1 Learning Model Predictive Control

Learning Model Predictive Control (LMPC) is a novel control theory that allows to find an optimal trajectory in an iterative task by learning from previous iterations. It is based on the concept of Iterative Learning Control (ILC). ILC is a control strategy that minimizes the tracking error of an iterative process by improving its trajectory with every iteration, learning from previous trajectories. It is limited to fixed time processes with given tracking references. [**?**] gives an overview on existing methods on ILC.
LMPC extends the framework of ILC by adding an optimization function which allows finding an optimal trajectory of optimal duration. Applying this strategy makes it possible to gradually find an optimal trajectory for complex nonlinear systems under only low computational cost.
Definition of iterative processes
There are two types of iterative processes that have to be distinguished: *Batch processes* and *Continuous repetitive processes* [**?**]. Batch processes are intermittently run and are usually modeled starting from the same initial state. As opposed to that, continuous repetitive processes transition directly from one iteration to the next one, meaning that the last state of an iteration matches the initial state of the next iteration.

Theory of LMPC Following theory has been adapted from [**?**].
First, we define the stage cost of a state $\boldsymbol{x}_t^j$ at time t in iteration j as follows:

$$h(\boldsymbol{x}_F, 0) = 0 \text{ and } h(\boldsymbol{x}_t^j, u_t^j) > 0 \;\forall\; \boldsymbol{x}_t^j \in \mathbb{R}^n \setminus \{\boldsymbol{x}_F\}, \boldsymbol{u}_t^j \in \mathbb{R}^m \setminus \{0\} \tag{1}$$

where $\boldsymbol{x}_F$ is the final state with $f(\boldsymbol{x}_F, 0) = 0$. Then, the cost of one iteration j is defined as the sum of its stage costs:

$$J^j = J^j_{0\to\infty}(\boldsymbol{x}_0^j) = \sum_{k=0}^{\infty} h(\boldsymbol{x}_k^j, \boldsymbol{u}_k^j). \tag{2}$$

Similarly, the cost-to-go of one specific state $\boldsymbol{x}_t^j$ of iteration j is defined as

$$J^j_{t\to\infty}(\boldsymbol{x}_t^j) = \sum_{k=t}^{\infty} h(\boldsymbol{x}_k^j, \boldsymbol{u}_k^j). \tag{3}$$

LMPC constructs a sampled safe set of feasible trajectories which is used as a terminal state constraint in the MPC formulation:

$$\mathcal{SS}^j = \left\{ \bigcup_{i\in M^j} \bigcup_{t=0}^{\infty} \boldsymbol{x}_t^i \right\} \tag{4}$$

with

$$M^j = \left\{ k \in [0, j] : \lim_{t\to\infty} \boldsymbol{x}_t^k = \boldsymbol{x}_F \right\}. \tag{5}$$

Additionally, a terminal cost function is defined over the sampled safe set:

$$Q^j(\boldsymbol{x}) = \begin{cases} \min\limits_{(i,t)\in F^j(\boldsymbol{x})} J^i_{t\to\infty}(\boldsymbol{x}), & \text{if } \boldsymbol{x} \in \mathcal{SS}^j \\ +\infty, & \text{if } \boldsymbol{x} \notin \mathcal{SS}^j \end{cases} \tag{6}$$

with

$$F^j(\boldsymbol{x}) = \left\{(i,t) : i \in [0,j], t \geq 0 \text{ with } \exists\ \boldsymbol{x}_t^i \in \mathcal{SS}^j \text{ and } \boldsymbol{x}_t^i = \boldsymbol{x}\right\}. \tag{7}$$

Using the terminal constraints and terminal cost we can write the finite time constrained optimal control problem:

$$J^{\text{LMPC},j}_{t\to t+N}(\boldsymbol{x}_t^j) = \min_{\boldsymbol{u}_{t|t},\ldots,\boldsymbol{u}_{t+N-1|t}} \left[\sum_{k=t}^{t+N-1} h(\boldsymbol{x}_{k|t}, u_{k|t}) + Q^{j-1}(\boldsymbol{x}_{t+N|t})\right] \tag{8a}$$

$$\text{s.t.} \tag{8b}$$

$$\boldsymbol{x}_{k+1|t} = f(\boldsymbol{x}_{k|t}, \boldsymbol{u}_{k|t}), \forall k \in [t,\ldots,t+N], \tag{8c}$$

