AMR-MPC: Sampled–data Model Predictive Control Using Adaptive Time–mesh Refinement, with Stability Guarantees

NMPC'18 – 6th IFAC Conference on Nonlinear Model Predictive Control | 19–22 August, 2018 | Madison, Wisconsin (USA)

Luís Tiago Paiva *,† and **Fernando A.C.C. Fontes** *,‡

* SYSTEC–ISR, Faculdade de Engenharia, Universidade do Porto, Portugal

† ltpaiva@fe.up.pt , ‡ faf@fe.up.pt

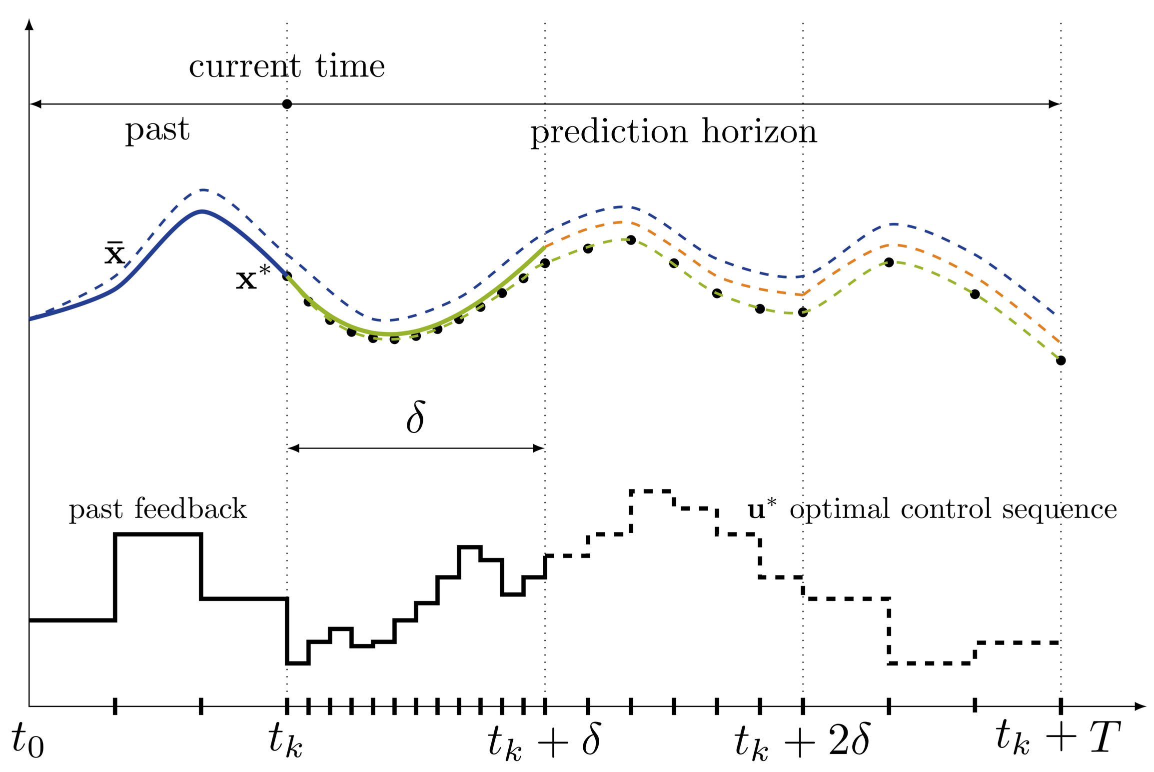
Main Ideia

What's you plan for tomorrow?
What's you plan for September 20?

Do you plan for tomorrow and for the next month with different levels of detail?

Why not do it with MPC?

- **Question**
  - Why should we use a fine mesh to compute the solution in $[t_k, t_k+T]$ if we only implement the control in $[t_k, t_k+\delta]$?
- **Idea**
  - To use a mesh that is finer in the left–end and coarser in the right–end of $[t_k, t_k+T]$.



Proposed AMR - MPC strategy

Abstract

- We address through MPC constrained nonlinear plant described by a continuous-time dynamical model: Sampled–data MPC.
- The numerical solution of the optimal control problems (OCP) involved must use, eventually, some form of discretization. However, there are several advantages in maintaining a continuous-time model until later stages.
- One advantage is that we can devise numerical procedures which, by exploiting additional freedom in selecting the discretization points, are more efficient when continuous-time models are used.
- In the numerical solution of nonlinear OCPs, the number of discretization nodes is a major factor affecting the computational time. Also, the location of such nodes is a major factor affecting the solution accuracy.
- The adaptive time–mesh refinement (AMR) algorithm iteratively finds an adequate time–mesh (selecting the number and location of the mesh–nodes) that satisfies a pre–defined bound on the error estimate of the obtained trajectories.
- Here, we discuss an extension to MPC of an AMR algorithm, which has shown to be efficient in solving nonlinear optimal control problems.
- We show how to guarantee that an MPC scheme using an AMR algorithm preserves stability.

