Constrained Control of Multi-Vehicle Systems for Smart Cities and Industry 4.0: from Model Predictive Control to Reinforcement Learning

Part 1: Background material on MPC

Speaker: Dr. Walter Lucia

UNIVERSITÉ Concordia UNIVERSITY

2023 IEEE Intelligent Vehicle Symposium

Outline

1. What is MPC?
2. Dual-mode Set-Theoretic MPC

What is MPC?

What is MPC?

- MPC = Model Predictive Control

MPC is an advanced feedback control strategy that uses the dynamical model of the plant to predict its future evolution and choose the “best” current control action

- MPC schemes try to emulate how we naturally behave in everyday situations:
  - MPC is like how we play chess!
  - MPC is like how we decide what to wear before going out!
  - MPC is like how we drive our car!

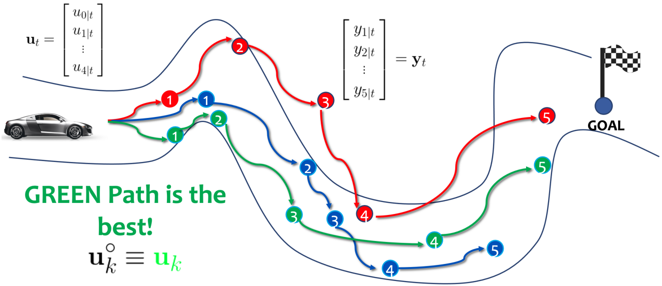
MPC and Weather Forecast

- MPC has preview capabilities (similar to feedforward control)

	Current weather situation	Future weather situation	
Decision based on the current weather situation			Without a forecast, one gets into difficulties
Decision based on the weather forecast			One can handle disturbances by planning in advance (with a forecast)

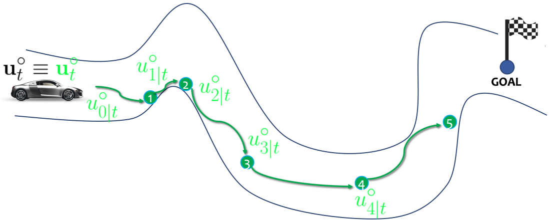
MPC and Driving a Car - I

- MPC can handle complex multi-input multi-output (MIMO) systems (e.g., a car)
- It can take into account constraints (e.g., road limits, acceleration constraints)
- It can minimize a given cost function (e.g., time to reach the destination)



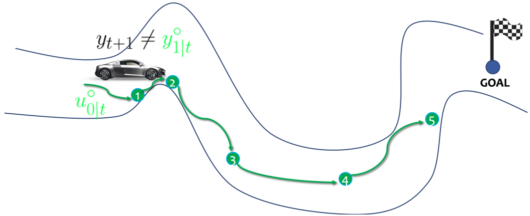
MPC and Driving a Car - II

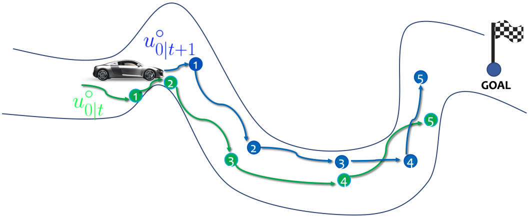
After we find u_t°, will we drive in an open-loop fashion?



- The answer is NO! After we move (i.e., after the first control move) we re-evaluate the new situation.
  - There actual position might be different from the predicted one (modeling errors, disturbances)
  - There might be time-varying constraints (e.g., another car along the path)

MPC and Driving a Car - III





MPC uses a Receding Horizon Control (RHC) approach!

MPC - Key Ingredients

- Prediction horizon N
- A cost function J

$$J(x,u) = \sum_{k=0}^{N-1} \|y_k - goal\|_2^2 + \rho\|u_k\|_2^2$$

- Prediction model

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= g(x_k) \end{aligned}$$

- Set of constraints

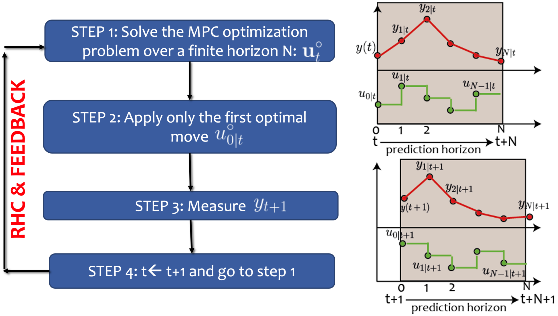
$$x_k \in \mathcal{X}, \quad y_k \in \mathcal{Y}, \quad u_k \in \mathcal{U}$$

