Cooperative mobile robots.

W. PERRUQUETTI
ALIEN INRIA - LNE,
LAGIS UMR CNRS 8146,
Ecole Centrale de Lille, 59650 Villeneuve D'Ascq.
wilfrid.perruquetti@ec-lille.fr.
Special thanks to :
Michael Defoort, Thierry Floquet and Anne Marie Kökösy.

26 Novembre 2009

OUTLINE

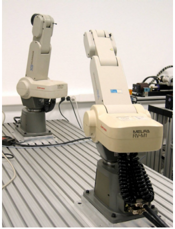
Robotics is a cross fertilizing area which aims at designing and using concrete physical devices with the following capabilities :

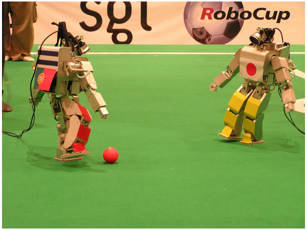
- action, (actuators)
- perception, (sensors)
- decision,
- interaction with the environment,

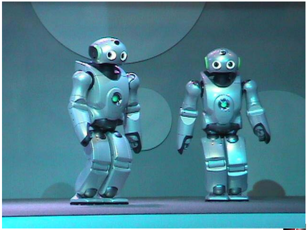
in order to fulfill a task with or without a human.
(The case $<<$ not $>>$: human-robot interactions)

Classification des robots :

- Robots mobiles : à roues, à chenilles, à pattes, selon le type de locomotion, aérien, sous-marin, terrestre ou spatial, avec une attention particulière pour la robotique humanoïde
- Robots fixes : manipulateurs, interface haptique, etc ...

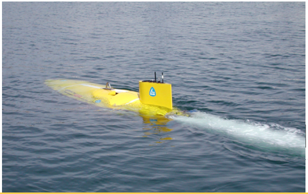

MELFA
RV-M1

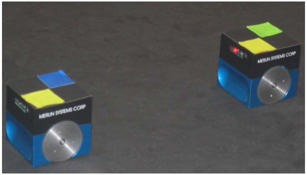

sgi
RoboCup




(COURTESY SONY)


EMERGENCY STOP
TRC




MERLIN SYSTEMS CORP
MERLIN SYSTEMS CORP

La robotique concerne l'intégration de quatre composantes :

- Conception, modélisation, analyse : nouveaux besoins
- Chaine de perception : capteurs qui << nourrissent >> les robots d'informations de nature diverse (signaux analogiques, numériques (par exemple image), etc . . .).
- Chaine d'action et de décision qui comporte plusieurs chainons : cognitif, la planification de tâches, la planification de mouvements, la commande et se terminant par les actionneurs.
- L'interaction avec l'environnement : collaboration robots/robots et/ou hommes et/ou monde physique.

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

Conséquences :

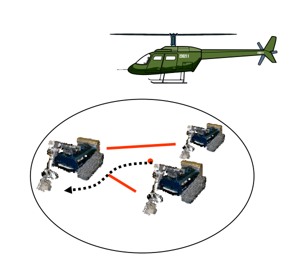
- téléopération de robots, (nouveaux enjeux).
- robots en réseaux : ce sont des dispositifs robotisés (manipulateurs, véhicules mobiles, robots humanoïdes, etc ...) qui sont connectés *via* un réseau de communication tel qu'un réseau local (LAN) ou le réseau internet (WAN) → faire coopérer un ensemble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

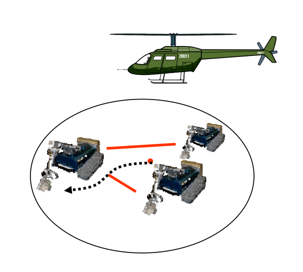


- to get complex behaviors by using simple agent based behaviors

☞ 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,
- . . .

