Model Predictive Control for Cooperative Guidance of Autonomous Vehicles

Julien Marzat

ICODE Seminar, CENTRALE-SUPELEC, Gif-Sur-Yvette, June 12 2015

ONERA
THE FRENCH AEROSPACE LAB
retour sur innovation

Research Team

Hélène Piet-Lahanier
Sylvain Bertrand
Julien Marzat
Yohan Rochefort (PhD student 2009-2013)
Arthur Kahn (PhD student 2012-)
Christophe Viel (PhD student 2014-)

Didier Dumur
Dominique Beauvois
Michel Kieffer

Topics addressed

Cooperative guidance of fleets of autonomous vehicles

- Global fleet objective, more efficient than sum of individual missions
- Cheaper individual vehicles with complementary sensors
- Decentralized implementation : reduced communication, robustness to vehicle loss, no supervisor
- Safety issues : collisions between vehicles and with obstacles

Model Predicive Control (MPC) - interest and challenges

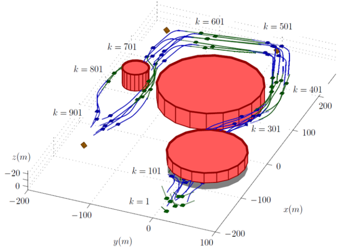
- Takes into account nonlinear vehicle models and constraints
- Same framework with multiple criteria for various missions
- Should be adapted to embedded implementation
- Experiments on mobile and aerial vehicles

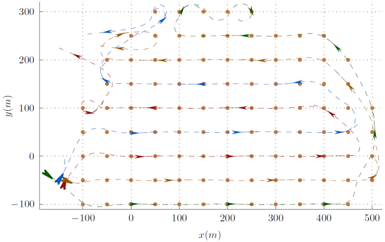
Main goal of this talk

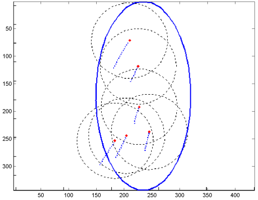
Challenging cooperative problems and possible solutions using MPC

Applications

- Waypoint navigation
  - Formation flight
  - Grid allocation for exploration
- Autonomous trajectory definition
  - Virtual structure formation flight
  - Area exploration with dynamic assignment of exit targets







ONERA
THE FRENCH AEROSPACE LAB

Outline

Outline

Distributed Model Predictive Control

Dynamical models

For each vehicle i, $\mathbf{x}_i(k+1) = f_i(\mathbf{x}_i(k), \mathbf{u}_i(k))$

Future trajectories

$$\mathbf{X}_i(k) = \begin{pmatrix} \mathbf{x}_i(k+1) \\ \mathbf{x}_i(k+2) \\ \vdots \\ \mathbf{x}_i(k+H_p) \end{pmatrix} \text{ and } \mathbf{U}_i(k) = \begin{pmatrix} \mathbf{u}_i(k), \\ \mathbf{u}_i(k+1), \\ \vdots \\ \mathbf{u}_\mathbf{i}(k+H_c-1) \end{pmatrix}$$

H_c control horizon, H_p prediction horizon

Cost function over future trajectories

$$J_i(\mathbf{U}_i(k), \mathbf{X}_i(k)) = \sum_{t=k+1}^{H_p} \varphi_i(\mathbf{x}_i(t), \mathbf{u}_i(t-1), t) + \Phi_i(\mathbf{x}_i(t+H_p))$$

Distributed Model Predictive Control

Optimisation under constraints

$$\text{Find } \mathbf{U}_i^* = \arg\min \; J_i(\mathbf{U}_i(k), \mathbf{X}_i(k))$$
$$\text{over } \mathbf{U}_i \in \mathcal{U}_i^{H_c}$$
$$\text{subject to } \forall t \in [k+1; k+H_p],$$
$$\mathbf{x}_i(t) \in \mathcal{X}_i, \; \mathbf{x}_i(t+1) = f_i(\mathbf{x}_i(t), \mathbf{u}_i(t))$$

Principle

At each timestep, apply the first input of $\mathbf{U}_i^*$ and iterate

Advantages

- includes knowledge of system dynamics and predictions
- natively handles constraints on input and state

Difficulties

- definition of cost function J_i
- solve a costly optimization problem at each timestep

Simplifying assumptions : identical vehicles, no communication delays

Typical vehicle model

2D dynamical model (straightforward 3D extension)

$$\mathbf{x}_i = (x_i, y_i, v_i, \chi_i)^T \text{ and } \mathbf{u}_i = (u_i^\omega, u_i^v)^T$$

