	local	local
Implem.	$--$ if $N_a \gg 1$	$++$ sequential resolution	$-$ if conflict with a lot of robots	$+$
Time reaching goal	$35s$	$39s$	$36s$	$36.5s$

Multi-robots coordination: decentralized approach

Difficulties

- shape of the obstacles (disk),
- blocking situations due to local minima.

Solutions

- ⋆ approximation of complex obstacle shape by polyhedrons (introduction of differentiable constraints),
- ⋆ decision graph.

Multi-robots coordination: decentralized approach

Difficulties

- shape of the obstacles (disk),
- blocking situations due to local minima.

Solutions

⋆ approximation of complex obstacle shape by polyhedrons (introduction of differentiable constraints),

⋆ decision graph.

Multi-robots coordination: decentralized approach

Difficulties

- shape of the obstacles (disk),
- blocking situations due to local minima.

Solutions

- ⋆ approximation of complex obstacle shape by polyhedrons (introduction of differentiable constraints),
- ⋆ decision graph.

Multi-robots coordination: decentralized approach

Difficulties

- shape of the obstacles (disk),
- blocking situations due to local minima.

Solutions

⋆ approximation of complex obstacle shape by polyhedrons (introduction of differentiable constraints),

⋆ decision graph.

Multi-robots coordination: decentralized approach

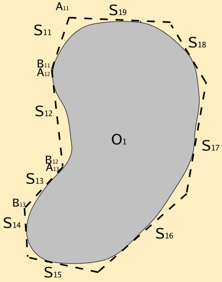
Difficulties

- shape of the obstacles (disk),
- blocking situations due to local minima.

Solutions

⋆ approximation of complex obstacle shape by polyhedrons (introduction of differentiable constraints),

⋆ decision graph.



Multi-robots coordination: decentralized approach

Difficulties

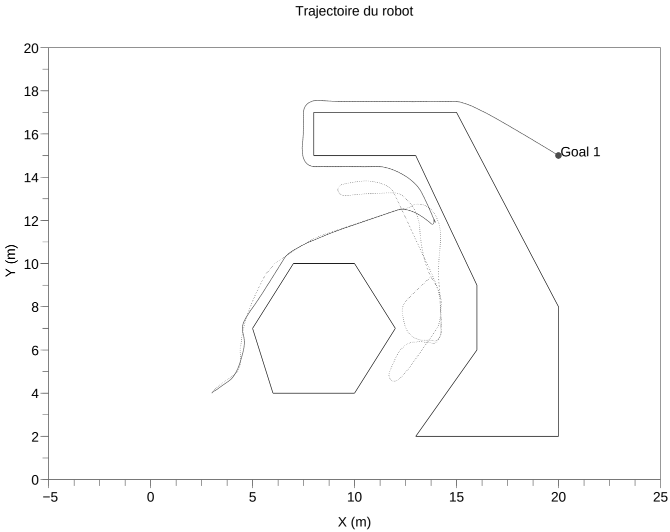
- shape of the obstacles (disk),
- blocking situations due to local minima.

Solutions

⋆ approximation of complex obstacle shape by polyhedrons (introduction of differentiable constraints),

⋆ decision graph.

Multi-robots coordination: decentralized approach



Multi-robots coordination: decentralized approach

Video

Multi-robots coordination: decentralized approach

Video

Multi-robots coordination: decentralized approach

Video

Bibliography

A. Astolfi, “Discontinuous control of nonholonomic systems”, *Systems and Control Letters*, 27(1), pp. 37–45, 1996.

M. Bennani and P. Rouchon (1995),“Robust stabilization of flat and chained systems”, *in Proceedings of the ECC,* Rome, Italie.

C.D. Boor, “A practical guide to splines ”, Springer, 1978.

R. Brockett, “Asymptotic stability and feedback stabilization”, in R.W. Brockett, R.S. Millman, and H.J. Sussmann (eds.), Differential geometric control theory (Boston, MA: Birkhauser), pp. 181–195, 1983.