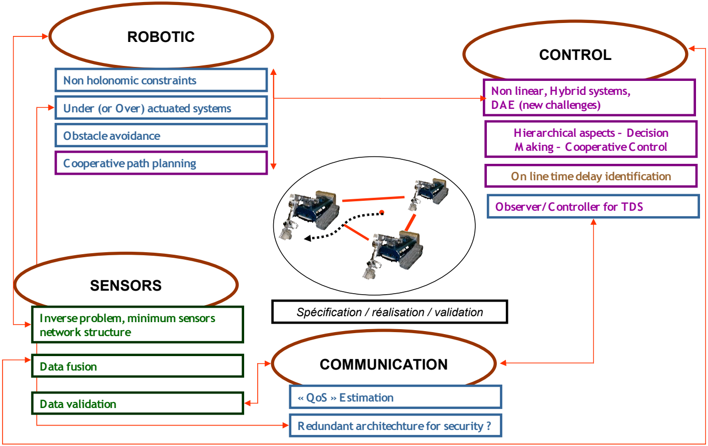
Framework : multidisciplinary research

- 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

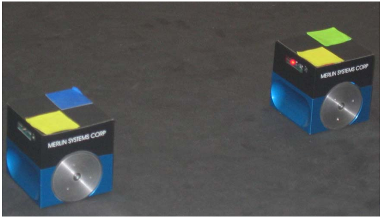
- modeling, path planning and control (constraints, nonlinear models, time delays, hierarchical aspects, hybrid system aspect, quantization . . .)
- graph theory,
- communication protocols,
- logical decision making, scheduling,
- . . .

Snap shot of Robocoop project / Big picture



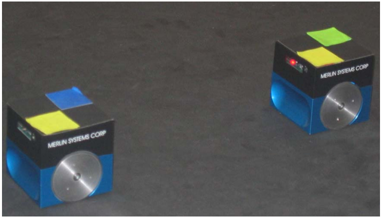
Our Goals

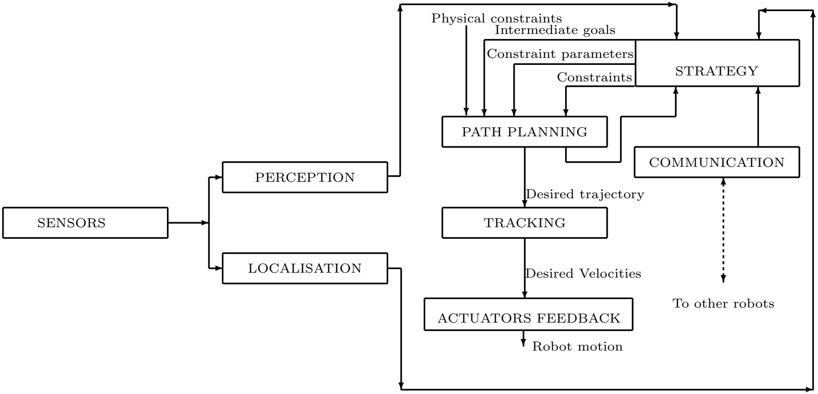
- ✔ path planning and path tracking
- ✔ test on benchmarks



Our Goals

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




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

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 ?

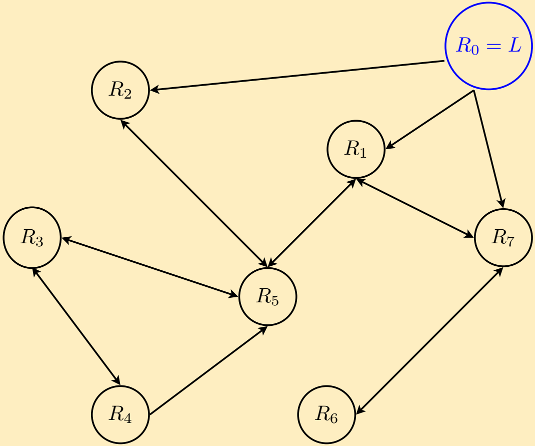
Within a group of mobile robots, some of them may play a particular role : leaders. Distinguish between fleets :

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

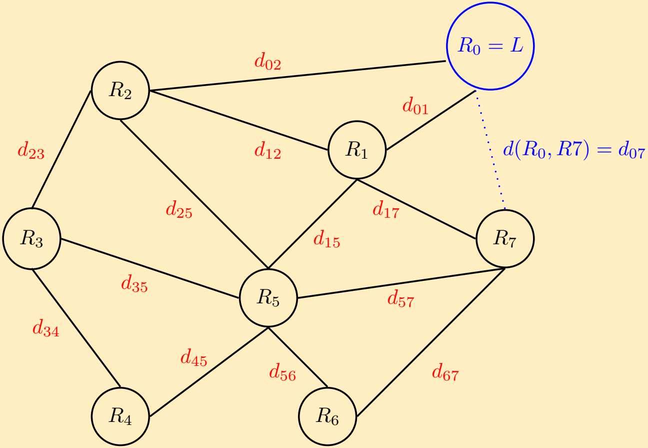
Questions

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

Communications graph $\mathcal{G}^c$



Formation graph $\mathcal{G}^f$



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"