(x, y) position, v speed, χ orientation,
(u^ω, u^v) angular and linear accelerations

$$\mathbf{x}_i(k+1) = f(\mathbf{x}_i(k), \mathbf{u}_i(k)) \text{ and } (\mathbf{x}_i, \mathbf{u}_i) \in \mathcal{X}_i \times \mathcal{U}_i \text{ such that}$$

$$\left\{ \begin{array}{rcl} x_i(k+1) & = & x_i(k) + \Delta t.v_i(k) \cos \chi_i(k) \\ y_i(k+1) & = & y_i(k) + \Delta t.v_i(k) \sin \chi_i(k) \\ v_i(k+1) & = & v_i(k) + \Delta t.u_i^v(k) \\ \chi_i(k+1) & = & \chi_i(k) + \Delta t.u_i^\omega(k) \end{array} \right.$$

$$\begin{array}{cc} v_{min} \leq v_i \leq v_{max} & -\omega_{max} \leq \omega_i \leq \omega_{max} \\ -\Delta v_{max} \leq u_i^v \leq \Delta v_{max} & -\Delta\omega_{max} \leq u_i^\omega \leq \Delta\omega_{max} \end{array}$$

Trigonometric nonlinearity, either on the state or on the input space

ONERA THE FRENCH AEROSPACE LAB

Basic costs for autonomous vehicle guidance

Lagrangian with mission and penalized constraint costs

$$J_i(k) = J_i^{nav}(k) + J_i^{safety}(k) + J_i^{u}(k)$$

Weights $W^{\bullet}$ for normalization and setting relative priorities

- Control cost $J_i^u = \sum_{n=k+1}^{k+H_c} W^{u,\omega} u_i^{\omega}(n)^2 + W^{u,v} u_i^{v}(n)^2$
- Navigation cost $J_i^{nav} = J_i^{nav,direct} + J_i^{nav,fleet}$
  Given a waypoint $\mathbf{p}_p$ and predicted robot position $\widehat{\mathbf{p}}_i(n|k)$,

$$\mathbf{p}_{i,p}^{ref}(n|k) = \mathbf{p}_i(k) + (n-k)\,\Delta t\, v_i \frac{\mathbf{p}_i(k) - \mathbf{p}_p}{\|\mathbf{p}_i(k) - \mathbf{p}_p\|}$$

$$J_i^{nav,direct} = W^{nd} \sum_{n=k+1}^{k+H_p} \left\|\widehat{\mathbf{p}}_i(n|k) - \mathbf{p}_{i,p}^{ref}(n|k)\right\|$$

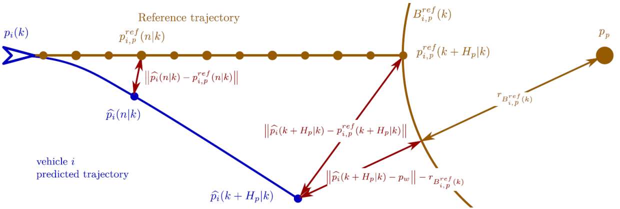
Basic costs for autonomous vehicle guidance

Lagrangian with mission and penalized constraint costs

$$J_i(k) = J_i^{nav}(k) + J_i^{safety}(k) + J_i^{u}(k)$$

Weights $W^{\bullet}$ for normalization and setting relative priorities

- Control cost $J_i^u = \sum_{n=k+1}^{k+H_c} W^{u,\omega} u_i^{\omega}(n)^2 + W^{u,v} u_i^{v}(n)^2$
- Navigation cost $J_i^{nav} = J_i^{nav,direct} + J_i^{nav,fleet}$



Attraction / repulsion costs

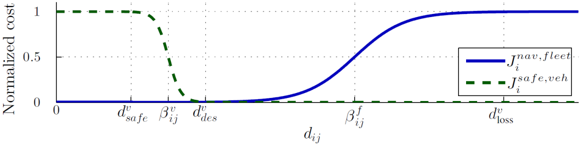
- Navigation cost (continued)

$$J_i^{nav,fleet} = W^{nv} \sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1 + \tanh\left(\alpha_{ij}^{f}\left(d_{ij}(n|k) - \beta_{ij}^{f}\right)\right)}{2}$$

- Safety cost $J_i^{safety} = J_i^{safe,veh}(k) + J_i^{safe,obs}(k)$

$$J_i^{safe,veh} = W^{sv} \sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1 - \tanh\left(\alpha_{ij}^{v}\left(d_{ij}(n|k) - \beta_{ij}^{v}\right)\right)}{2} \quad (N \text{ vehicles})$$

$$J_i^{safe,obs} = W^{so} \sum_{o=1}^{N^o} \sum_{n=k+1}^{k+H_p} \frac{1 - \tanh\left(\alpha_{io}^{o}\left(d_{io}(n|k) - \beta_{io}^{o}\right)\right)}{2} \quad (N^o \text{ obstacles})$$