Bibliography

- C. Canudas de Wit, B. Siciliano and G. Bastin (Eds), "Theory of Robot Control", Communications and control engineering series, Springer-verlag, Berlin Heidelberg New-York, 1996.
- M. Defoort, T. Floquet, W. Perruquetti, A. Kökösy, "Tracking of a unicycle-type mobile robot using integral sliding mode", *ICINCO*, Barcelone, Espagne, 2005.
- M. Defoort, T. Floquet, A. Kökösy, W. Perruquetti, "Commande coopérative d'une formation de robots mobiles", in *proc. IEEE CIFA 2006*, Bordeaux, France, 30 mai – 1er juin, 2006.

Bibliography

- M. Defoort, T. Floquet, A. Kökösy, W. Perruquetti, "Integral sliding mode control for trajectory tracking of a unicycle type mobile robot", textitInt. Journal of Integrated Computer Aided Engeneering, to appear, 2006.
- J. Desai, J. Ostrowski and V. Kumar, "Modeling and control of formation of nonholonomic mobile robots", *IEEE Robotics and Automation*, 17(6) pp. 905–308, 2001.
- S.V. Drakunov, T. Floquet and W. Perruquetti, "Stabilization and tracking control for an extended Heisenberg system with a drift", *Systems ans control Letters* **54** (2005), pp. 435–445.
- M. Fliess, J.Lévine, Ph.Martin, et P.Rouchon. "Sur les systèmes non linéaires différentiellement plats", *C.R. Acad. Sci. Paris*, I–315, pp. 619–624, 1992.

Bibliography

- M. Fliess, J. Lévine, Ph. Martin et P. Rouchon, "Linéarisation par bouclage dynamique et transformations de Lie-Bäcklund.", *C.R. Acad. Sci. Paris*, I 317, pp. 981–986, 1993.
- M. Fliess, J. Lévine, Ph. Martin and P. Rouchon, "Flatness and defect of nonlinear systems : introductory theory and examples", *Int. J. Control*, Vol. 61 (6), pp. 1327–1361, 1995.
- M. Fliess, J. Lévine, Ph. Martin and P. Rouchon, "Design of trajectory stabilizing feedback for driftless flat systems", *in proc. ECC 95*, Rome, Italie, pp. 1882–1887, 1995.

Bibliography

- M. Fliess, J.Lévine, Ph.Martin, et P.Rouchon. "A lie- bäcklund approach to equivalence and flatness of nonlinear systems", *IEEE Trans Automatic Control*, 44, pp. 922–937, 1999.
- Michel Fliess et Richard Marquez. "Continuous-time linear predictive control and flatness: a module-theoretic setting with examples", *International Journal of Control*, 73(7):606, 2000.
- T. Floquet, J.-P. Barbot and W. Perruquetti, "Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems", *Automatica*, 39(6), pp. 1077–1083, 2003.

Bibliography

- J.P. Hespanha, D. Liberzon and A.S. Morse, "Logic-based switching control of a nonholonomic system with parametric modeling uncertainty ", *Systems and Control Letters*, 38, pp. 167–177, 1999.
- ZP. Jiang and H. Nijmeijer, "Tracking control of mobile robots: a case study in backsteeping", *Automatica*, **33**, No 7, pp. 1393-1399, 1997.
- J.P. Laumond, "La robotique mobile", *Hermès*, Traité IC2 INformation – Commande – Communication.
- P. Martin, P. Rouchon, "Feedback linearization and driftless systems", *Mathematics of Control, Signals and Systems*, Vol. 7, pp. 235–254, 1994.
- P. Martin, P. Rouchon, "Any controllable driftless systems with 3 inputs and 5 states is flat", *Systems & Control Letter, Vol. 25*, pp.