On the way to integration

Philosophy integrated layer :
"Strategic layer" (goal planning) + "Tactical layer" (guidance, navigation) + "Reflexive layer" (obstacle avoidance)

Solution

☞ Generate and execute a (sub)-optimal path planning which satisfy :

- geometric formation and communications constraints,
- obstacle avoidance constraints,
- given boundary conditions,
- other constraints : time constraints (rescue missions), energy constraints (batteries duration, . . .)

On the way to integration

Philosophy integrated layer :
“Strategic layer” (goal planning) + “Tactical layer” (guidance, navigation) + “Reflexive layer” (obstacle avoidance)

Solution

☞ Generate and execute a (sub)-optimal path planning which satisfy :

- geometric formation and communications constraints,
- obstacle avoidance constraints,
- given boundary conditions,
- other constraints : time constraints (rescue missions), energy constraints (batteries duration, . . .)

On the way to integration

Philosophy integrated layer :
"Strategic layer" (goal planning) + "Tactical layer" (guidance, navigation) + "Reflexive layer" (obstacle avoidance)

Solution

☞ Generate and execute a (sub)-optimal path planning which satisfy :

- geometric formation and communications constraints,
- obstacle avoidance constraints,
- given boundary conditions,
- other constraints : time constraints (rescue missions), energy constraints (batteries duration, . . .)

On the way to integration

Philosophy integrated layer :
"Strategic layer" (goal planning) + "Tactical layer" (guidance, navigation) + "Reflexive layer" (obstacle avoidance)

Solution

☞ Generate and execute a (sub)-optimal path planning which satisfy :

- geometric formation and communications constraints,
- obstacle avoidance constraints,
- given boundary conditions,
- other constraints : time constraints (rescue missions), energy constraints (batteries duration, . . .)

On the way to integration

Philosophy integrated layer :
“Strategic layer” (goal planning) + “Tactical layer” (guidance, navigation) + “Reflexive layer” (obstacle avoidance)

Solution

☞ Generate and execute a (sub)-optimal path planning which satisfy :

- geometric formation and communications constraints,
- obstacle avoidance constraints,
- given boundary conditions,
- other constraints : time constraints (rescue missions), energy constraints (batteries duration, . . .)

General constraints

We would like to include in the path planning the following constraints :

1. some constraints due to physics (energy limitation, maximal velocity and acceleration of the robots)
2. obstacle avoidance,
3. collision avoidance with the robots and other mobile objects,
4. distances between robots (communications),
5. geometry of the formation
6. ...

Somme settings

Group of N_i [a] mobile robots related to the i^{th} mobile robot evolving in a partially known space with N_o obstacles.

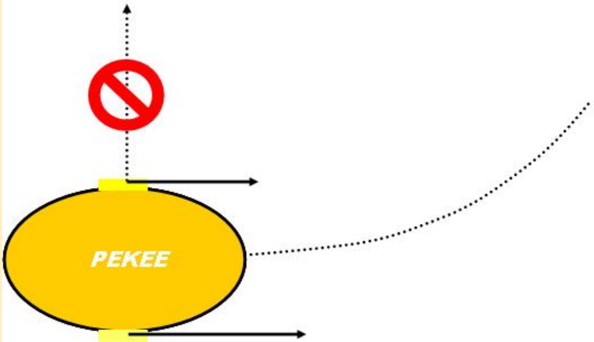
- each obstacle O_m $(m \in \{1, \ldots, N_o\})$ is covered by a disc centered at (x_m^o, y_m^o) with radius r_m^o (if complex geometry use a covering of discs).
- I_N the set $\{1, \ldots, N\}$,
- the $n^{\text{th}} \in I_N$ mobile robot denoted by A_n and located at (x_n, y_n) occupies a space modeled by a disc of radius r_n centered at (x_n, y_n).
- robot A_n : X_n and U_n denotes respectively the state variables and the control variables.

---

a. Index i (dropped sometimes) will refer to some properties or known objects linked to the i^{th} mobile robot. $N = N_i$!