The delicate question of weighting

Multi-objective optimization problem under constraints, such that

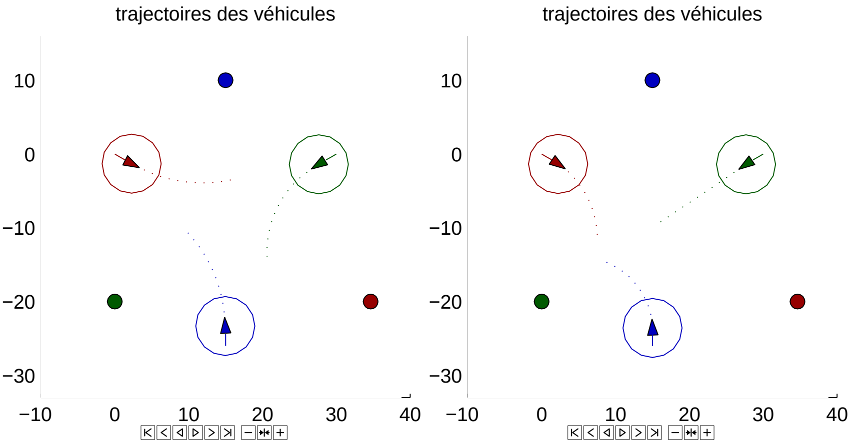
- constraints on inputs only $\Rightarrow$ natively taken into account
- constraints on state $\Rightarrow$ penalization costs
- weighted sums of sub-costs

How to determine weights without too much ad-hoc tuning ?

- normalize sub-costs between 0 and 1
- choose weights such that penalization terms are the largest ones in constraint regions
  - navigation *vs* control sollicitation = classical LQ trade-off between tracking and energy consumption
  - collision *vs* other costs = several orders of magniture difference, such that safety is the only significant cost in defined regions

Illustration of basic costs

$d_{des} = 6, d_{safe} = 4$ $\qquad$ $d_{des} = 8, d_{safe} = 4$



Computational issues

MPC guidance problems involve nonconvex and multimodal cost functions on constrained input spaces

How to compute the (sub)optimal cost during one timestep ?
$\Rightarrow$ Methods sorted by increasing computation time

- Discretization (fixed computation time)
  - deterministic grid
  - random search
- Local search : gradient descent and variants
- Global search : large choice of expensive optimizers

No free lunch ! We usually choose :

- Deterministic grid on architectures with low computational resources
- Global optimizers on more powerful embedded computers

Discretized search

Heuristic rules

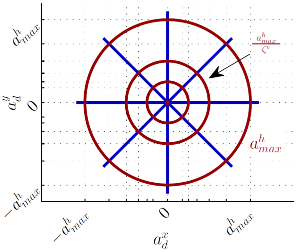
- Define a set of control candidates such that
  - null and extremal control inputs are included
  - candidates are distributed over the entire control space with increased density around null control input
  - same control input value on the entire control horizon
- Predict cost value for each candidate trajectory and select the best

Example in the 2D space :

$$\mathcal{S}^{\omega} = \left\{ \frac{2\pi p}{\eta^{\omega}} \right\},\ p = 1 \dots \eta^{\omega}$$

$$\mathcal{S}^{v} = \left\{ \frac{\Delta v_{max}}{(\eta^{v})^{p}} \right\},\ p = 0 \dots \eta^{v}$$

$$\mathcal{S} = \{\mathcal{S}^{v} \times \mathcal{S}^{\omega}\} \cup \{0, 0\}$$



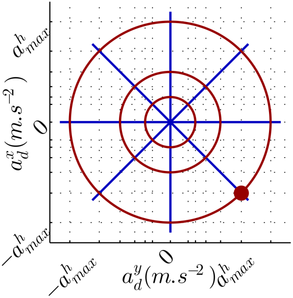
ONERA

THE FRENCH AEROSPACE LAB

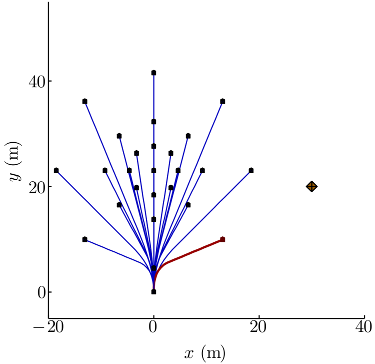
Discretized search - illustration

Itération 1 ($\mathbf{v}_i\,(1) = [0\ 2\ 0]$ m. s^{-1})

