From Centralized to Decentralized Control of Complex Systems

In the Pursuit of Efficient Control Approaches

Ionela Prodan

Université Grenoble Alpes, Laboratoire de Conception et d'Intégration des Systèmes (LCIS EA 3747), Valence, F-26009, France,
ionela.prodan@lcis.grenoble-inp.fr,
https://sites.google.com/site/iprodanionela/

Fall semester 2019

Lecture 4

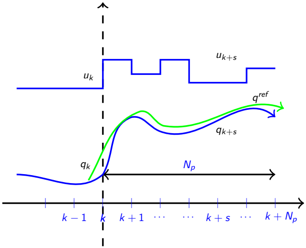
Outline

1. An optimization-based approach for control of complex systems
   - Mixed-integer techniques
   - Flat trajectory

Model Predictive Control (MPC)

Propoi (1963), Cutler et al. (2007), Richalet and O'Donovan (2009)

- Optimization-based control law
- Implicit (on-line) vs. explicit (off-line) implementation
- Constraints handling
- Can be implemented in a distributed fashion



model of the system

reference trajectory

$$\arg\min_{\mathbf{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to: } \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1, \\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}$$

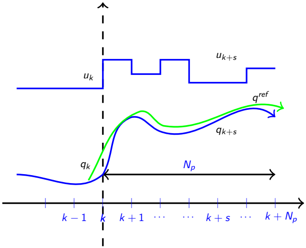
constraint sets representation

optimization solver

Model Predictive Control (MPC)

Propoi (1963), Cutler et al. (2007), Richalet and O'Donovan (2009)

- Optimization-based control law
- Implicit (on-line) vs. explicit (off-line) implementation
- Constraints handling
- Can be implemented in a distributed fashion



model of the system

reference trajectory

$$\arg\min_{\mathbf{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to: } \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1, \\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}$$

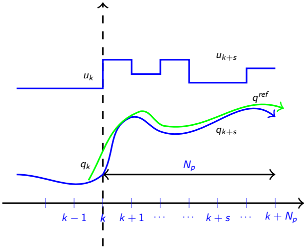
constraint sets representation

optimization solver

Model Predictive Control (MPC)

Propoi (1963), Cutler et al. (2007), Richalet and O'Donovan (2009)

- Optimization-based control law
- Implicit (on-line) vs. explicit (off-line) implementation
- Constraints handling
- Can be implemented in a distributed fashion



model of the system

reference trajectory

$$\arg\min_{\mathbf{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to: } \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1, \\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}$$

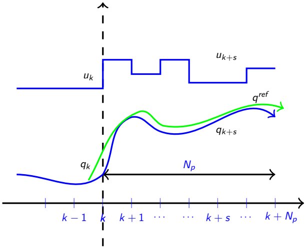
constraint sets representation

optimization solver

Model Predictive Control (MPC)

Propoi (1963), Cutler et al. (2007), Richalet and O'Donovan (2009)

- Optimization-based control law
- Implicit (on-line) vs. explicit (off-line) implementation
- Constraints handling
- Can be implemented in a distributed fashion



model of the system

reference trajectory

$$
\begin{aligned}
&\arg\min_{\mathbf{u}} V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))\\
&\text{subject to: } \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1,\\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}
\end{aligned}
$$

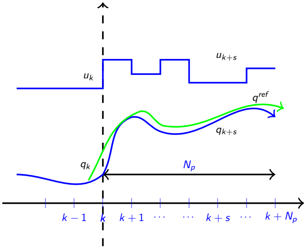
constraint sets representation

optimization solver

Model Predictive Control (MPC)

Propoi (1963), Cutler et al. (2007), Richalet and O'Donovan (2009)

- Optimization-based control law
- Implicit (on-line) vs. explicit (off-line) implementation
- Constraints handling
- Can be implemented in a distributed fashion



model of the system

reference trajectory

$$
\begin{aligned}
&\arg\min_{\mathbf{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))\\
&\text{subject to:} \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1,\\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}
\end{aligned}
$$

constraint sets representation

optimization solver

Mixed-Integer Programming (MIP)

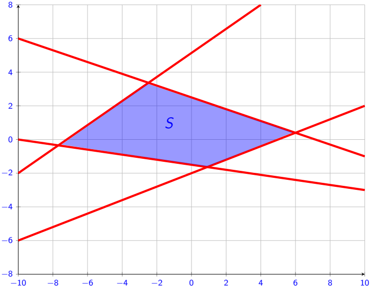
Grundel et al. (2007), Jünger et al. (2009)

- Flexible mathematical model for the formulation of decision and control problems based on optimization
  - combinatorial allocation problem
  - multicast routing problem
- Flexible mathematical model for the formulation of collision avoidance problems involving the control of Multi-Agent Systems
  - path following with obstacle and collision avoidance
  - formation control with collision avoidance
- Fast off-the-shelf solvers available
  - CPLEX, OSL, etc.
- Strong theoretical foundations
  - characterization of tractable special cases
  - NP-hard in general, but can also solve many large problems in practice