Robot de type unicycle

Non Holonomy (Admissible path)



Kinematic constraint : a robot should not be reduced to a single point (x, y).

Types of models : see [5] “Theory of Robot Control”, C. Canudas de Wit, B. Siciliano and G. Bastin (Eds).

1. Kinematic model (take into account non holonomic constraints)

   Posture $=(x, y, \theta)$ in most of the case.

2. Dynamical model (KM + dynamics induced by actuators (most of the time electrical motors))

Kinematic model is enough

Kinematic model which includes *non holonomic constraints* this is non integrable constraints of the form

$$\boxed{\dot{q} = B(q)u,} \tag{1}$$

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])

$\dot{x} = f(x,u), x \in \mathbb{R}^n, u \in \mathbb{R}^m$ flat $\Leftrightarrow$ il existe m fonctions qui paramétrisent tout !

Donc fixer ces fonctions c'est fixer le comportement dynamique du système !

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 [4]) 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))

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= w \end{aligned} \tag{2}$$

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

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= v\frac{\tan(\phi)}{l} \\ \dot{\phi} &= w \end{aligned} \tag{3}$$

Kinematic model : flatness is the key point

Flat Outputs : (x, y).

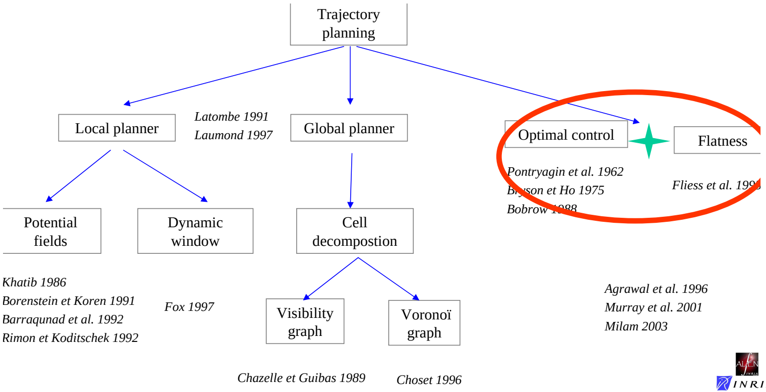
Indeed :

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

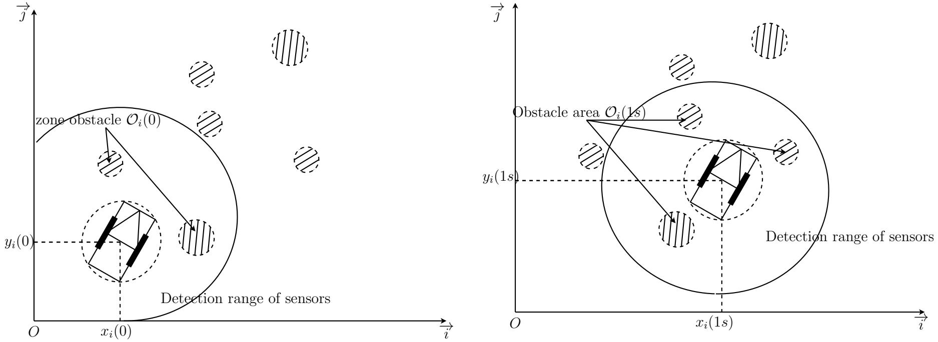
Path planning for a single robot

Motion planning

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



Motion planning for a single robot : problem setup



Single robot : off-line algorithm

Dynamic optimisation based on flatness

Criteria :

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

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

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

Single robot : off-line algorithm

Dynamic optimisation based on flatness

Criteria :

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

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

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

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

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

Single robot : off-line algorithm

Dynamic optimisation based on flatness

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

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

Single robot : off-line algorithm

Dynamic optimisation based on flatness

Criteria :

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

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

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

Single robot : off-line algorithm

Dynamic optimisation based on flatness

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

- $\dot{q}_i(t) = f_i(q_i(t), u_i(t))$
- $\begin{cases} q_i(t_{initial}) = q_{i,initial} \\ q_i(t_{final}) = q_{i,final} \\ u_i(t_{initial}) = u_{i,initial} \\ u_i(t_{final}) = u_{i,final} \end{cases}$ $\rightleftarrows$ $q_i = \varphi_1(z_i, \dot{z}_i, \ddot{z}_i)$ $u_i = \varphi_2(z_i, \dot{z}_i, \ddot{z}_i)$
- $u_i(t) \in \mathcal{U}_i$
- $\forall O_{m_i} \in \mathcal{O}_i(t_{initial})$ $d(q_i(t), O_{m_i}) \geq \rho_i + r_{m_i}$

Single robot : off-line algorithm

Dynamic optimisation 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.