Espace de commande



Trajectoires correspondantes sur l'horizon de prédiction



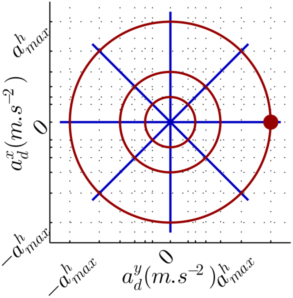
ONERA

THE FRENCH AEROSPACE LAB

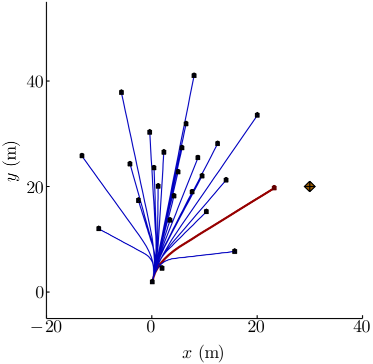
Discretized search - illustration

Itération 3 ($\mathbf{v}_i\,(1) = [0\ 2\ 0]$ m. s^{-1})

Espace de commande



Trajectoires correspondantes sur l'horizon de prédiction

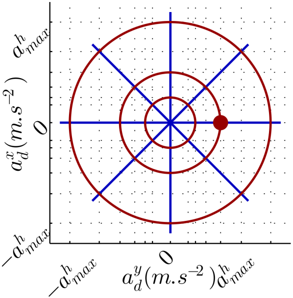


ONERA
THE FRENCH AEROSPACE LAB

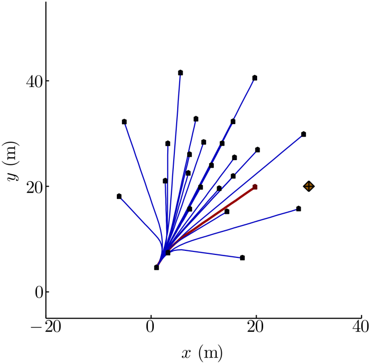
Discretized search - illustration

Itération 6 ($\mathbf{v}_i(1) = [0\ 2\ 0]$ m. s^{-1})

Espace de commande



Trajectoires correspondantes sur l'horizon de prédiction

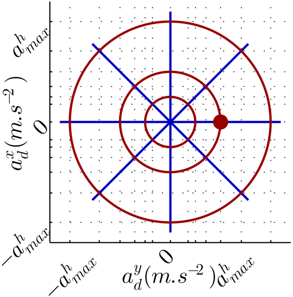


ONERA
THE FRENCH AEROSPACE LAB

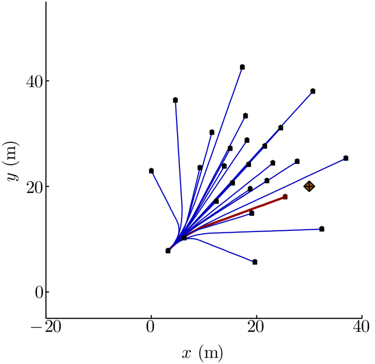
Discretized search - illustration

Itération 10 ($\mathbf{v}_i(1) = [0\ 2\ 0]\ \mathrm{m.s^{-1}}$)

Espace de commande



Trajectoires correspondantes sur l'horizon de prédiction

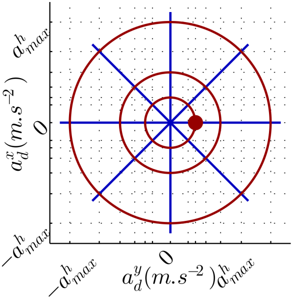


ONERA
THE FRENCH AEROSPACE LAB

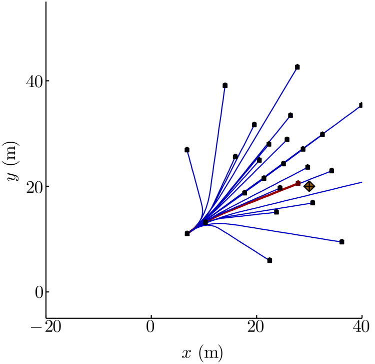
Discretized search - illustration

Itération 15 ($\mathbf{v}_i(1) = [0\ 2\ 0]\ \mathrm{m.s^{-1}}$)

Espace de commande



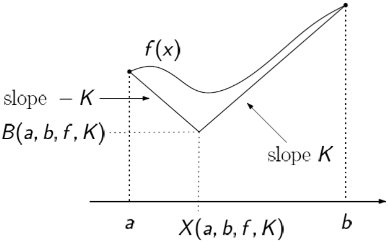
Trajectoires correspondantes sur l'horizon de prédiction