MIP – multi-agent context

Consider a bounded polyhedral set

$$S = \left\{x \in \mathbb{R}^n : h_i x \leq k_i,\ i = 1 : N\right\}$$



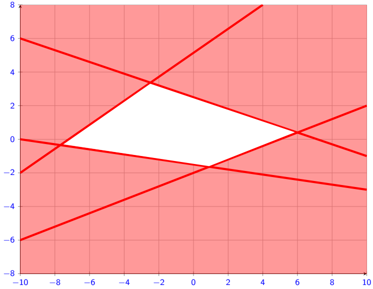
MIP – multi-agent context

Consider a bounded polyhedral set

$$S = \left\{ x \in \mathbb{R}^n : h_i x \leq k_i,\ i = 1 : N \right\}$$

Consider the complement of S

$$\mathcal{C}(S) \triangleq cl(\mathbb{R}^n \setminus S) = \bigcup_i \mathcal{R}^-(\mathcal{H}_i), \quad i = 1 : N$$



MIP – multi-agent context

Consider a bounded polyhedral set

$$S = \left\{x \in \mathbb{R}^n : h_i x \leq k_i,\ \ i = 1 : N\right\}$$

Consider the complement of S

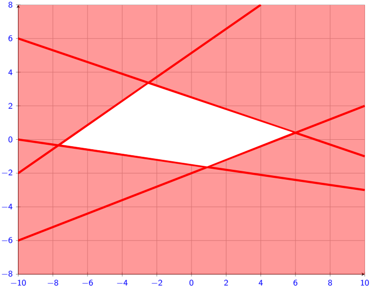
$$\mathcal{C}(S) \triangleq cl(\mathbb{R}^n \setminus S) = \bigcup_i \mathcal{R}^-(\mathcal{H}_i), \quad i = 1 : N$$

Define $\mathcal{C}(S)$ in a linear representation

$$-h_i x \leq -k_i + M\alpha_i, \quad i = 1 :\ N$$

$$\sum_{i=1}^{i=N} \alpha_i \leq N - 1$$

with $(\alpha_1, \ldots, \alpha_N) \in \{0,1\}^{N}$



MIP – multi-agent context

Consider a bounded polyhedral set

$$S = \left\{x \in \mathbb{R}^n : h_i x \le k_i,\ i = 1 : N\right\}$$

Consider the complement of S

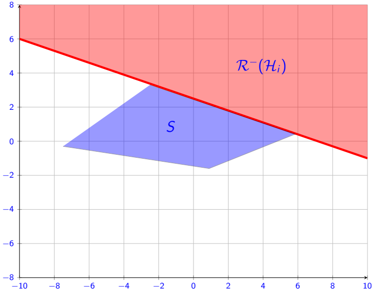
$$\mathcal{C}(S) \triangleq cl(\mathbb{R}^n \setminus S) = \bigcup_i \mathcal{R}^-(\mathcal{H}_i), \quad i = 1 : N$$

Define $\mathcal{C}(S)$ in a linear representation

$$-h_i x \le -k_i + M\alpha_i, \quad i = 1 :\ N$$

$$\sum_{i=1}^{i=N} \alpha_i \le N - 1$$

with $(\alpha_1, \ldots, \alpha_N) \in \{0, 1\}^{\ N}$



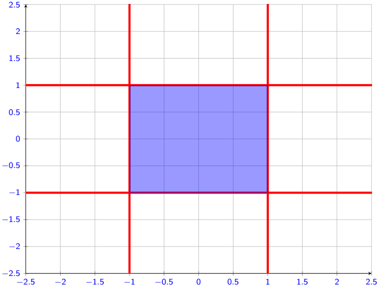
Any of the regions $\mathcal{R}^-(\mathcal{H}_i)$ of $\mathcal{C}(S)$ can be obtained by a suitable choice of binary variables

$$\mathcal{R}^-(\mathcal{H}_i) \longleftrightarrow (\alpha_1, \ldots, \alpha_N)^i \triangleq (1, \ldots, 1, \underbrace{0}_{i}, 1, \ldots, 1)$$

MIP representation – Illustrative example

Consider a polytope $P \subset \mathbb{R}^2$ given by

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

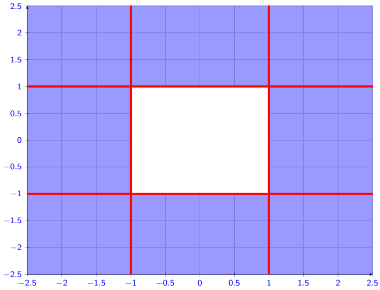


MIP representation – Illustrative example

and its complement $\mathcal{C}(P)$ by

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} -1 + M\alpha_1 \\ -1 + M\alpha_2 \\ -1 + M\alpha_3 \\ -1 + M\alpha_4 \end{bmatrix}$$

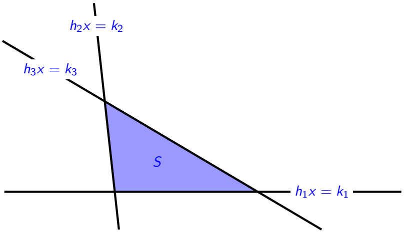
in the classical mixed-integer formulation.