Single robot : off-line algorithm

Dynamic optimisation 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)

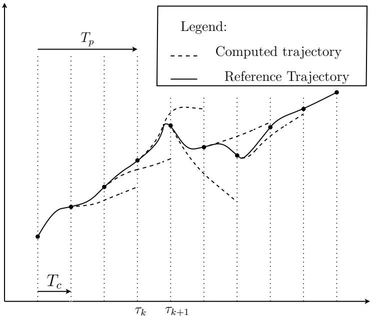
☞ Computation of the corresponding control inputs using the flatness properties of the system

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 motion planner

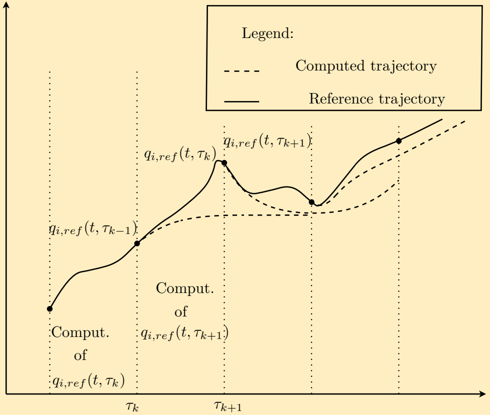
- $T_p (> 0)$: planning horizon
- $T_c (> 0)$: update period
- $\tau_k (k \in \mathbb{N}, \tau_k = t_{initial} + kT_c)$: updates



Single robot : on-line algorithm

Implementation

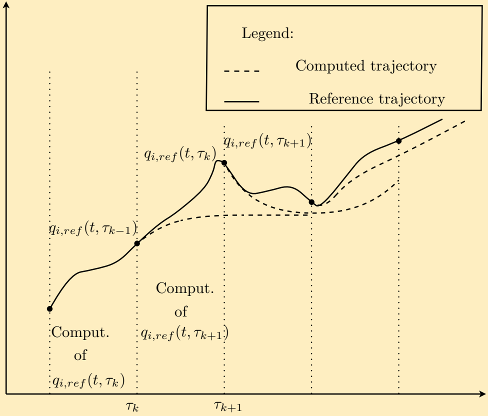
- ☞ initialisation step : computations before the movement of the robot
- ☞ step of iterative computations : computations over any interval $[\tau_{k-1}, \tau_k)$



Single robot : on-line algorithm

Implementation

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



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 :

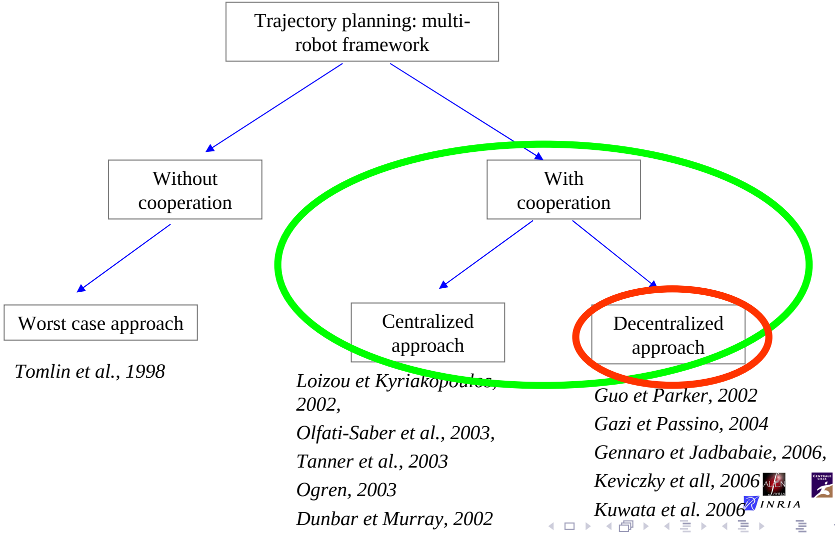
- terminal constraints
- physical constraints (nonholonomic, maximum velocities, . . .)
- obstacle avoidance
- minimum distances between robots (collision avoidance)
- maximum distances between robots (respect of the broadcasting range)