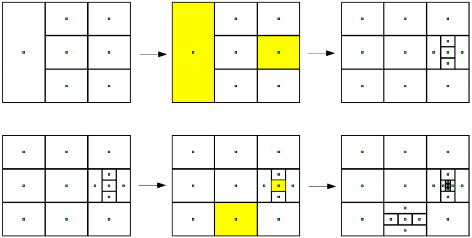


ONERA
THE FRENCH AEROSPACE LAB

Global optimizer : DIRECT (DIviding RECTangles)

Lipschitzian optimization (without knowledge of the Lipschitz constant)





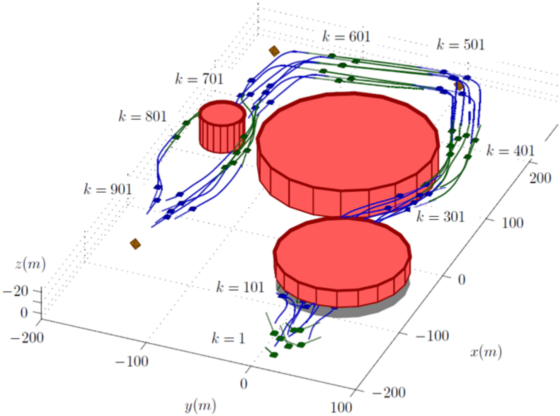
Matlab and C++ versions

Very efficient implementation in Python package *nlopt*

Outline

Guidance toward predefined objectives

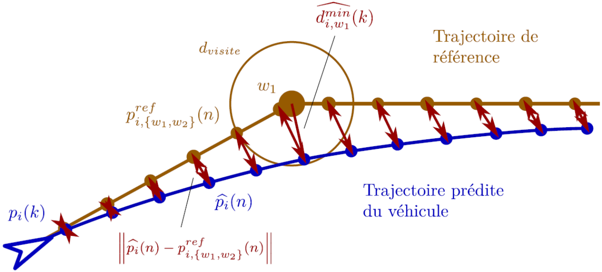
- Application of MPC fleet costs to realistic 3D quadrotor models
- Discretized search approach



ONERA
THE FRENCH AEROSPACE LAB

Cooperative waypoint grid allocation for exploration

- Navigation cost modified to take into account 2 successive waypoints
- Consensus procedure based on this cost, computed from each vehicle to the nearest candidate successive waypoints



Outline

Virtual structure approach for formation flight

Three main approaches for formation flying control

- **Leader following** (Wang 1991, Desai 1998, Jadbabaie 2003):
  One agent is defined as more important than the others. The others will be dependent of the leader.
- **Behaviour rules** (Parker 1998, 2012, Balch 1998):
  The agents must follow some rules depending on the environment and the mission: approach described so far.
- **Virtual structure** (Lewis 1996, Barnes 2009, Bacon 2012, Ren 2004):
  A virtual link is defined to move the agents together. The virtual structure can be fixed or evolve depending on the environment.

Virtual structure approach for formation flight

Formation control by restraining the UAVs inside an area

Double layer control:

- Higher layer: Virtual structure control
  - Reach the final destination of the fleet
  - Collision avoidance with obstacles by shaping the ellipse
- Lower layer: decentralized UAV control
  - Reach the area
  - Repartition within the area
  - Collision avoidance between agents

Virtual structure : higher layer

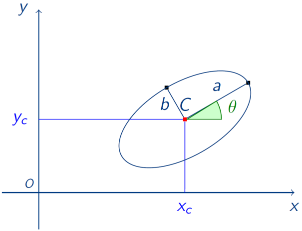
Ellipse of center $\mathbf{p_c} = [x_c \ y_c]$ and characteristic matrix $\mathbf{M}$
Defined for every point of the space $\mathbf{p} = [x \ y]$ as:

$$(\mathbf{p} - \mathbf{p_c})^T \mathbf{M}^{-1} (\mathbf{p} - \mathbf{p_c}) \leq 1$$

Characteristic matrix $\mathbf{M}$:

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T$$

θ: angle between the long axis and the horizontal
a : length of the long axis of the ellipse
b : length of the short axis of the ellipse



Virtual structure : higher layer

Search for the inputs such that:

$$\widehat{u}_\theta, \widehat{u}_a, \widehat{u}_b, \widehat{u}_v, \widehat{u}_\alpha = \arg \min_{\substack{u_v,\, u_\alpha, \\ u_\theta,\, u_a,\, u_b,}} J_z$$

where

$$J_z = J_{target} + J_v + J_{ab} + J_c.$$

The components of J_z are designed such that

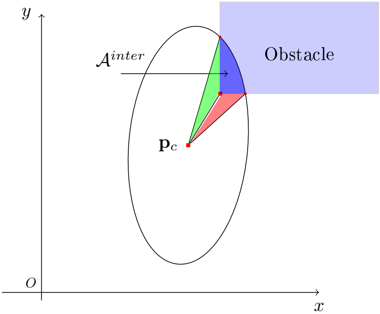
- J_{target} drives the ellipse to its target;
- J_c modifies the matrix $\mathbf{M}$ to avoid obstacles.
- J_v keeps the ellipse area close to the initial one, $\mathcal{A}$;
- J_{ab} keeps a and b close to their initial values a_0 and b_0;