We acknowledge the support of FEDER/COMPETE2020/NORTE2020/POCI/PIDDAC/MCTES/FCT funds through grants SFRH/BPD/126683/2016, UID/IEEA/00147/006933–SYSTEC, PTDC-EEI-AUT-2933-2014|-16858–TOCCATA, and POCI-01-0145-FEDER-031447|FCT–UPWIND.

Optimal Control and AMR

The OCP starting from an initial state $\mathbf{x}_k \in \mathbb{X}_0$:

$\mathscr{P}(\mathbf{x}_k)$:

$$\text{Minimize} \int_0^T \mathrm{L}(\mathbf{x}(s), \mathbf{u}(s))\,ds + \mathrm{G}(\mathbf{x}(T)) \tag{1a}$$

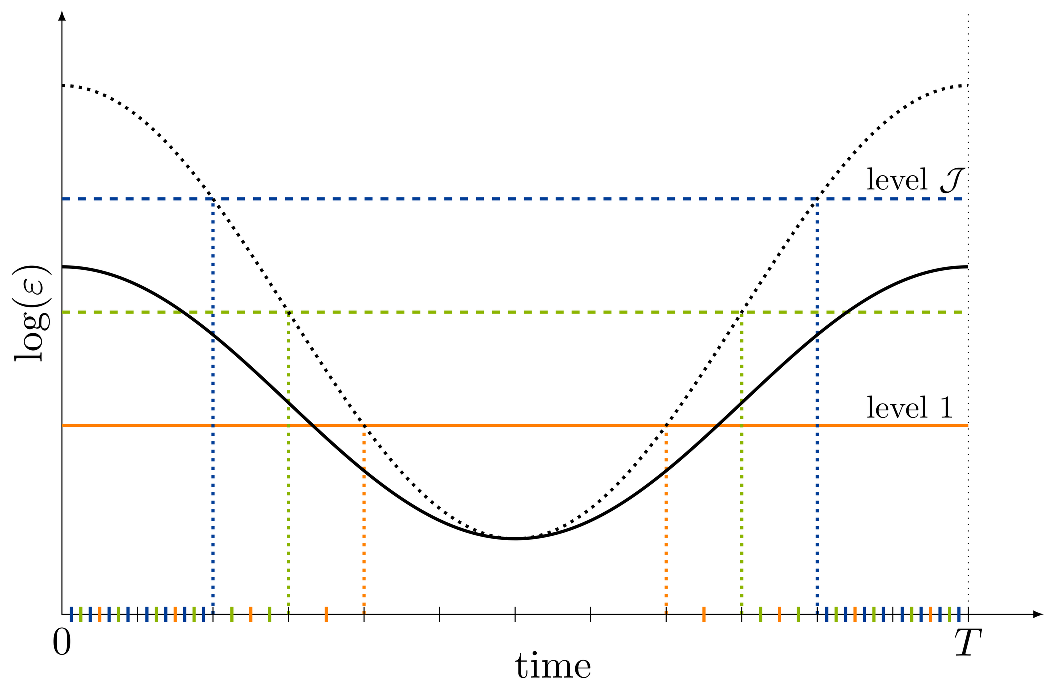
subject to (1b)

$$\dot{\mathbf{x}}(s) = \mathbf{f}(\mathbf{x}(s), \mathbf{u}(s)) \quad \text{a.e. } s \in [t_0, t_f], \tag{1c}$$
$$\mathbf{x}(0) = \mathbf{x}_k, \tag{1d}$$
$$\mathbf{x}(T) \in X_f, \tag{1e}$$
$$\mathbf{x}(s) \in \mathbb{X} \quad \forall s \in [t_0, t_f], \tag{1f}$$
$$\mathbf{u}(s) \in \mathbb{U} \quad \text{a.e. } s \in [t_0, t_f]. \tag{1g}$$

- We consider an initial optimization mesh $\pi_0 = \{s_i\}_{i=0\ldots N}$ in $[t_0, t_f]$ where the control functions can change value, containing all sampling instants in $[t_0, t_f]$.
- We solve the OCP with piecewise constant control, using direct methods. The model is initially discretized in π_0, transcribed into an NLP, and solved with standard solvers.
- The efficiency and the accuracy of the numerical solution strongly depends on the chosen time–mesh. The selection of an adequate mesh is not known a priori; it is generated via an iterative procedure: the AMR algorithm.