MIP representation – Illustrative example

Consider a triangle from $\mathbb{R}^2$ given by

$$\begin{aligned} h_1 x &\leq k_1 \\ h_2 x &\leq k_2 \\ h_3 x &\leq k_3 \end{aligned}$$

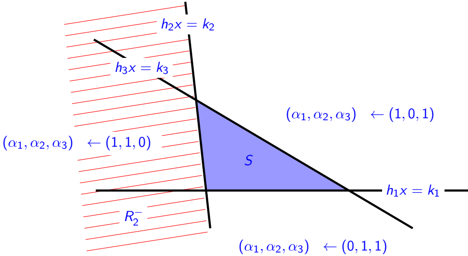


MIP representation – Illustrative example

and its complement

$$\begin{aligned} -h_1 x &\leq -k_1 + M\alpha_1 \\ -h_2 x &\leq -k_2 + M\alpha_2 \\ -h_3 x &\leq -k_3 + M\alpha_3 \end{aligned}$$

in the classical mixed-integer formulation.



MIP representations

Do we really need N binary variables for representing the complement of a convex region?

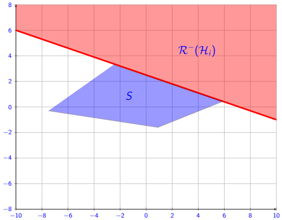
Logarithmic representation

For each region $\mathcal{R}^-(\mathcal{H}_i)$ a unique combination of binary variables $\lambda^i \in \{0,1\}^{\lceil log_2 N \rceil}$ is associated. Then, the affine functions $\alpha_i : \{0,1\}^{\lceil log_2 N \rceil} \to \{0\} \cup [1, \infty)$ are constructed:

$$\alpha_i(\lambda) = \sum_{k=0}^{\lceil log_2 N \rceil} \left(\lambda_k^i + (1 - 2\lambda_k^i) \cdot \lambda_k \right).$$

λ_k denotes the kth component of λ and λ_k^i its value for the tuple associated to region $\mathcal{R}^-(\mathcal{H}_i)$:

$$\alpha_i(\lambda) = \begin{cases} 0, & \text{only if } \lambda = \lambda^i \\ \geq 1, & \text{for any } \lambda \neq \lambda^i \end{cases}$$

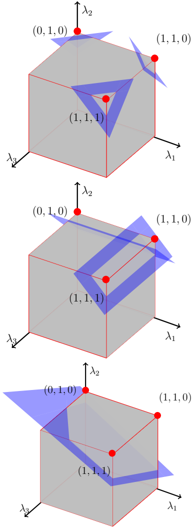


which leads to the compact formulation

$$-h_i x \leq -k_i + M\alpha_i(\lambda), \quad i = 1 : N,$$
$$0 \leq \beta_l(\lambda).$$

Interdicted tuples

In the mixed-integer representation we interdict tuples which describe the obstacle:



- in the classical formulation we force that at least one constraint is active:

$$\sum_{i=1}^{i=N} \alpha_i \leq N - 1$$

- in the logarithmic formulation
  - multiple constraints to interdict tuples Prodan et al. (2012b)

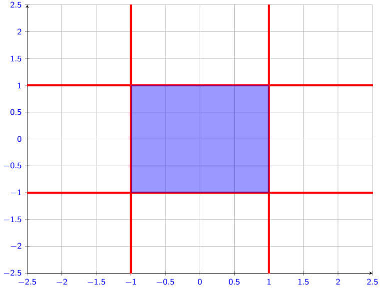
$$0 < \beta_l(\lambda)$$

  - if the allocated tuples are ordered a single constraint suffices Afonso and Galvão (2013)

Illustrative example

Consider a polytope $P \subset \mathbb{R}^2$ given by

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

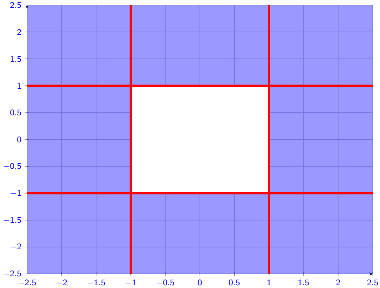


Illustrative example

and its complement $\mathcal{C}(P)$ by

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} -1 + M\alpha_1 \\ -1 + M\alpha_2 \\ -1 + M\alpha_3 \\ -1 + M\alpha_4 \end{bmatrix}$$

in the classical MI formulation.

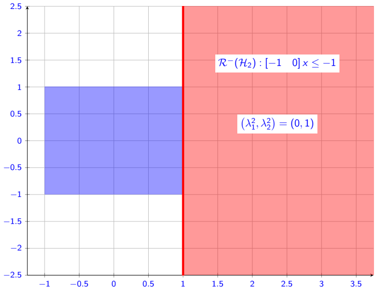


Illustrative example

and its complement $\mathcal{C}(P)$ by

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} -1 + M(\quad \lambda_1 + \lambda_2) \\ -1 + M(1 - \lambda_1 + \lambda_2) \\ -1 + M(1 + \lambda_1 - \lambda_2) \\ -1 + M(2 - \lambda_1 - \lambda_2) \end{bmatrix}$$

in the reduced MI formulation.