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_t^j, \tag{8d}$$

$$\boldsymbol{x}_{k|t} \in \mathcal{X}, \boldsymbol{u}_k \in \mathcal{U}, \forall k \in [t,\ldots,t+N], \tag{8e}$$

$$\boldsymbol{x}_{t+N|t} \in \mathcal{SS}^{j-1}. \tag{8f}$$

It can be shown that this control strategy is recursively feasible and asymptotically stable. Additionally, the iteration cost J^j does not increase with the iteration index j and the trajectory converges to a local optimum for $j \to \infty$. The LMPC strategy presented above applies for batch processes, i.e. the initial state x_0^j has to be the same for each iteration. However, it is not applicable for continuous repetitive systems, for which the final state of an iteration corresponds to the initial state of the next iteration. The next section shows the extension for the repetitive case.

2 Repetitive Learning Model Predictive Control

This section presents a learning control strategy for continuous repetitive systems. We prove ... using the mathematical tools we introduced in section 1.

Not-so-mathematical description of the proving strategy: Use two consecutive laps, stacked together, for the safe set and cost function.
Start by constructing a safe set consisting of laps 0 and 1 in which initial and final states are equal (path following). Then start lap 2 and use this safe set and its Q function. At the end of lap 2, the controller can predict beyond the finish line, into the safe set of lap 1. After the finish line has been reached and the laps have been switched (from lap 2 to lap 3), we can use the stacked safe set of laps 0+1 and laps 1+2. However, the initial state is not necessarily the same as before.
This idea of stacking safe sets and Q functions is presented below.

Periodicity and switching condition First, we assume that the state dynamics and stage cost are periodic with periodicity vector $\boldsymbol{p}$:

$$f(\boldsymbol{x}_k + \boldsymbol{p}, \boldsymbol{u}_k) = f(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{p} \tag{9}$$

$$h(\boldsymbol{x} + \boldsymbol{p}, \boldsymbol{u}) = h(\boldsymbol{x}, \boldsymbol{u}). \tag{10}$$

If the dynamics are periodic in one state, the periodicity is expressed using a standard basis vector $\boldsymbol{p} = P\boldsymbol{e}_i$, where P is the periodicity in state i.
The switching from one iteration to the next one happens when the periodic state reaches P, i.e. $\boldsymbol{x}\boldsymbol{e}_i^T \geq P$.
Continuous repetitive systems are defined by smooth transitions between two iterations, i.e. the final state of one iteration becomes the initial state of the following iteration. This implicates that, at the end of one iteration, the control needs to take the beginning of the next iteration into account in order to find an optimal input.
We achieve this by creating a safe set that extends beyond the final state of one iteration to the end of the consecutive iteration.

Definitions First, we define the stacked sampled Safe Set $\tilde{\mathcal{SS}}^j$ at iteration j that extends beyond the finish line (*note:* it basically just contains stacks of two consecutive trajectories, e.g. laps 0+1, 1+2, 2+3, ...)

$$\tilde{\mathcal{SS}}^j = \left\{ \bigcup_{i \in M^j} \left[\left(\bigcup_{t=0}^{\infty} \boldsymbol{x}_t^i \right) \cup \left(\left(\bigcup_{t=0}^{\infty} \boldsymbol{x}_t^{i+1} \right) + \boldsymbol{p} \right) \right] \right\} \tag{11}$$

with

$$M^j = \left\{ k \in [0, j] : \lim_{t \to \infty} \boldsymbol{x}_t^{k+1} = \boldsymbol{x}_F \right\} \tag{12}$$

We can also write this definition of the stacked sampled safe set as follows:

$$\tilde{\mathcal{SS}}^j = \left\{ \mathcal{SS}^j \cup \left(\mathcal{SS}^{j+1} + \boldsymbol{p} \right) \right\} \tag{13}$$

Note that this definition of the stacked sampled safe set can also be written as a function of x:

$$\boldsymbol{x} \in \tilde{\mathcal{SS}}^j \rightarrow \begin{cases} \boldsymbol{x} \in \mathcal{SS}^j & \text{if } \boldsymbol{x} < P, \\ \boldsymbol{x} \in \mathcal{SS}^{j+1} & \text{if } \boldsymbol{x} \geq P \end{cases} \tag{14}$$

Similarly to the single iteration cost-to-go from eq. 3, we define the *stacked cost-to-go* $\tilde{J}^j_{t \to \infty}(\boldsymbol{x}_t^j)$ of iteration j as follows:

$$\tilde{J}^j_{t \to \infty}(\boldsymbol{x}_t^j) = \begin{cases} J^j_{t \to \infty}(\boldsymbol{x}_t^j) + J^{j+1} & \text{if } \boldsymbol{x} < P, \\ J^{j+1}_{t \to \infty}(\boldsymbol{x}_t^{j+1}) & \text{if } \boldsymbol{x} \geq P \end{cases} \tag{15}$$

Note: the stacked cost-to-go is a piecewise defined function, it is continuous through the lap switching condition (at P) and it is zero at $2P$. It is illustrated in fig. 1.