The AMR algorithm

- The adaptive mesh refinement procedure starts with a coarse mesh used to solve the NLP problem associated to the OCP to apprehend the main structure of the solution.
- Then, it adds new node points in the needed subintervals. This procedure adds more node points to the intervals in higher levels of refinement and it adds less node points to those in lower refinement levels.
- The refinement process is repeated until a certain stopping criterion is achieved.



The multi–level AMR technique.

The Extended AMR Algorithm

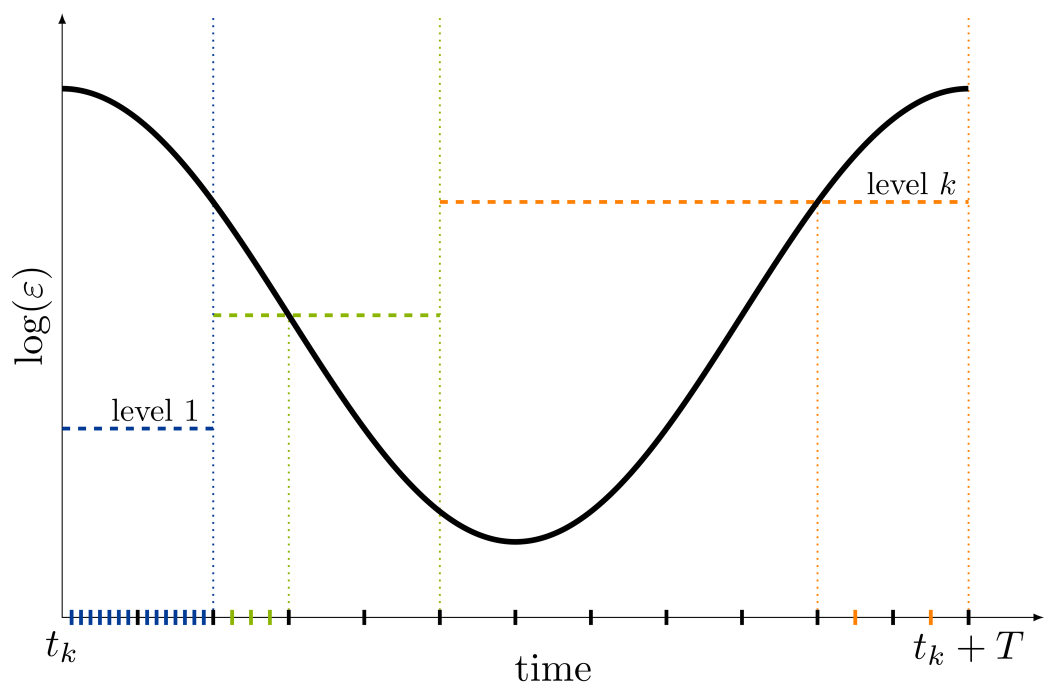
The AMR algorithm is extended to allow time–dependent refinement levels $\varepsilon^{\max}(s)$.

$$\varepsilon^{\max}(s) = \begin{cases} \varepsilon_1, & s \in [t_k, t_k + \beta_1 T] \\ \varepsilon_2, & s \in]t_k + \beta_1 T, t_k + \beta_2 T] \\ \ldots & \\ \varepsilon_j, & s \in]t_k + \beta_{\mathscr{J}-1} T, t_k + \beta_{\mathscr{J}} T] \end{cases}$$

where $0 < \beta_1 < \beta_2 < \ldots < \beta_{\mathscr{J}} < 1$ are user–defined scalars.

It is expected the procedure to add more node points in intervals that contain time instants close to the initial time of $[t_k, t_k+T]$.

However, the user does not specify directly the number of nodes to use in each interval, rather it specifies a threshold for the local absolute error on the trajectory in each interval.



The extended AMR strategy

MPC and AMR

Given a sampling step $\delta > 0$, a prediction horizon T, and a sequence of sampling instants $\{t_k\}_{k\geq 0}$ with $t_{k+1} = t_k + \delta$,

The sampled–data MPC algorithm follows the procedure:

1. Measure state of the plant $\mathbf{x}_{t_k}$;
2. Determine $\bar{\mathbf{u}} : [t_k, t_k+T] \to \mathbb{R}^m$ solution of OCP $\mathscr{P}(\mathbf{x}_{t_k})$.
3. Apply the control $\mathbf{u}^*(t) := \bar{\mathbf{u}}(t)$ to the plant in the interval $t \in [t_k, t_k+\delta]$, disregarding the remaining control $\bar{\mathbf{u}}(t), t > t_k+\delta$;
4. Repeat this procedure for the next sampling time instant $t_k+\delta$.