In the reduced representation only $N_0 = \lceil \log_2 4 \rceil = 2$ binary variables are needed.

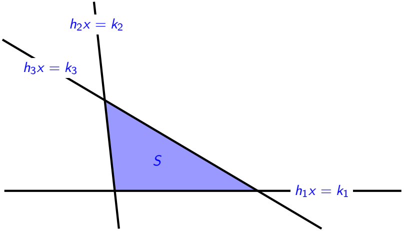
For region $\mathcal{R}^-(\mathcal{H}_2)$ associate tuple $(\lambda_1^2, \lambda_2^2) = (0, 1)$ which leads to the mapping

$$\alpha_2 = 1 + \lambda_1 - \lambda_2$$

Illustrative example

Consider a triangle from $\mathbb{R}^2$ given by

$$\begin{aligned} h_1 x &\leq k_1 \\ h_2 x &\leq k_2 \\ h_3 x &\leq k_3 \end{aligned}$$

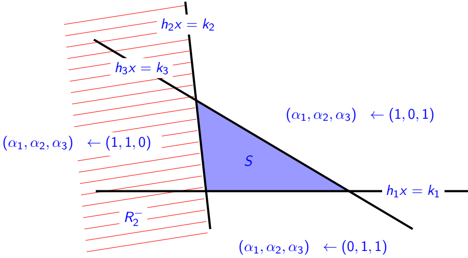


Illustrative example

and its complement

$$\begin{aligned} -h_1 x &\leq -k_1 + M\alpha_1 \\ -h_2 x &\leq -k_2 + M\alpha_2 \\ -h_3 x &\leq -k_3 + M\alpha_3 \end{aligned}$$

in the classical MI formulation.

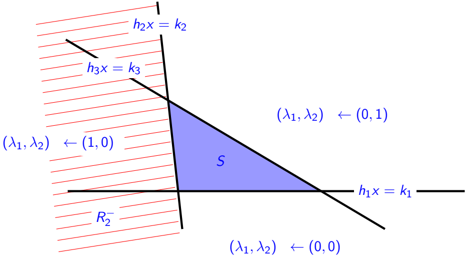


Illustrative example

and its complement

$$\begin{aligned} -h_1x &\leq -k_1 + M(\lambda_1 + \lambda_2) \\ -h_2x &\leq -k_2 + M(1 - \lambda_1 + \lambda_2) \\ -h_3x &\leq -k_3 + M(1 + \lambda_1 - \lambda_2) \end{aligned}$$

in the reduced MI formulation.



In the reduced representation only $N_0 = \lceil \log_2 3 \rceil = 2$ binary variables are needed.

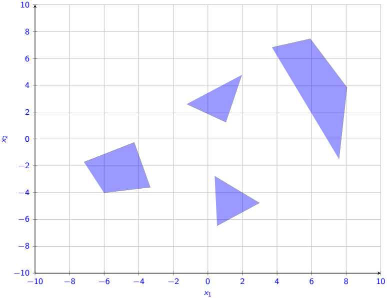
For region $\mathcal{R}^-(\mathcal{H}_2)$ associate tuple $(\lambda_1^2, \lambda_2^2) = (1, 0)$ which leads to the mapping

$$\alpha_2(\lambda) = 1 - \lambda_1 + \lambda_2$$

Non-connected and non-convex regions

Consider the complement $\mathcal{C}(\mathbb{S}) = cl(\mathbb{R}^n \setminus \mathbb{S})$ of a union of polyhedral sets $\mathbb{S} = \bigcup_l S_l$

$$\mathcal{A}(\mathbb{H}) = \bigcup_{l=1,\ldots,\gamma(N)} \underbrace{\left(\bigcap_{i=1}^{N} R^{\sigma_l(i)}(\mathcal{H}_i) \right)}_{A_l}$$

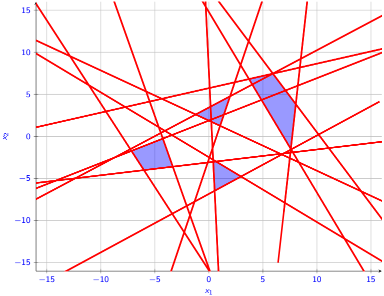


[0] Prodan I., Stoican F., Olaru S. and Niculescu S-I. (2016): Mixed-Integer Representations in Control Design, SpringerBriefs in Control, Automation and Robotics Series, Springer.

Non-connected and non-convex regions

Consider the complement $\mathcal{C}(\mathbb{S}) = cl(\mathbb{R}^n \setminus \mathbb{S})$ of a union of polyhedral sets $\mathbb{S} = \bigcup_l S_l$

$$
\begin{aligned}
&\vdots\\
A_l &\begin{cases} \sigma_l(1)h_1 x & \leq \sigma_l(1)k_1 + M\alpha_l(\lambda) \\ & \vdots \\ \sigma_l(N)h_N x & \leq \sigma_l(N)k_N + M\alpha_l(\lambda) \end{cases}\\
&\vdots\\
&0 \leq \beta_l(\lambda)
\end{aligned}
$$



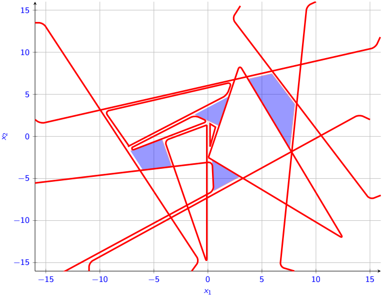
Using the hyperplanes $\mathcal{H}_i$ we partition the space into disjoint cells A_l and we associate a linear combination of binary variables $\alpha_l(\lambda)$ to each cell.