This stacking extends the cost-to-go beyond the finish line and allows at the end of one lap to consider the beginning of the next lap (for which the predicted $x_{t+N|t} > P$).
Using this definition of the stacked cost-to-go we define the stacked Q-function:

$$\tilde{Q}^j(\boldsymbol{x}) = \begin{cases} \min\limits_{(i,t) \in F^j(\boldsymbol{x})} \tilde{J}^i_{t \to \infty}(\boldsymbol{x}), & \text{if } x \in \tilde{\mathcal{SS}}^j, \\ +\infty & \text{if } x \notin \tilde{\mathcal{SS}}^j. \end{cases} \tag{16}$$

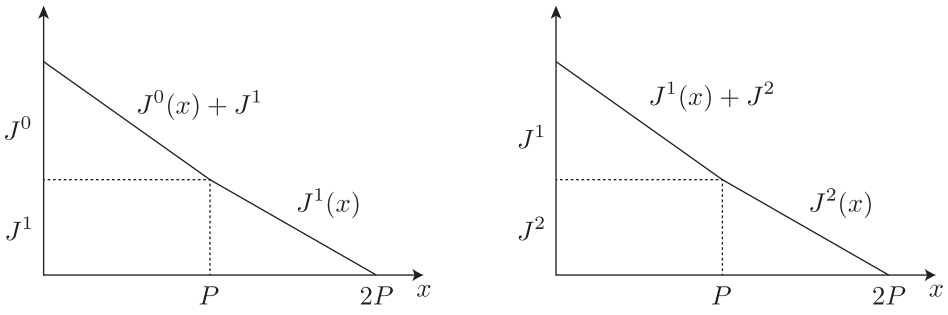

Figure 1: Stacked cost to go $\tilde{J}^0$ and $\tilde{J}^1$

We can also write the stacked Q function depending on x:

$$\tilde{Q}^j(x) = \begin{cases} \min\limits_{i,t} \left[J^i_{t\to\infty}(x) + J^{i+1} \right], & \text{if } x < P, \\ \min\limits_{i,t} \left[J^{i+1}_{t\to\infty}(x) \right], & \text{if } x > P. \end{cases} \tag{17}$$

With these definitions, we redefine the LMPC function which yields the repetitive LMPC function:

$$\tilde{J}^{\text{LMPC},j}_{t\to t+N}(\boldsymbol{x}^j_t) = \min_{\boldsymbol{u}_{t|t},\ldots,\boldsymbol{u}_{t+N-1|t}} \left[\sum_{k=t}^{t+N-1} h(\boldsymbol{x}_{k|t}, u_{k|t}) + \tilde{Q}^{j-2}(\boldsymbol{x}_{t+N|t}) \right] \tag{18a}$$

$$\text{s.t.} \tag{18b}$$

$$\boldsymbol{x}_{k+1|t} = f(\boldsymbol{x}_{k|t}, \boldsymbol{u}_{k|t}), \forall k \in [t, ..., t+N], \tag{18c}$$

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}^j_t, \tag{18d}$$

$$\boldsymbol{x}_{k|t} \in \mathcal{X}, \boldsymbol{u}_k \in \mathcal{U}, \forall k \in [t, ..., t+N], \tag{18e}$$

$$\boldsymbol{x}_{t+N|t} \in \tilde{\mathcal{S}}\tilde{\mathcal{S}}^{j-2}. \tag{18f}$$

Note: Since we use stacks of 2 consecutive laps for the safe set and Q function, we can't use the previous safe set $j-1$ but instead the one even further back, $j-2$. Our stacked safe set $\tilde{\mathcal{S}}\tilde{\mathcal{S}}^0$ also has to contain two successful laps before the LMPC starts!

