

LMPC Cookbook

November 2018

1 Preliminaries

The Learning Model Predictive Controller (LMPC) is a control framework designed for iterative tasks. At each j th “iteration” or “trial” of the control task, we store the closed-loop trajectory and the associated input sequence,

$$\begin{aligned}\mathbf{x}^j &= [x_0^j, \dots, x_{T^j}^j] \\ \mathbf{u}^j &= [u_0^j, \dots, u_{T^j-1}^j]\end{aligned}$$

where T^j is the time at which the task is completed.

The key idea is to use the stored data to compute a convex safe set and an approximation to the value function. The convex safe set is defined as the convex hull of the union of the stored data,

$$\mathcal{CS}^j = \text{Conv}\left(\bigcup_{i=0}^j \bigcup_{t=0}^{T^i} x_k^i\right) = \{x \in \mathbb{R}^n : \exists \lambda_k^i \geq 0, \sum_{i=0}^j \sum_{t=0}^{T^i} \lambda_k^i = 1, \sum_{i=0}^j \sum_{t=0}^{T^i} \lambda_k^i x_k^i = x\}. \quad (1)$$

Now, recall the definition of the cost-to-go associated with the stored state x_k^i ,

$$Q_k^j = h(x_k^j, u_k^j) + Q_{k+1}^j \quad (2)$$

for $Q_{T^j}^j = h(x_{T^j}^j, 0), \forall i \geq 0$. The above cost-to-go of the stored trajectories is used to define the Q -function

$$\begin{aligned}Q^{j,*}(x) &= \min_{\lambda_k^j \in [0,1]} \sum_{i=0}^j \sum_{k=0}^{T^i} Q_k^i \lambda_k^i \\ \text{s.t.} \quad & \sum_{i=0}^j \sum_{k=0}^{T^i} x_k^i \lambda_k^i = x, \quad \sum_{i=0}^j \sum_{k=0}^{T^i} \lambda_k^i = 1,\end{aligned} \quad (3)$$

which will be used as a terminal cost in the Finite Time Optimal Control Problem (FTOCP) solved by the LMPC.

2 LMPC Implementation

The safe set and function approximations described in the previous section are used to define the LMPC. At each time t of the j th iteration we solve the following finite time optimal control problem

$$J_{t \rightarrow t+N}^{\text{LMPC},j}(x_t^j) = \min_{\mathbf{U}_t^j, \boldsymbol{\lambda}_t^j} \left[\sum_{k=t}^{t+N-1} h(x_{k|t}^j, u_{k|t}^j) + \sum_{i=0}^{j-1} \sum_{k=0}^{T^i} Q_k^i \lambda_k^i \right] \quad (4a)$$

$$\text{s.t. } x_{t|t}^j = x_t^j, \quad (4b)$$

$$x_{k+1|t}^j = f(x_{k|t}^j, u_{k|t}^j), \quad \forall k = t, \dots, t+N-1 \quad (4c)$$

$$x_{k|t}^j \in \mathcal{X}, u_{k|t}^j \in \mathcal{U}, \quad \forall k = t, \dots, t+N-1 \quad (4d)$$

$$\lambda_k^i \geq 0, \sum_{i=0}^{j-1} \sum_{t=0}^{T^i} \lambda_k^i = 1, \sum_{i=0}^{j-1} \sum_{t=0}^{T^i} \lambda_k^i x_k^i = x_{t+N|t} \quad (4e)$$

where $\mathbf{U}_t^j = [u_{t|t}^j, \dots, u_{t+N-1|t}^j]$ is the vector of open loop sequences and the vector $\boldsymbol{\lambda}_t^j \in \mathbb{R}^{\Pi_i^{j-1} T^i}$ collects the multipliers associated with each recorded data. Equation (4a) represents the cost function which is the summation of the running cost $h(\cdot, \cdot)$ and the approximation to the value function $Q^{*,j-1}(\cdot)$ defined in (3). Equations (4b)-(4c) and (4d) represent the initial condition, dynamics constraint, state and input constraints, respectively. Finally, equation (4e) enforces the latest predicted state $x_{t+N|t}$ into the convex hull of the recorded states. Let

$$\mathbf{U}_t^{j,*} = [u_{t|t}^{j,*}, \dots, u_{t+N-1|t}^{j,*}]$$

be the optimal solution to (4) at time t of the j th iteration, we apply to the system the first element of the optimizer vector

$$u_t = u_{t|t}^{j,*}$$

The finite time optimal control problem (4) is repeated at time $t+1$, based on the new state $x_{t+1|t+1} = x_{t+1}^j$, until the iteration is completed.

3 Example

On my GitHub page (<https://github.com/urosolia/LMPC.SimpleExample>) I have uploaded a simple Matlab Code (based on YALMIP and MPT, installation steps: <https://www.mpt3.org/Main/Installation>), where the LMPC is used to solve the following infinite time constrained optimal control problem

$$\begin{aligned} J_{0 \rightarrow \infty}^*(x_S) &= \min_{u_0, u_1, \dots} \sum_{k=0}^{\infty} x_k^\top Q x_k + u_k^\top R u_k \\ \text{s.t. } \quad &x_{k+1} = A x_k + B u_k, \quad \forall k \geq 0 \\ &x_k \in \mathcal{X}, \quad u_k \in \mathcal{U}, \quad \forall k \geq 0 \\ &x_0 = x_S. \end{aligned}$$

List of files in the GitHub Repo:

- *Main.m*: This file first loads the system matrices and the data from a first feasible solution. Afterwards executes the LMPC controller (*LMPC.m*) and plots the results.
- *LMPC.m*: This file runs the LMPC controller for $\#Iterations$. At each time t of the j th iteration the FTOCP (4) (*FTOCP.m*) is solved. Furthermore, when j th iteration is completed the closed-loop data are used to build the safe set and approximation to the value function.
- *FTOCP.m*: This file solves the FTOCP (4) using the data from the previous iterations to compute the convex safe set and the cost-to-go associated with the recorded data, which is computed in *ComputeCost.m*
- *ComputeCost.m*: This file computes the cost-to-go associated with the recorded data.

4 References

This short guide illustrates how to implement the LMPC proposed in:

- Ugo Rosolia and Francesco Borrelli. "Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework." In IEEE Transactions on Automatic Control (2018).
- Ugo Rosolia and Francesco Borrelli. "Learning Model Predictive Control for Iterative Tasks: A Computationally Efficient Approach for Linear System." IFAC-PapersOnLine 50.1 (2017).