# 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 assume to 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 all of the union of the stored data,

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

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

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

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

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

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 defined the LMPC. At each time $t$ of the $j$th iteration we solve the following finite time optimal control problem

$$
\begin{aligned}
J^{\text{LMPC},j}_{t\to t+N}(x^j_t) = \min_{\mathbf{U}^j_t,\boldsymbol{\lambda}^j_t} \quad & \left[\sum_{k=t}^{t+N-1} h(x^j_{k|t}, u^j_{k|t}) + \sum_{i=0}^{j-1}\sum_{k=0}^{T^i} Q^i_k \lambda^i_k\right] && \text{(4a)}\\
\text{s.t.} \quad & x^j_{t|t} = x^j_t, && \text{(4b)}\\
& x^j_{k+1|t} = f(x^j_{k|t}, u^j_{k|t}), \qquad \forall k = t, \cdots, t+N-1 && \text{(4c)}\\
& x^j_{k|t} \in \mathcal{X}, u^j_{k|t} \in \mathcal{U}, \qquad \forall k = t, \cdots, t+N-1 && \text{(4d)}\\
& \lambda^i_k \geq 0, \sum_{i=0}^{j-1}\sum_{t=0}^{T^j} \lambda^i_k = 1, \sum_{i=0}^{j-1}\sum_{t=0}^{T^j} \lambda^i_k x^i_k = x_{t+N|t} && \text{(4e)}
\end{aligned}
$$

where $\mathbf{U}^j_t = [u^j_{t|t}, \dots, u^j_{t+N-1|t}]$ is the vector of open loop sequences and the vector $\boldsymbol{\lambda}^j_t \in \mathbb{R}^{\Pi^{j-1}_i T^i}$ collects the multiplies 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

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

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

$$u_t = u^{j,*}_{t|t}$$

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

## 3 Example

On my GitHub page () I have uploaded a simple Matlab Code (based on YALMIP https://yalmip.github.io/), where the LMPC is used to solve the following infinite time constrained optimal control problem

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

List of files in the GitHub Repo:

- *Main.m*: This files 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 files 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 files solves the FTOCP (4) using the data from the previous iterations to compute the convex safe set and the of the cost-to-go associated with the recorded data, which is compute in *ComputeCost.m*
- *ComputeCost.m*: This file computes the cost-to-go associated with the recorded data.

# 4 References

This short guide illustrated how to implemented 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).