Bibliographie

- P. Martin, P. Rouchon, "Any controllable driftless systems with m inputs and $m+2$ states is flat", *Proc. IEEE CDC, New Orleans*, LA, pp. 167–175, 1995.
- M. B. Milam, N. Petit, and R. M. Murray, "Constrained trajectory generation for microsatellite formation flying", In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pp. 328–333, 2001.
- R. Murray and S. Sastry, "Nonholonomic Motion Planning: Steering Using Sinusoids", *IEEE Trans. on Automatic Control*, 38(5), pp. 700–716, 1993.

Bibliography

- J. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift", *Systems and Control Letters*, 18(2), pp. 147–158, 1992.
- C. Samson, "Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots", *IEEE Trans. on Automatic Control*, 40, pp. 64–77, 1995.
- Herbert G. Tanner and Amit Kumar,"Towards Decentralization of Mutli-robot Navigation Functions", *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 4143–4148, 2005.
- Herbert G. Tanner and Amit Kumar, "Formation Stabilization of Multiple Agents Using Decentralized Navigation Functions", Robotics: Science and Systems I, S. Thrun, G. Sukhatme, S. Schaal and O. Brock (eds), MIT Press, pp 49–56, 2005.

Bibliography

Z.-P. Jiang, E. Lefeber and H. Nijmeijer, "Saturated stabilization and track control of a nonholonomic mobile robot", *Systems and Control Letters*, 42, pp. 327–332, 2001.

Z.P. Jiang, "Robust exponential regulation of nonholonomic systems with uncertainties", *Automatica* 36, pp. 189–209, 2000.

Z.P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form", *IEEE Transactions on Automatic Control*, Vol. 44, No 2, pp. 265–279, 1999.

Bibliography

A. Astolfi, "Discontinuous control of nonholonomic systems", *Systems and Control Letters*, 27(1), pp. 37–45, 1996.

M. Bennani and P. Rouchon (1995), "Robust stabilization of flat and chained systems", *in Proceedings of the ECC*, Rome, Italie.

C.D. Boor, "A practical guide to splines ", Springer, 1978.

R. Brockett, "Asymptotic stability and feedback stabilization", in R.W. Brockett, R.S. Millman, and H.J. Sussmann (eds.), Differential geometric control theory (Boston, MA: Birkhauser), pp. 181–195, 1983.

Bibliography

- C. Canudas de Wit, B. Siciliano and G. Bastin (Eds), "Theory of Robot Control", Communications and control engineering series, Springer-verlag, Berlin Heidelberg New-York, 1996.
- M. Defoort, T. Floquet, W. Perruquetti, A. Kökösy, "Tracking of a unicycle-type mobile robot using integral sliding mode", *ICINCO*, Barcelone, Espagne, 2005.
- M. Defoort, T. Floquet, A. Kökösy, W. Perruquetti, "Commande coopérative d'une formation de robots mobiles", in *proc. IEEE CIFA 2006*, Bordeaux, France, 30 mai – 1er juin, 2006.

Bibliography

- M. Defoort, T. Floquet, A. Kökösy, W. Perruquetti, "Integral sliding mode control for trajectory tracking of a unicycle type mobile robot", *Int. Journal of Integrated Computer Aided Engeneering*, 2006.
- M. Defoort, Palos J., Kökösy A., Floquet T., W. Perruquetti, "Performance-based reactive navigation for nonholonomic mobile robots ", *Robotica*, 2009.
- M. Defoort, T. Floquet, A. Kökösy, W. Perruquetti, "Sliding mode formation control for cooperative autonomous mobile robots ", *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 11, pp. 3944–3953, Nov. 2008.
- J. Desai, J. Ostrowski and V. Kumar, "Modeling and control of formation of nonholonomic mobile robots", *IEEE Robotics and Automation*, 17(6) pp. 905–308, 2001.

Bibliography

S.V. Drakunov, T. Floquet and W. Perruquetti, "Stabilization and tracking control for an extended Heisenberg system with a drift", *Systems ans control Letters* **54** (2005), pp. 435–445.

M. Fliess, J.Lévine, Ph.Martin, et P.Rouchon. "Sur les systèmes non linéaires différentiellement plats", *C.R. Acad. Sci. Paris*, I–315, pp. 619–624, 1992.