Virtual structure : higher layer

Collision avoidance with obstacles

$$J_c = w_c \sum_{k=1}^{HP} \sum_{l=1}^{NbO} \frac{HP-k}{HP} . \mathcal{A}_{l,t+k}^{inter}$$

Computation of the area of intersection



Virtual structure : lower layer

UAV control

Search the inputs of each agent i such that:

$$u_i^v, u_i^\omega = \arg\min J_i^d \tag{1}$$

where

$$J_i^d = J_i^t + J_i^{safe} + J_i^{n1} + J_i^{n2} + J_i^u \tag{2}$$

The components of J_d are designed such that

- J_i^t drives the UAV inside the area;
- J_i^{safe} modifies the direction and the speed to avoid collision with other UAVs;
- J_i^{n1} keeps the speed of the UAV close to a chosen value;
- J_i^{n2} keeps the orientation of the UAV close to the one of the structure;
- J_i^u minimizes the energy consumption in terms of control inputs.

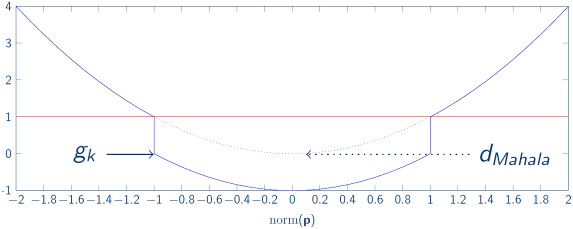
Virtual structure : lower layer

Attraction of the UAVs toward the center of the area :

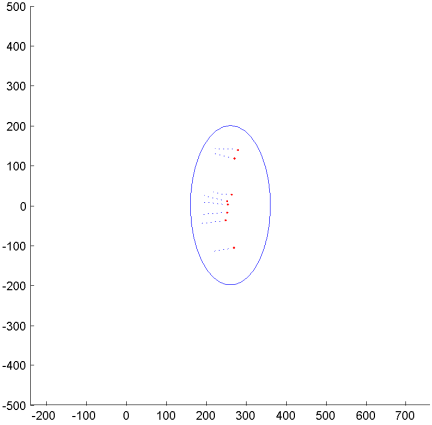
$$J_i^t = w_t \sum_{k=1} \frac{(HP-k)}{HP} g_k(i),$$

Potential field in the area $g_k(i)$ derived from the Mahalanobis distance.

$$d_{Mahala}(\mathbf{p}) = \sqrt{(\mathbf{p}-\mathbf{p}_c)^T \mathbf{M}(\mathbf{p}-\mathbf{p}_c)}$$

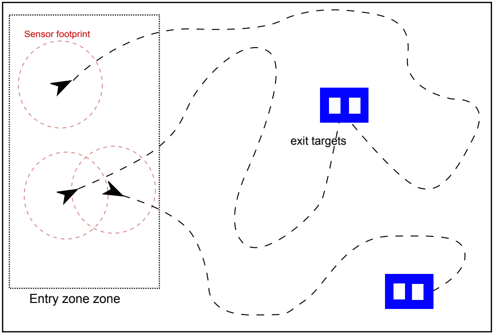


Virtual structure - simulation results



HP_{zone} =30, HC_{zone} =5, $a_{initial}$ =200, $b_{initial}$ =100, $\alpha_{ellipse} = \frac{\pi}{2}$, v_0 =4

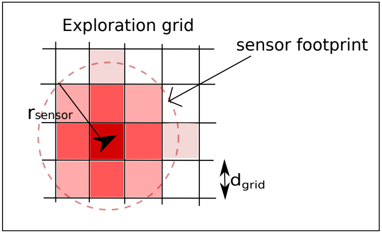
Area exploration with dynamic target assignment



- n_v vehicles and n_c exit locations
- 2 objectives to fulfill:
  - Online trajectory planning that favour exploration
  - Online reassignment of targets
- Constraints:
  - Constrained dynamics
  - Collision avoidance
  - Fixed mission time

Area exploration with dynamic target assignment

Exploration reward: we want to maximize $\Omega = \bigcup_{\substack{t=1..t_f \\ i=1..n_v}} \mathcal{D}_i^t$