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

Limitation to direct extension via a supervisor

1. Prohibitive computation time (centralized approche))
2. Problems due to the supervisor (if destroyed ...)

Solution to 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

Multi-robots coordination : centralized approach

Limitation to direct extension via a supervisor

1. Prohibitive computation time (centralized approche))
2. Problems due to the supervisor (if destroyed ...)

Solution to 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

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

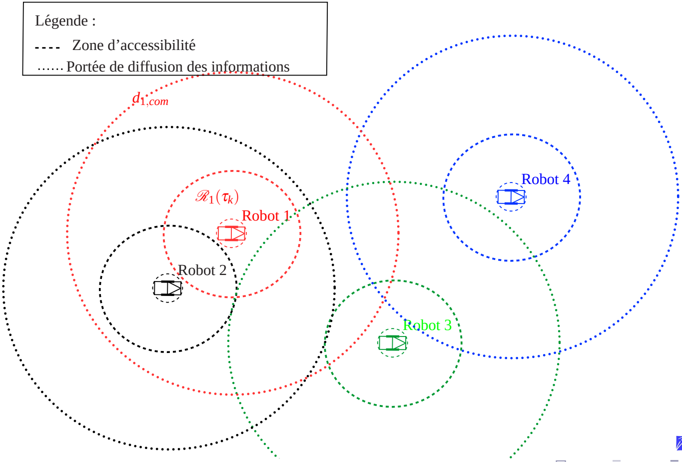
Solution

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

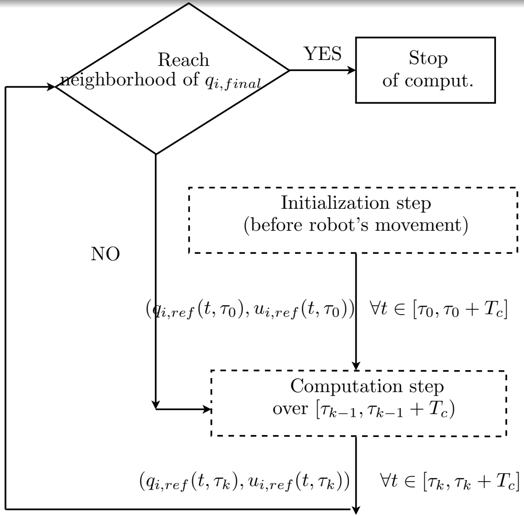
- Intuitive horizon $T_d \in \mathbb{R}^+$
- Planning horizon $T_p \in \mathbb{R}^+$ $(T_p \leq T_d)$
- Update horizon $T_c \in \mathbb{R}^+$ $(T_c \leq T_p)$
- $\widehat{q}_i(t, \tau_k)$, $\widehat{u}_i(t, \tau_k)$: presumed trajectory of robot i beginning at τ_k with $t \in [\tau_k, \tau_k + T_d]$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

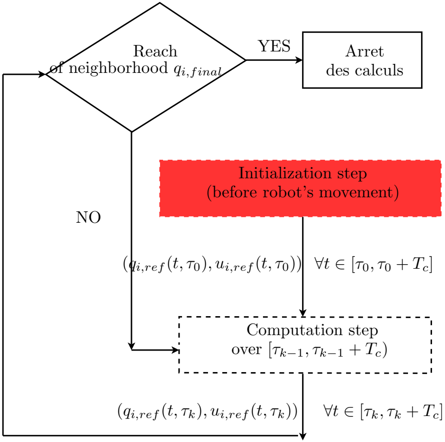
Few notations of robot i

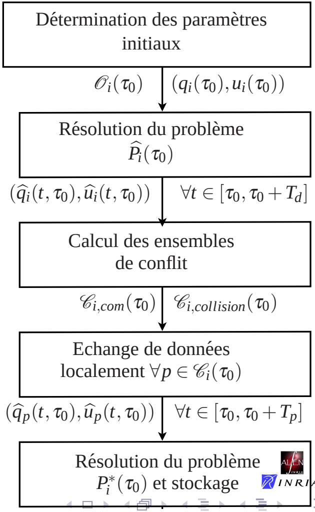
- Intuitive horizon $T_d \in \mathbb{R}^+$
- Planning horizon $T_p \in \mathbb{R}^+$ $(T_p \leq T_d)$
- Update horizon $T_c \in \mathbb{R}^+$ $(T_c \leq T_p)$
- $\widehat{q}_i(t, \tau_k)$, $\widehat{u}_i(t, \tau_k)$: presumed trajectory of robot i beginning at τ_k with $t \in [\tau_k, \tau_k + T_d]$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