Bibliography

- M. Fliess, J. Lévine, Ph. Martin et P. Rouchon, “Linéarisation par bouclage dynamique et transformations de Lie-Bäcklund.”, *C.R. Acad. Sci. Paris*, I 317, pp. 981–986, 1993.
- M. Fliess, J. Lévine, Ph. Martin and P. Rouchon, “Flatness and defect of nonlinear systems : introductory theory and examples”, *Int. J. Control*, Vol. 61 (6), pp. 1327–1361, 1995.
- M. Fliess, J. Lévine, Ph. Martin and P. Rouchon, “Design of trajectory stabilizing feedback for driftless flat systems”, *in proc. ECC 95*, Rome, Italie, pp. 1882–1887, 1995.

Bibliography

- M. Fliess, J.Lévine, Ph.Martin, et P.Rouchon. "A lie- bäcklund approach to equivalence and flatness of nonlinear systems", *IEEE Trans Automatic Control*, 44, pp. 922–937, 1999.
- Michel Fliess et Richard Marquez. "Continuous-time linear predictive control and flatness: a module-theoretic setting with examples", *International Journal of Control*, 73(7):606, 2000.
- T. Floquet, J.-P. Barbot and W. Perruquetti, "Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems", *Automatica*, 39(6), pp. 1077–1083, 2003.

Bibliography

- J.P. Hespanha, D. Liberzon and A.S. Morse, "Logic-based switching control of a nonholonomic system with parametric modeling uncertainty ", *Systems and Control Letters*, 38, pp. 167–177, 1999.
- ZP. Jiang and H. Nijmeijer, "Tracking control of mobile robots: a case study in backsteeping", *Automatica*, **33**, No 7, pp. 1393-1399, 1997.
- J.P. Laumond, "La robotique mobile", *Hermès*, Traité IC2 INformation – Commande – Communication.
- P. Martin, P. Rouchon, "Feedback linearization and driftless systems", *Mathematics of Control, Signals and Systems*, Vol. 7, pp. 235–254, 1994.
- P. Martin, P. Rouchon, "Any controllable driftless systems with 3 inputs and 5 states is flat", *Systems & Control Letter, Vol. 25*, pp.

Bibliographie

P. Martin, P. Rouchon, “Any controllable driftless systems with m inputs and $m+2$ states is flat”, *Proc. IEEE CDC, New Orleans*, LA, pp. 167–175, 1995.

M. B. Milam, N. Petit, and R. M. Murray, “Constrained trajectory generation for microsatellite formation flying”, In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pp. 328–333, 2001.

R. Murray and S. Sastry, “Nonholonomic Motion Planning: Steering Using Sinusoids”, *IEEE Trans. on Automatic Control*, 38(5), pp. 700–716, 1993.

Bibliography

- J. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift", *Systems and Control Letters*, 18(2), pp. 147–158, 1992.
- C. Samson, "Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots", *IEEE Trans. on Automatic Control*, 40, pp. 64–77, 1995.
- Herbert G. Tanner and Amit Kumar,"Towards Decentralization of Mutli-robot Navigation Functions", *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 4143–4148, 2005.
- Herbert G. Tanner and Amit Kumar, "Formation Stabilization of Multiple Agents Using Decentralized Navigation Functions", Robotics: Science and Systems I, S. Thrun, G. Sukhatme, S. Schaal and O. Brock (eds), MIT Press, pp 49–56, 2005.

Bibliography

- Z.-P. Jiang, E. Lefeber and H. Nijmeijer, "Saturated stabilization and track control of a nonholonomic mobile robot", *Systems and Control Letters*, 42, pp. 327–332, 2001.
- Z.P. Jiang, "Robust exponential regulation of nonholonomic systems with uncertainties", *Automatica* 36, pp. 189–209, 2000.
- Z.P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form", *IEEE Transactions on Automatic Control*, Vol. 44, No 2, pp. 265–279, 1999.