[0] Prodan I., Stoican F., Olaru S. and Niculescu S-I. (2016): Mixed-Integer Representations in Control Design, SpringerBriefs in Control, Automation and Robotics Series, Springer.

Non-connected and non-convex regions

Consider the complement $\mathcal{C}(\mathbb{S}) = cl(\mathbb{R}^n \setminus \mathbb{S})$ of a union of polyhedral sets $\mathbb{S} = \bigcup_l S_l$

$$
\begin{array}{c}
\vdots \\
A_l \begin{cases} \sigma_l(1) h_1 x & \leq \sigma_l(1) k_1 + M\alpha_l(\lambda) \\ & \vdots \\ \sigma_l(N) h_N x & \leq \sigma_l(N) k_N + M\alpha_l(\lambda) \end{cases} \\
\vdots \\
0 \leq \beta_l(\lambda)
\end{array}
$$



The number of cells can be reduced through merging procedures.

[0] Prodan I., Stoican F., Olaru S. and Niculescu S-I. (2016): Mixed-Integer Representations in Control Design, SpringerBriefs in Control, Automation and Robotics Series, Springer.

Obstacle avoidance problems

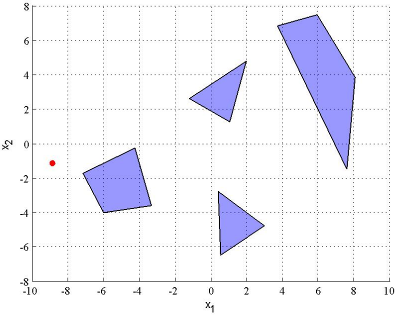
Consider a dynamical agent characterized by the LTI dynamics:

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k),$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $y(k) \in \mathbb{R}^p$ the agent state, input and output, respectively.

Collision avoidance condition:
For any obstacle S_l and an agent characterized by its dynamical state $x(k)$ we have:

$$\{x(k)\} \cap S_l = \emptyset, \quad \forall l = 1 \ldots N_o.$$



[0] (Stoican et al., ECC'13)

Obstacle avoidance problems

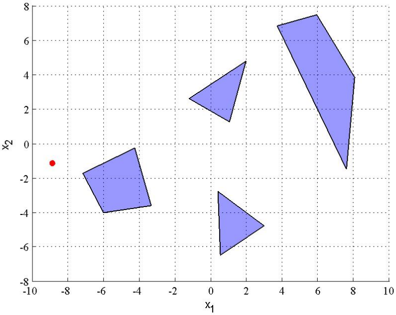
Consider a dynamical agent characterized by the LTI dynamics:

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k),$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $y(k) \in \mathbb{R}^p$ the agent state, input and output, respectively.

MIP representation of the feasible space:

- 14 hyperplanes
- 106 regions obtained with hyperplane arrangements
- 10 cells describing the interdicted regions
- 96 cells describing the feasible region
- $N_0 = 4$ the number of the binary variables



[0](Stoican et al., ECC'13)

Obstacle avoidance problems

Consider a dynamical agent characterized by the LTI dynamics:

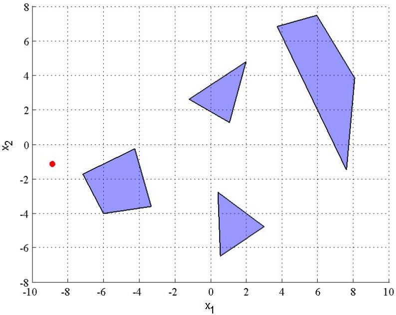
$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k),$$

with $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$ and $y(k) \in \mathbb{R}^p$ the agent state, input and output, respectively.

Solve the MIQP optimization problem over a finite prediction horizon:

$$u^* = \arg \min_{u(k),\dots u(k+N_p-1)} \sum_{i=0}^{N_p-1} \|x(k+i+1)\|_Q + \|u(k+i)\|_R,$$

$$\begin{aligned} \text{s.t. } & x(k+i+1) = Ax(k+i) + Bu(k+i), \\ & y(k+i) \in \mathcal{Y}, \; u(k+i) \in \mathcal{U}, \\ & x(k+i+1) \notin \mathbb{S}, \quad l = 1 \dots N_o. \end{aligned}$$



Conclusion: 72% complexity reduction of binary variables.