Discrete representation:
Matrix G represents the level of exploration of a cell, $G_{kl} \in [0,1]$.
When a vehicle comes at distance d of G_{kl}, the exploration level obtained is given by $f_{explore}$

$$f_{explore}(d) = \begin{cases} 0 & \text{si } d \geq r_{sensor} \\ \frac{1}{2}\left(1 + cos\left(\frac{\pi d}{r_{sensor}}\right)\right) & \text{if } d < r_{sensor} \end{cases}$$

Cost function : $J_i^{gri} = W^{gr}.(\widehat{G}_{t_0+H_p} - G_{t_0})$

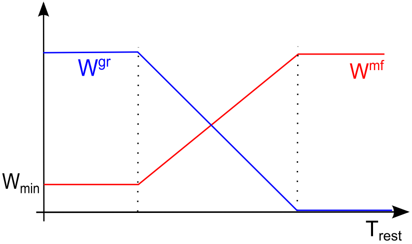
ONERA
THE FRENCH AEROSPACE LAB

Area exploration with dynamic target assignment

Navigation cost to exit targets

Time-varying weighting

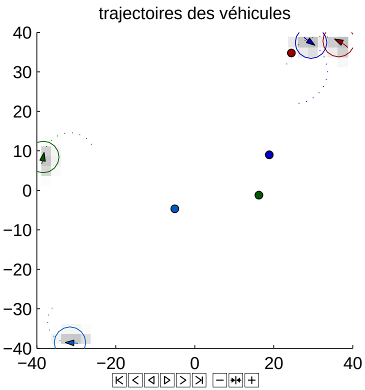
- Useless to move immediately toward the target
- Temporal management of priorities may be beneficial
  - Assignment is made at the beginning of the mission
  - Weighting is made by taking into account the remaining time and the distance to the target
  - We verify that exit constraints are satisfied at the end of the mission



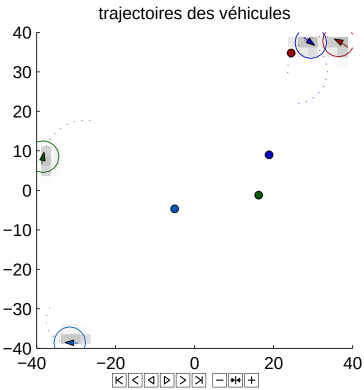
ONERA THE FRENCH AEROSPACE LAB

Area exploration with dynamic target assignment

With fixed weighting



With dynamic weighting



Area exploration with dynamic target assignment

Optimal dynamic assignment of the exits:
balance between distance to target and remaining time

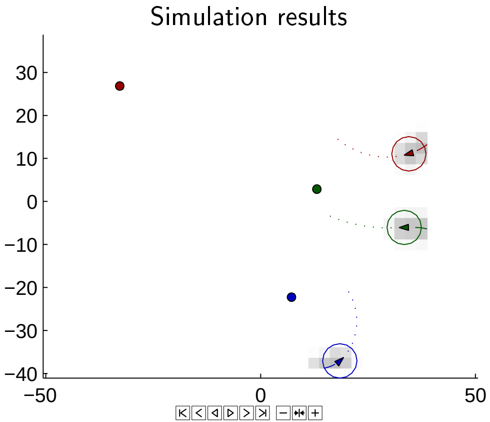
$$\text{Matrix of vehicle/target costs } r_{ij} = \arg\min_{\mathbf{U}_i} J_{ij}$$

$$J_{ij} = \text{Control cost} + \text{distance} + \text{remaining time}$$

Costs are centralised and an optimal assignment of the targets is performed at each timestep with the Hungarian algorithm
Three cases taken into account (iterative assignments)

- One vehicle per target and $n_c = n_v$
- At most n_{max} vehicles per target
- At least n_{min} vehicles per target

Area exploration with dynamic target assignment



expl/dyn. reassign.	no/no	yes/no	yes/yes
Average coverage	21%	45%	58%

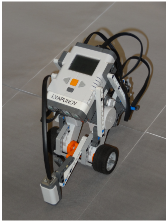
Outline

ONERA

THE FRENCH AEROSPACE LAB

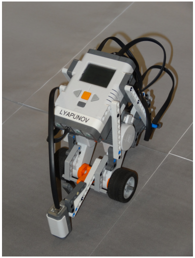
Experimental robotic platforms

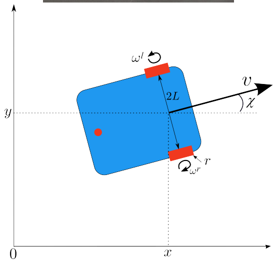
- Mobile robots
  - LEGO Mindstorms NXT
  - E-puck (ICODE funding)
  - Robotnik Summit XL
- Aerial vehicles
  - Parrot AR Drones
  - Asctec Pelican



