Asynchronous NMPC with Variable Update Time (ASAP-MPC)

Dries Dirckx, Mathias Bos, Bastiaan Vandewal, Wilm Decré, Jan Swevers
MECO Research Team, Department of Mechanical Engineering, KU Leuven
Flanders Make@KU Leuven, Leuven, Belgium
Email: dries.dirckx@kuleuven.be

1 Introduction & Problem Setting

Nonlinear Model Predictive Control (NMPC) for trajectory planning is a topic that currently receives a widespread interest from the (robotics) research community. Classical NMPC schemes are supposed to stabilise a nonlinear system, reject disturbances to it and lead to agile control of that system which requires high NMPC update rates. The control inputs are computed by solving an Optimal Control Problem (OCP), minimising an objective e.g. time of the total trajectory or distance to be travelled to reach a target, subject to constraints such as collision avoidance, a model of the system dynamics, actuator limits, etc.

For nonlinear systems in complex environments or with limited onboard computation power, e.g. drones or autonomous vehicles, reaching high update rates and guaranteeing feasibility is often intractable. A widely used approach to reduce computational complexity is the Real-Time Iterations (RTI) technique [1]. RTI applies only the first iteration of an optimization solver to the system. It does not guarantee constraint satisfaction and can lead to collisions and infeasible controls that do not satisfy the modelled nonlinear system dynamics.

We propose an alternative NMPC strategy - Asynchronous NMPC (ASAP-MPC) - that deals with finite computation times that can extend over multiple update intervals but guarantees feasibility with respect to the constraints by solving the OCP to convergence. A low-level stiff feedback controller is added to guarantee that the actual state $x(t)$ tracks the computed NMPC solution such that $x(t) \approx \hat{x}(t)$. The expected finite computation time is maximally m samples.

2 Asynchronous NMPC

Figure 1 shows the principle for a one-dimensional example. Suppose at time t_i, a new solution A is available. Since a finite computation delay is expected, based on solution A and the current on-trajectory state $\tilde{x}(t_i)$, an estimation of the future on-trajectory state $\hat{x}(t_{i+m})$ is made. The next solution B is constrained to start from this future point. Awaiting solution B, solution A (red) is tracked up to m samples in the future. Once computed, solution B is stitched to A at $\hat{x}(t_{i+m})$.

If solution B arrives earlier at $t_j < t_{i+m}$, solution A is still tracked up to t_{i+m} after which the newly computed solution B is tracked up to t_{j+m}. Suppose the next solution (C) arrives just-in-time after m samples at $t_k = t_{j+m}$, solution C is stitched to solution B and immediately tracked. The future trajectory from solution B is discarded. Repeating this update procedure leads to a continuous trajectory in time for $x(t)$, constructed by asynchronously stitching new solutions to each other.

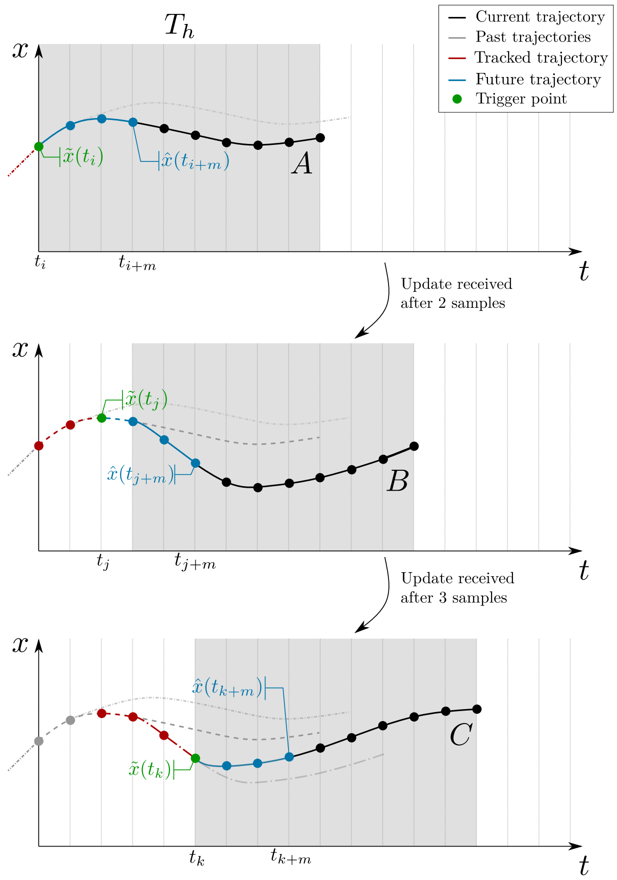

Figure 1: Working principle of the update strategy in ASAP-MPC, illustrated for a one-dimensional example.

References

[1] M. Diehl, R. Findeisen, F. Allgöwer "A Stabilizing Real-Time Implementation of Nonlinear Model Predictive Control," Real-Time PDE-Constrained Optimization, 2007.

Acknowledgement This research is supported by Flanders Make ICON FROGS: Flexible and Robust Robotic Gripping Solutions. Flanders Make is the Flemish strategic research centre for the manufacturing industry.