[0] (Stoican et al., ECC'13)

Obstacle and collision avoidance example

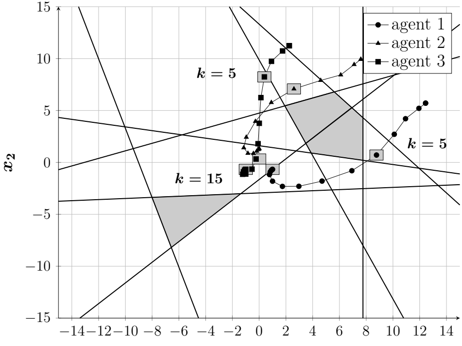
Collision avoidance conditions:

1. for any obstacle S_l and any agent characterized by its dynamical state $x_i(k)$ and the associated safety region S_i^a, the collision avoidance conditions are:

$$(\{x_i(k)\} \oplus S_i^a) \cap S_l = \emptyset, \quad \forall i = 1 \dots N_a, \ \forall l = 1 \dots N_o.$$

2. for any two agents characterized by their dynamical states $x_i(k)$, $x_j(k)$ and their associated safety regions S_i^a, S_j^a, the collision avoidance conditions are:

$$(\{x_i(k)\} \oplus S_i^a) \cap \left(\{x_j(k)\} \oplus S_j^a\right) = \emptyset, \quad \forall i,j = 1 \dots N_a, \ i \neq j.$$



Obstacle and collision avoidance example

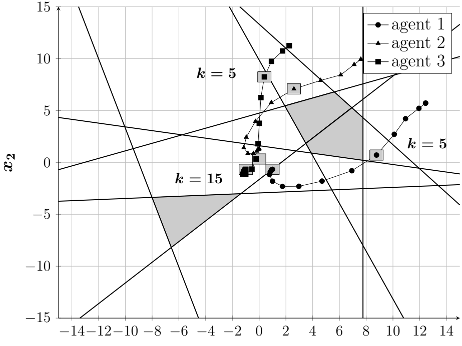
Collision avoidance conditions:

1. for any obstacle S_l and any agent characterized by its dynamical state $x_i(k)$ and the associated safety region S_i^a, the collision avoidance conditions are:

$$x_i(k) \notin (\{-S_i^a\} \oplus S_l), \quad \forall i = 1 \dots N_a, \ \forall l = 1 \dots N_o,$$

2. for any two agents characterized by their dynamical states $x_i(k)$, $x_j(k)$ and their associated safety regions S_i^a, S_j^a, the collision avoidance conditions are:

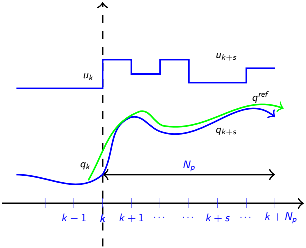
$$x_i(k) - x_j(k) \notin \left(\{-S_i^a\} \oplus S_j^a\right), \quad \forall i, j = 1 \dots N_a, \ i \neq j.$$



Model Predictive Control (MPC)

Propoi (1963), Cutler et al. (2007), Richalet and O'Donovan (2009)

- Optimization-based control law
- Implicit (on-line) vs. explicit (off-line) implementation
- Constraints handling
- Can be implemented in a distributed fashion



model of the system

reference trajectory

$$\arg\min_{\mathbf{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to: } \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p-1, \\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p-1. \end{cases}$$

constraint sets representation

optimization solver

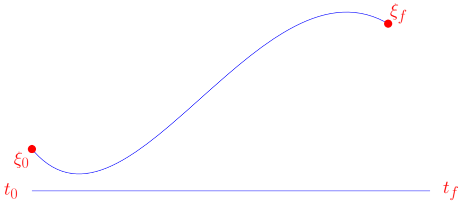
Reference trajectory generation

Consider the system

$$\dot{x}(t) = f(x(t), u(t)),$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$.

How we can generate a feasible reference trajectory $(\xi^{ref}(t), u^{ref}(t))$ that steers the model from an initial state $\xi^{ref}(t_0)$ to a final state $\xi^{ref}(t_f)$, over a fixed time interval $[t_0, t_f]$?



Reference trajectory generation – differential flatness

Consider the continuous nonlinear system

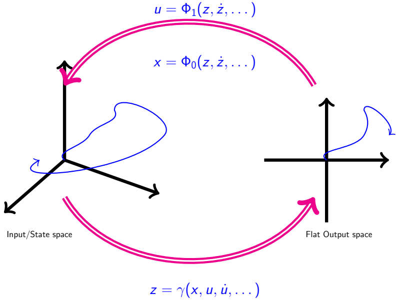
$$\dot{x}(t) = f(x(t), u(t)),$$

it is called differentially flat if there exist $z(t)$ s.t. the states and inputs can be algebraically expressed in terms of $z(t)$ and a finite number of its derivatives (Fliess et al. (1995)):

$$x(t) = \Phi_0(z(t), \dot{z}(t), \cdots, z^{(q)}(t)),$$
$$u(t) = \Phi_1(z(t), \dot{z}(t), \cdots, z^{(q+1)}(t)),$$

where

$$z(t) = \gamma(x(t), u(t), \dot{u}(t), \cdots, u^{(q)}(t)).$$



- For any linear and nonlinear flat system, the number of flat outputs equals the number of inputs Lévine (2009)
- For linear systems, the flat differentiability is implied by the controllability property Sira-Ramírez and Agrawal (2004)