LEGO Mindstorms NXT mobile robots

- Computational capabilities:
  ARM 48MHz with ATMega 20MHZ
  64kB DRAM and 256 kB memory
- NXC language
- Bluetooth communication:
  - 1 master and at most 3 slaves
  - position information shared at 20Hz between 2 robots
- 2-wheel differential structure, wheel encoders (accuracy $\pm 1°$)
- Easy integration of a wide range of sensors





ONERA
THE FRENCH AEROSPACE LAB

Objectives

- Many cooperative guidance laws in the literature, mostly evaluated in simulation
- Assess whether cooperative guidance laws and distributed estimation can be applied on robots with limited computing capacities
- Search for a flexible, low-cost robotic experimental platform for cooperative guidance $\rightarrow$ tests on Lego Mindstorms NXT
- Demonstration scenario: fleet coordination with collision/obstacle avoidance

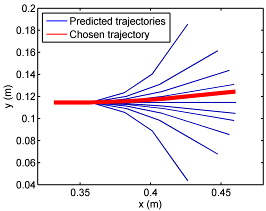
Search for (sub)optimal cost

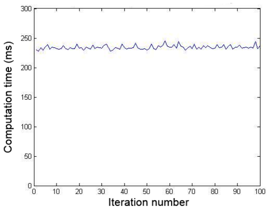
Discretization of the space as a set $\mathcal{S}$ of candidate control inputs where the cost J_i is computed and the argument of the smallest is applied

1. The same control input is applied at all control steps on H_c
2. $\mathcal{S}$ includes the null and extreme control inputs
3. Increased density around the null control input.

Here, $\mathcal{S}$ reduces to, with a varying step γ,

$$\mathcal{S} = \frac{2\pi\gamma}{\eta^{\omega}} \qquad \text{with } \gamma \in [1, \eta^{\omega}]$$

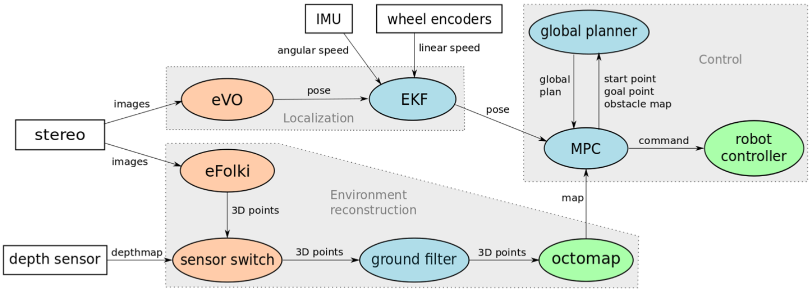




Experimental results - Lego Mindstorms

Experimental results - Robotnik XL

Vision-based autonomous exploration and mapping
Embedded optimization using nlopt/DIRECT



Outline

Conclusions and perspectives

Summary

- Unified MPC framework for cooperative guidance
- Generic cost functions for basic tasks
- Many different concepts needed to address realistic problems
- Experiment-oriented solutions for optimal input selection
- Successful first experiments on mobile robots

Perspectives

- Take into account delays and reduced inter-vehicle communication
- Cooperative localization with distributed vision sensors
- Experiments on fleets of aerial vehicles for autonomous environment mapping and formation flight

References

1. Bertrand S., Marzat J., Piet-Lahanier H., Kahn A., Rochefort Y., MPC Strategies for Cooperative Guidance of Autonomous Vehicles, Aerospace Lab Journal, vol. 8, 2014.
2. Rochefort Y., Piet-Lahanier H., Bertrand S., Beauvois D., Dumur D., Model predictive control of cooperative vehicles using systematic search approach, Control Engineering Practice, 2014.
3. Marzat J., Piet-Lahanier H., Kahn A., Cooperative guidance of Lego Mindstorms NXT mobile robots, INSTICC/IEEE ICINCO 2014.
4. Gorecki T., Piet-Lahanier H., Marzat J., Balesdent M., Cooperative guidance of UAVs for area exploration with final target allocation, IFAC ACA 2013.
5. Piet-Lahanier H., Kahn A., Marzat J., Cooperative guidance laws for manoeuvring target interception, IFAC ACA 2013.
6. Kahn A., Marzat J., Piet-Lahanier H., Formation flying control via elliptical virtual structure, IEEE ICNSC 2013.
7. Rochefort Y., Bertrand S., Piet-Lahanier H., Beauvois D., Dumur D., Cooperative nonlinear model predictive control for flocks of vehicles, IFAC EGNCA 2012.
8. Rochefort Y., Piet-Lahanier H., Bertrand S., Beauvois D., Dumur D. Guidance of flocks of vehicles using virtual signposts. IFAC World Congress 2011.

ONERA
THE FRENCH AEROSPACE LAB