Receding Horizon MPC - Algorithm

MPC optimization

$$
\begin{aligned}
\min_{u_0,\ldots,u_{N-1}} \quad & J(x,u) \quad s.t. \\
x_{k+1} &= f(x_k, u_k), \\
y_k &= g(x_k), \\
x_k &\in \mathcal{X}, \\
y_k &\in \mathcal{Y}, \\
u_k &\in \mathcal{U}, \\
x_0 &= x_t
\end{aligned}
$$

Predict-Optimize-Repeat



Different names used for MPC schemes

$$\min_{u_0,\dots,u_{N-1}} \quad J(x,u) \quad s.t.$$
$$x_{k+1} = f(x_k, u_k),$$
$$y_k = g(x_k),$$
$$x_k \in \mathcal{X},\; y_k \in \mathcal{Y},\; u_k \in \mathcal{U},$$
$$x_0 = x_t$$

- **Implicit vs Explicit MPC** $\leftarrow$ optimization solved online vs offline
- **Linear vs Nonlinear MPC** $\leftarrow$ prediction model linear vs nonlinear
- **Robust vs Stochastic MPC** $\leftarrow$ prediction model with bounded vs stochastic uncertainties
- **Hybrid MPC** $\leftarrow$ prediction model integrating logic and dynamics
- **Data-driven MPC** $\leftarrow$prediction model based on ML methods
- **Etc...**

MPC theoretical challenges and popular solutions

- Is stability guaranteed? Is recursive feasibility ensured?
  - Example: In general, if $N < \infty$ there is no guarantee that the problem will remain feasible, even for a simple linear nominal MPC.
- Having N large is beneficial ($N = \infty$ would be the best). However, this increases the size of the optimization problem
- With a small N, a popular way to ensure stability and recursive feasibility consists in adding to the MPC problem a terminal constraint

$$x_N \in \mathcal{X}_T$$

For a plant subject to bounded disturbances, X_N is typically a Robust Control Invariant (RCI), i.e.,

$$\forall x \in \mathcal{X}_N \rightarrow \exists u \in \mathcal{U} : Ax + Bu + d \in \mathcal{X}_N, \;\; \forall d \in \mathcal{D}$$

Dual-mode Set-Theoretic MPC

Set-Theoretic MPC

- For the rest of this tutorial, we will mainly use a particular MPC solution, known in the literature as dual-mode Set-Theoretic MPC (ST-MPC), see [*Angeli et al., Automatica 2008*]
- ST-MPC can be considered an MPC solution halfway between the explicit and the implicit implementations
- ST-MPC benefits:
  - It uses set-theoretic arguments [*Blanchini, Springer 2008*] (e.g., RCI sets and Robust one-step controllable sets) to ensure, by construction, stability and recursive feasibility
  - Most of the required computations are moved into an offline phase, leaving online a computationally affordable MPC problem ($N = 1$).

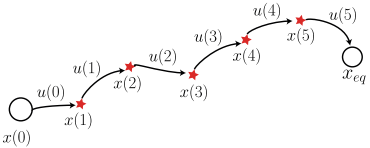
Dual-Mode Set-Theoretic Model Predictive Controller[1]

[1] See [*Angeli et al., Automatica 2008*] for more details

Reachability for Discrete-Time Linear Systems

(Unconstrained Plant)

$$x(k+1) = Ax(k) + Bu(k)$$



1. Any point of the state space can be reached in n steps if the system is reachable
2. If the system is reachable, then we can easily design a state-feedback control to drive the system $x(k)$ to any desired equilibrium pair (x_{eq}, u_{eq}), e.g., using the LQR controller

$$u(k) = -K(x(k) - x_{eq}) + u_{eq}$$

Reachability

(Constrained Plant with Disturbances)

$$x(k+1) = Ax(k) + Bu(k) + d(k)$$
$$u(k) \in \mathcal{U}, \quad x(k) \in \mathcal{X}, \quad d(k) \in \mathcal{D}$$

- In a constrained system, we have to be careful to use LQR
- It might not be possible to use a single state feedback controller satisfying all the constraints

$$u(k) = -K(x(k) - x_{eq}) + u_{eq}$$

(Intuition): If K is a fixed matrix, then the magnitude of $u(k)$ depends on how far $x(k)$ is from x_{eq}. For some $x(0)$, we could violate the constraints.

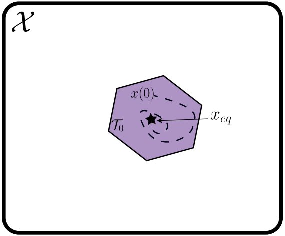
How can we ensure constraints fulfillment?