In the AMR - MPC algorithm, step 2 is modified to:

2. (a) Select the intervals $S_{k,j}$ to be refined according to the time–dependent levels of refinement $\bar{\varepsilon}(t)$ and generate a new time–mesh;
   (b) Determine $\bar{\mathbf{u}} : [t_k, t_k+T] \to \mathbb{R}^m$ solution to the OCP $\mathscr{P}(\mathbf{x}_k)$, in the new time–mesh;

Stability of AMR–MPC

The AMR-MPC startegy preserves stability, if the *design parameters* are selected to satisfy:

SC Sufficient condition for Stability. The design parameters T, L, G and X_f satisfy:

1. The set X_f is a subset of $\mathbb{X}$, is closed, and contains the origin. The function G is Lipschitz continuous and positive definite. The function L is continuous and there exists a function $M : \mathbb{R}^n \to \mathbb{R}_+$ which is continuous, positive definite and radially unbounded, such that $L(\mathbf{x}, \mathbf{u}) \geq M(\mathbf{x})$ for all $\mathbf{u} \in U$.
2. The horizon T is such that $\mathbb{X}_0$ is contained in $\mathscr{A}_0$, when controls from $\mathscr{U}(\pi_0)$ are used.
3. There exists a control law $k_f : [0, \delta] \times X_f \to \mathbb{R}^m$, with $k_f \in \mathscr{U}(\pi_0)$, such that for all $\mathbf{x}_f \in X_f$,

$$G(\mathbf{x}(\delta; \mathbf{x}_f, k_f)) - G(\mathbf{x}_f) \leq - \int_0^{\delta} L(\mathbf{x}(t; \mathbf{x}_f, k_f), k_f)\,dt, \tag{SCa}$$

$$\mathbf{x}(\delta; \mathbf{x}_f, k_f) \in X_f, \qquad \mathbf{x}(t; \mathbf{x}_f, k_f) \in X, \quad \text{all } t \in [0, \delta]. \tag{SCb}$$

and

$$k_f(t, \mathbf{x}_f) \in U, \quad \text{a.e. } t \in [0, \delta], \tag{SCc}$$

The Main Result

H1 Sets $\mathbb{X}$, $\mathbb{X}_0$ and $\mathbb{U}$ are compact, contain the origin and $\mathbf{f}(0,0) = 0$.

H2 The system is asymptotically controllable to the origin on $\mathbb{X}_0$.

H3 Function $\mathbf{f}$ is continuous, and $\mathbf{x} \mapsto \mathbf{f}(\mathbf{x}, \mathbf{u})$ is Lipschitz.

Theorem Assume the system satisfies hypotheses H1–H3. If the design parameters T, L, G and X_f satisfy the stability condition SC, then applying the AMR–MPC strategy starting from any $\mathbf{x}_0 \in \mathbb{X}_0$ and with some initial mesh $\pi_0 \supset (\Pi \cap [t_0, t_f])$ we have:

1. all optimal control problems involved in the AMR-MPC strategy, $\mathscr{P}(\mathbf{x}_k)$ for all $k \geq 0$, are feasible and have a minimum.
2. the closed-loop trajectory $\mathbf{x}^*$ is asymptotically attracted to the origin, that is $\mathbf{x}^*(t) \to 0$ as $t \to +\infty$.

Concluding remarks

Main advantages/features of AMR-MPC

- Obtains faster and/or more accurate solutions to the OCPs than with equidistant-spaced meshes (e.g. when discrete-time models are used).
- Can use continuous-time models of the plant directly. Discretization is automated and there is no need to choose a priori the discretization time step.
- Even if the optimization procedure is interrupted at an early stage (in real–time optimization) a solution (which might be less accurate) might still be provided.

References

[1] Fernando A.C.C. Fontes. (2001) *A general framework to design stabilizing nonlinear model predictive controllers*, Systems and Control Letters, 42(2), pp. 127–143.

[2] Luís Tiago Paiva, Fernando A.C.C. Fontes. (2015). *Adaptive Time–Mesh Refinement in Optimal Control Problems With State Constraints*, Discrete and Continuous Dynamical Systems, 35 (9), pp. 4553–4572.