Flat trajectory – Constructive details

Differentially flat systems are well suited to problems requiring trajectory planning $\Rightarrow$ it reduces the problem of trajectory generation to finding a trajectory of the flat outputs:

- assume an interval $t \in [0, T]$
- boundary conditions $x(0) = x_0, x(T) = x_f$, $u(0) = u_0, u(T) = u_f$
- choose a basis function $\Lambda(t) = [\ldots \Lambda^i(t) \ldots]$
- parametrize the flat output $z(t) = \sum_{i=1}^{N_\alpha} \alpha_i \Lambda^i(t)$
- and its derivatives $z^{(q)}(t) = \sum_{i=1}^{N_\alpha} \alpha_i \Lambda^{(q)}(t)$
- obtain coefficients α_i from the boundary conditions
- go back to $x(t)$ and $u(t)$

Constructive details – II

There are several issues:

- there are many basis functions but not all of them all well-suited
  - polynomials (t^i): poor numerical performance, their dimension depends on the number of conditions imposed on the inputs, states and their derivatives
  - Bésier basis functions: their dimension depends on the number of conditions imposed on the inputs, states and their derivatives
  - B-spline basis functions: their degree depends only up to which derivative is needed to ensure continuity
- state and input constraints are not enforced: we impose constraints at the boundaries (or even in intermediate points) but what happens “in-between” ?
- shortcomings which are not usually taken into account:
  - imposition of avoidably strict constraints;
  - the trajectory passing through two consecutive way-points intersects one (or more) obstacles.

Example for and UAV dynamics

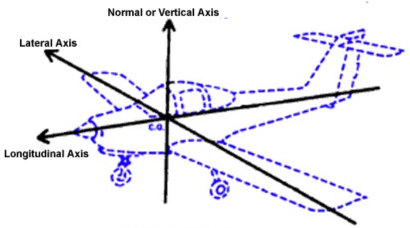
Bencatel et al. (2011)

Consider the 3-DOF kinematic model

$$\dot{x}(t) = V_a(t)\cos\Psi(t) + W_x$$
$$\dot{y}(t) = V_a(t)\sin\Psi(t) + W_y$$
$$\dot{\Psi}(t) = \frac{g\ \tan\Phi(t)}{V_a(t)}$$

described generally by

$$\dot{\xi}(t) = f(\xi(t), u(t))$$



The Axes of an Airplane

with $\xi(t) = \left[x^T(t)\ y^T(t)\ \Psi^T(t)\right]^T \in \mathbb{R}^3$ the state vector, $u = \left[V_a^T(t)\ \Phi^T(t)\right]^T \in \mathbb{R}^2$ the input vector and W_x, W_y the wind velocity components.

Specifications:

- air relative velocity $V_a \in [18, 25]$ m/s;
- bank angle $\Phi \in [-0.43, 0.43]$ rad;
- rate of change of V_a is limited to the maximum acceleration the aircraft can produce, i.e., $0.1 \sim 0.2$ m/s^2;
- variation of Φ is limited to $0.5 \sim 1.1$ rad/s.

Illustrative example

3-DOF model of an airplane in which the autopilot forces coordinated turns (zero side-slip) at a fixed altitude:

$$\dot{x}(t) = V_a(t)\cos\Psi(t),$$
$$\dot{y}(t) = V_a(t)\sin\Psi(t),$$
$$\dot{\Psi}(t) = \frac{g \ \tan\Phi(t)}{V_a(t)}$$

$$\Psi(t) = \arctan\left(\frac{\dot{z}_2(t)}{\dot{z}_1(t)}\right),$$
$$V_a(t) = \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)},$$
$$\Phi(t) = \arctan\left(\frac{1}{g}\frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}}\right).$$

with

- states are the position $(x(t), y(t))$ and the heading (yaw) angle $\Psi(t) \in [0, 2\pi]$ rad
- inputs signals are the airspeed velocity $V_a(t)$ and the bank (roll) angle $\Phi(t)$
- $z(t) = \begin{bmatrix} z_1(t) & z_2(t) \end{bmatrix}^T = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T$ is the flat output

Flat trajectory generation

Find a reference trajectory $(\xi^{ref}(t), u^{ref}(t))$ that steers the model from an initial state $\xi^{ref}(t_0)$ to a final state $\xi^{ref}(t_f)$, over a fixed time interval $[t_0, t_f]$:

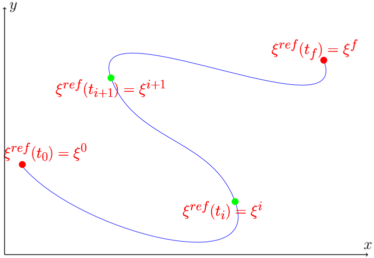
$$\begin{cases} \xi^{ref}(t) &= \eta_0(z(t), \dot{z}(t)), \\ u^{ref}(t) &= \eta_1(z(t), \dot{z}(t), \ddot{z}(t)), \end{cases} \quad \text{with } z(t) = [z_1(t)\ z_2(t)]^T \in \mathbb{R}^2 \text{ the flat output.}$$