Robust Control Invariant Region (RCI)

- **Idea:** probably we can design a state-feedback controller that can be used in a small region $\mathcal{T}_0 \subset \mathcal{X}$ around the equilibrium point

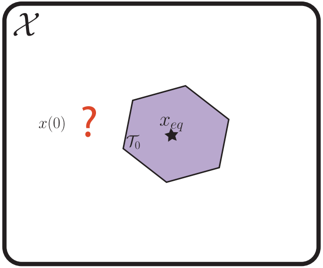
$$u(k) = -K(x(k) - x_{eq}) + u_{eq}, \quad \forall\, x(k) \in \mathcal{T}_0$$

- If K exists, and it also keeps the system state within $\mathcal{T}_0 \subset \mathcal{X}$, we call $\mathcal{T}_0$ a **R**obust **C**ontrol **I**nvariant (**RCI**) region under K.
- In the literature, there are different tools to build **RCI** regions for a given controller [*Blanchini, Springer 2008*].
- For ST-MPC the smallest $\mathcal{T}_0$ is of interest, see e.g., [*Rakovic et al., TAC 05*]



Controller Working Region: Drawback

$$u(k) = K(x(k) - x_{eq}) + u_{eq}, \quad \forall\, x(k) \in \mathcal{T}_0$$



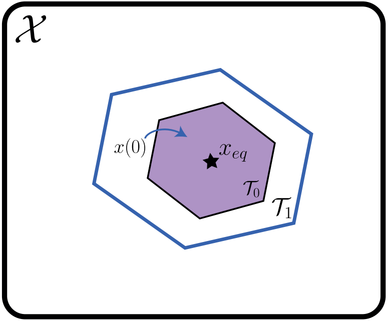
What can we do if the initial condition $x(0)$ is outside of controller's domain of attraction $\mathcal{T}_0$?[a]

[a] The domain of attraction is the region $\mathcal{T}_0$ where the controller works satisfying the constraints

Enlarging the Domain $\mathcal{T}_0$: one-step controllable sets

- We can enlarge the controller's working region by resorting to the concept of **R**obust **O**ne-**S**tep **C**ontrollable (ROSC) set.
- A set $\mathcal{T}_1 \subseteq \mathcal{X}$ is ROSC to $\mathcal{T}_0$ if the following is true:

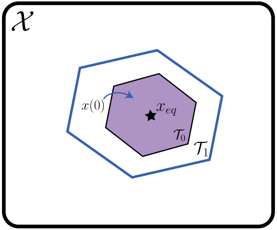
$$\forall\, x(0) \in \mathcal{T}_1 \rightarrow\ \exists\, u \in \mathcal{U} :\ Ax(0) + Bu + d \in \mathcal{T}_0,\ \forall\, d \in \mathcal{D}$$



Enlarging the Domain $\mathcal{T}_0$: one-step controllable sets

$$\mathcal{T}_1 := \{x(0) \in \mathcal{X} : \exists\, u \in \mathcal{U} : Ax(0) + Bu + d \in \mathcal{T}_0, \forall\, d \in \mathcal{D}\}$$

- For polyhedral sets: $\boxed{\mathcal{T}_1 = (\mathcal{X} \ominus \mathcal{W}) \oplus (-B\mathcal{U})A}$, see, e.g., [*Borrelli et al, CUP 17*]

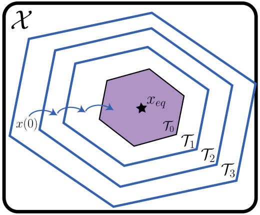


- Now, by construction, we can use 2 different controllers (dual-mode control):
  1. A controller to go from $\mathcal{T}_1 \to \mathcal{T}_0$
  2. A controller to stay in $\mathcal{T}_0$ and reach x_{eq} (if no disturbances): $u(k) = -K(x(k) - x_{eq}) + u_{eq}$

Family of One-Step Controllable Sets and Controller

- We can recursively apply the following definition to enlarge as much as we can the controller's domain

$$\mathcal{T}_i := \{x \in \mathcal{X} : \exists\, u \in \mathcal{U} : Ax + Bu + d \in \mathcal{T}_{i-1}, \forall\, d \in \mathcal{D}\}, \quad \forall\, i > 0$$



- Example: in the figure above, starting from $x(0) \in \mathcal{T}_3$, **by construction**, with at most 3 control moves we can reach the terminal region $\mathcal{T}_0$

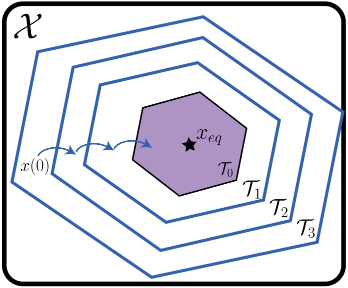
How do we actually compute the controller's actions $u(k)$ to ensure the transitions $\mathcal{T}_3 \rightarrow \mathcal{T}_0$?