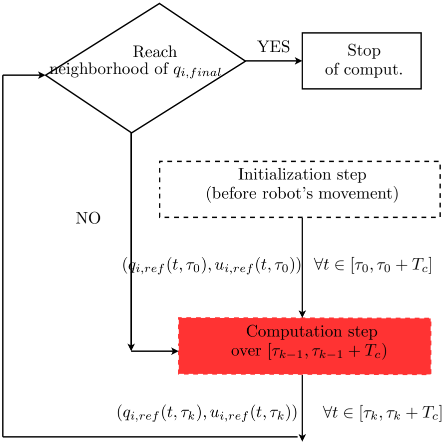


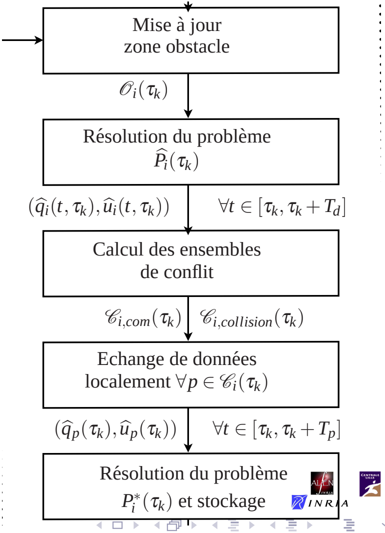
Multi-robots coordination : decentralized approach



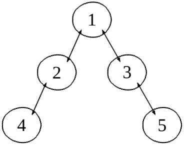


Multi-robots coordination : decentralized approach





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)

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	$35s$	$39s$	$36s$	$36.5s$

Strongly decentralized

Video

Strongly decentralized

Video

Challenges

- Nonlinear dynamics,
- Presence of perturbations (unmodelled dynamics, sensor noise, external disturbances)
- How deal with the stabilization problem at low or zero velocity ?
- How to integrate cooperation into the control design ?
- Leader or not ?

Facts :

- 90 percent of the job is done by nominal control (path planning from which the open loop control is obtained thanks to differential flatness),
- 10 by feedback !

Several solutions were proposed, a challenging problem being control design taking into account cooperation.

Nouvelle architechture de commande (robots communicants) :

- nouveaux pb en Automatique (présence de retards variables, effet de quantification des données, aspects distribués de la commande . . .)
- partage des capteurs extéroceptifs (une caméra associée à un robot, un télémètre à un second, un compteur Geiger-Müller à un troisième, etc.) : traitement de l'information (conditionnement des signaux, fusion de données, etc...)

Quelques points durs :

- la conception et la commande de robots mobiles *via* des réseaux : dynamiques limitées (quelques m/s) ;
- capteurs distants : pbs liés à une identification performante + mesure temps réel précise de l'état du réseaux (perturbations électromagnétiques, pertes d'informations, gigue, retards, etc.) ;
- les QoS (actuelles) sur des réseaux hétérogènes asynchrones filaires et/ou sans fil ne sont pas adaptées en termes de temps de réponse, d'accès au médium pour la commande de processus ;
- les architectures de commandes logicielles et matérielles temps réel devront être adaptées à des systèmes communicants et distribués dont les systèmes d'exploitations seront certainement hétérogènes ainsi que les supports matériels.

La conclusion pour RECAP .... voici un nouveau terrain de jeux où les capteurs “bougent” et donnent naissance à de nouveaux problèmes pour la communauté scientifique

- topologie du réseau change dans le temps et l'espace,
- les entités qui bougent sont plus ou moins connues (modèles cinématique + dynamique : on peut avec les techniques de l'automatique estimer certaines variables).... il faut utiliser ces connaissances,
- il y a déjà eu, du temps des RTP, des interaction automatique/ informatique (commande à travers les réseaux, commande pour la gestion de congestion etc...) : let's do it again...
- ces problèmes d'optimisations en décentralisé ne peuvent-ils pas répondre à d'autres problèmes de RECAP ?

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

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.

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.