The corresponding reference state and input for the system are:

$$\begin{cases} \xi^{ref}(t) &= \begin{bmatrix} z_1(t) & z_2(t) & \arctan\left(\frac{\dot{z}_2(t)}{\dot{z}_1(t)}\right) \end{bmatrix}^T, \\ u^{ref}(t) &= \begin{bmatrix} \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)} & \arctan\left(\frac{1}{g}\frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}}\right) \end{bmatrix}^T, \end{cases}$$

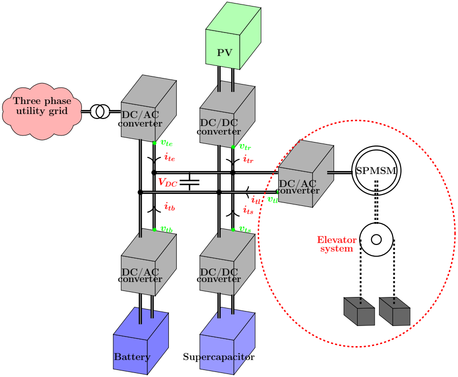
Introduce a set of way-points through which the vehicle must pass:

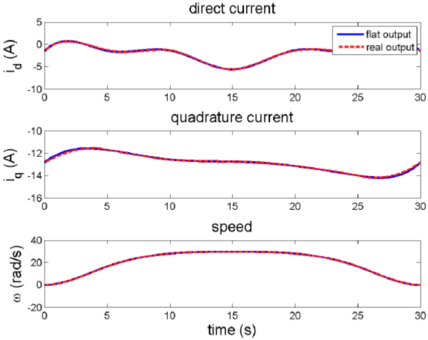
$$\mathbb{P} \triangleq \{p^i = (\xi^i, u^i), \quad i = 0 : N_w\}.$$



Example for the DC microgrid system

Generation of a speed profile for the elevator system Hung Pham et al. (2015):





[0] Pham, T., I. Prodan, D. Genon-Catalot et L. Lefèvre: *Efficient energy management for an elevator system under a constrained optimization framework*, in Proceedings of the 19th IEEE International Conference on System Theory, Control and Computing, pp. 613–618, 2015.

References I

Rubens JM Afonso and Roberto KH Galvão. Comments on “enhancements on the hyperplanes arrangements in mixed-integer programming techniques *Journal of Optimization Theory and Applications*, pages 1–8, 2013.

R. Bencatel, M. Faied, J. Sousa, and A.R. Girard. Formation control with collision avoidance. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 591–596, Orlando, Florida, USA, 12-15 December 2011.

C.R. Cutler et al. Method for removal of pid dynamics from mpc models, 2007. US Patent 7,263,473.

M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.

D. Grundel, R. Murphey, and Pardalos P.M. *Cooperative systems, Control and optimization*, volume 588. Springer Verlag, 2007.

T. Hung Pham, I. Prodan, D. Genon-Catalot, and L. Lefèvre. Efficient energy management for an elevator system under a constrained optimization framework. In *In Proceedings of the 19th IEEE International Conference on System Theory, Control and Computing*, pages –, 2015.

M. Jünger, M. Junger, T.M. Liebling, D. Naddef, G. Nemhauser, and W.R. Pulleyblank. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer Verlag, 2009.

J. Lévine. *Analysis and control of nonlinear systems: A flatness-based approach*. Springer Verlag, 2009.

I. Prodan, R. Bencatel, S. Olaru, J. Sousa, C. Stoica, and S.I. Niculescu. Predictive control for autonomous aerial vehicles trajectory tracking. In *In Proceedings of the IFAC Nonlinear Model Predictive Control Conference*, pages 508–513, Noordwijkerhout, The Netherlands, 23-27 August 2012a.

I. Prodan, F. Stoican, S. Olaru, and Niculescu S.I. Enhancements on the Hyperplanes Arrangements in Mixed-Integer Techniques. *Journal of Optimization Theory and Applications*, 154(2):549–572, 2012b. doi: 10.1007/s10957-012-0022-9.

I. Prodan, S. Olaru, R. Bencatel, J. Sousa, C. Stoica, and S.I. Niculescu. Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Control Engineering Practice*, 21(10):1334–1349, 2013. doi: 10.1016/j.conengprac.2013.05.010.

A.I. Propoi. Use of linear programming methods for synthesizing sampled-data automatic systems. *Automation and Remote Control*, 24(7):837–844, 1963.

J. Richalet and D. O’Donovan. *Predictive Functional Control: Principles and Industrial Applications*. Springer, 2009.

H. Sira-Ramírez and S. Agrawal. *Differential Flatness*. Marcel Dekker, New York, 2004.

F. Stoican, I. Prodan, and S. Olaru. Enhancements on the hyperplane arrangements in mixed integer techniques. In *In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3986–3991, Orlando, Florida, USA, 12-15 December 2011.

F. Stoican, I. Prodan, and S. Olaru. Hyperplane arrangements in mixed-integer programming techniques. collision avoidance application with zonotopic sets. In *In Proceedings of the IEEE European Control Conference*, pages 3155–3160, 2013.