Dual-Mode Set-theoretic Control (ST-C) Algorithm[2]

Offline Operations

1. Design $u(k) = -K(x(k) - x_{eq}) + u_{eq}$ and find the associated smallest **RCI** region $\mathcal{T}_0$
2. Build a family of ROSC sets $\{\mathcal{T}_i\}_{i=0}^{N}$ until the desired state-space region is covered[a]

[a] or until the set growth saturates



Online Operations

1. At each k, **Find** the minimum $i(k)$ such that $x(k) \in \mathcal{T}_{i(k)}$
2. **If** $i(k) > 0$ then

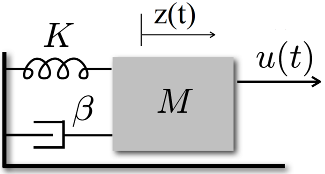
$$u(k) = \arg\min ||Ax(k) + Bu - x_{eq}||_2^2, \; s.t.$$
$$Ax(k) + Bu \in (\mathcal{T}_{i(k)-1} \ominus \mathcal{D}), \; u \in \mathcal{U}$$

Else $u(k) = -K(x(k) - x_{eq}) + u_{eq}$

[2] [*Angeli et al., Automatica 2008*]

Example - Using Matlab

- Mass-Spring-Damper system

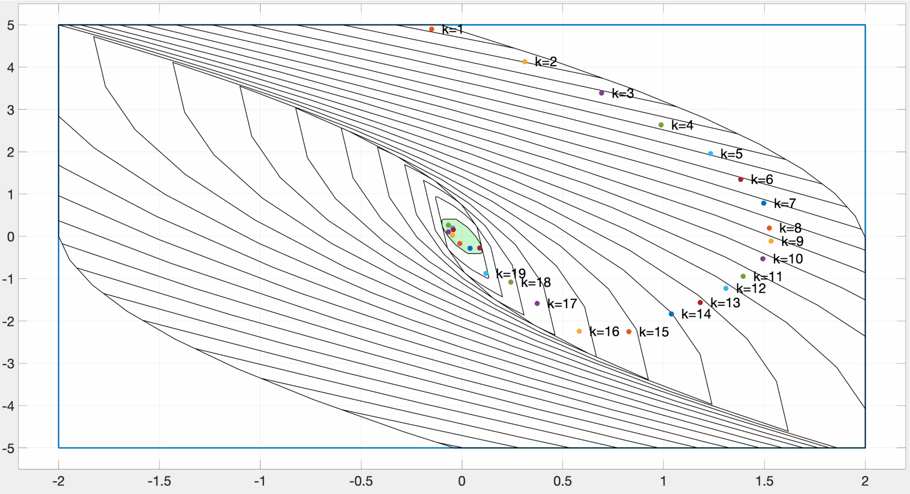


$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -\frac{\beta}{M} \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}}_{B} u(t)$$

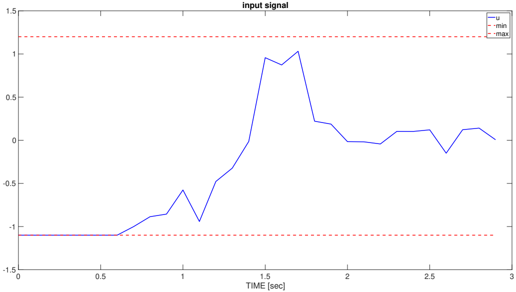
- Assuming $M = 25$, $K = 24$, and $\beta = 8$, design a ST-MPC controller to stabilize the system

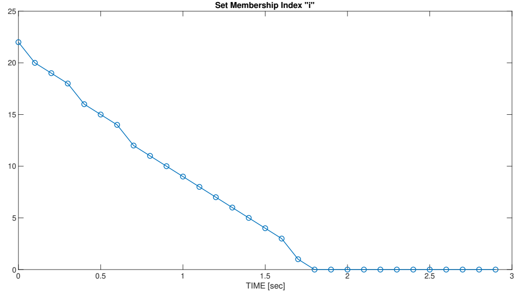
MATLAB implementation. Code available on github:
`https://github.com/PreCyseGroup/ST-MPC`

Demo Results (I)



Demo Results (II)





End of Part 1 - Questions ?

Thank You!

References I

- F. Blanchini, S. Miani
  Set-theoretic methods in control
  Springer, 2008.
- D. Angeli, A. Casavola, G. Franzè, E. Mosca
  An ellipsoidal off-line MPC scheme for uncertain polytopic discrete-time systems
  Automatica, Vol. 42, No. 12, pp. 3113-3119, 2008.
- S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, D. Q. Mayne
  Invariant approximations of the minimal robust positively invariant set.
  IEEE Transactions on Automatic Control, Vol. 50, No. 3, pp. 406-410, 2005.
- F. Borrelli, A. Bemporad, M. Morari,
  Predictive control for linear and hybrid systems.
  Cambridge University Press, 2017.