Recursive feasibility Recursive feasibility throughout one iteration is guaranteed as usual. However, we have to prove feasibility even through the switching between two laps.
Consider state $\boldsymbol{x}^j_t$ in iteration j. Then the terminal constraint enforces $\boldsymbol{x}^{*,j}_{t+N|t} \in \tilde{\mathcal{S}}\tilde{\mathcal{S}}^{j-2}$. Assume that $\boldsymbol{x}^j_t$ is the last state of iteration j, meaning that $\boldsymbol{x}^{*,j}_{t+1|t} \geq P$ and therefore $\boldsymbol{x}^{*,j}_{t+N|t} \geq P$.
Using the definition of the stacked safe set in eq. 14 leads to $\boldsymbol{x}^{*,j}_{t+N|t} \in \mathcal{S}\mathcal{S}^{j-1}$. As before, the state update and prediction are assumed identical, leading to

$$\boldsymbol{x}^{*,j}_{t+1|t} = \boldsymbol{x}^j_{t+1} = \boldsymbol{x}^{j+1}_0 \tag{19}$$

In iteration $j+1$ we use the the stacked safe set $\tilde{\mathcal{SS}}^{j-1}$ with $\boldsymbol{x} < P$, leading to $\boldsymbol{x} \in \mathcal{SS}^{j-1}$. This means that the state trajectory

$$\left(x^{*,j}_{t+1|t}, \quad x^{*,j}_{t+2|t}, \quad ..., \quad x^{*,j}_{t+N-1|t}, \quad x^{i^*}_{t^*|t}, \quad x^{i^*}_{t^*+1|t}\right) = \tag{20}$$

$$\left(x^{*,j+1}_{0|t}, \quad x^{*,j+1}_{1|t}, \quad ..., \quad x^{*,j+1}_{N-2|t}, \quad x^{i^*}_{t^*|t}, \quad x^{i^*}_{t^*+1|t}\right) \tag{21}$$

is still feasible in iteration $j+1$.

Asymptotic stability The proof for asymptotic stability within an iteration works as in the standard LMPC case. As usual, we can show that $J^{\text{LMPC},j}_{0\to N}(x^j_t)$ is a decreasing Lyapunov function in lap j and we can show that $J^{\text{LMPC},j+1}_{0\to N}(x^j_t)$ is a decreasing Lyapunov function in lap $j+1$. However, we are interested in the stability when the lap switching happens. Looking at the LMPC cost at the state at the switching condition shows an incontinuity.
Consider state $\boldsymbol{x}^j_t = \boldsymbol{x}^{j+1}_0$ as the last state of iteration j and the first state of iteration $j+1$.

$$J^{\text{LMPC},j}_{0\to N}(x^j_t) = \min_u \left[\sum_{k=0}^{N-1} h(x_{k|t}, u_{k|t}) + \tilde{Q}^{j-2}(x_{N|t})\right] \qquad \text{and } x_{N|t} > P \tag{22}$$

$$= \min_u \left[\sum_{k=0}^{N-1} h(x_{k|t}, u_{k|t}) + Q^{j-1}(x_{N|t})\right] \tag{23}$$

$$= \min_u \left[\sum_{k=0}^{N-1} h(x_{k|t}, u_{k|t}) + \min_{i,t}\left[J^i_{t\to\infty}(x)\right]\right] \qquad \text{and } i \in [0, j-1] \tag{24}$$

$$J^{\text{LMPC},j+1}_{0\to N}(x^{j+1}_0) = \min_u \left[\sum_{k=0}^{N-1} h(x_{k|0}, u_{k|0}) + \tilde{Q}^{j-1}(x_{N|0})\right] \qquad \text{and } x_{N|0} < P \tag{25}$$

$$= \min_u \left[\sum_{k=0}^{N-1} h(x_{k|0}, u_{k|0}) + \min_{i,t}\left[J^i_{t\to\infty}(x) + J^{i+1}\right]\right] \qquad \text{and } i \in [0, j-1] \tag{26}$$

and therefore

$$J^{\text{LMPC},j+1}_{0\to N}(x^{j+1}_0) \geq J^{\text{LMPC},j}_{0\to N}(x^j_t). \tag{27}$$

This result essentially just shows that the LMPC cost at the initial state of lap $j+1$ is larger or equal to the LMPC cost at the same state, but evaluated in the previous lap j.
Question: Can we still assume stability through iterations since the cost function is stable all the way *to* the switching condition and it is stable starting *from* the switching condition?

Non decreasing iteration cost Using the strategy above, we will have to focus on the transition between two laps. The "problem" is that - at the beginning of an iteration, right after the transition - the controller doesn't even

notice that laps were switched. The safe set is the same and the Q function only increased by a constant value.