

# Stability Analysis of Hypersampled Model Predictive Control

Yaashia Gautam, Marco M. Nicotra

**Abstract**—This paper introduces a new framework for analyzing the stability of discrete-time model predictive controllers acting on continuous-time systems. The proposed framework introduces the distinction between *discretization time* (used to generate the optimal control problem) and *sampling time* (used to implement the controller). The paper not only shows that these two time constants are independent, but also motivates the benefits of selecting a sampling time that is smaller than the discretization time. The resulting approach, hereafter referred to as Hypersampled Model Predictive Control, overcomes the traditional trade-off between performance and computational complexity that arises when selecting the sampling time of traditional discrete-time model predictive controllers.

## I. INTRODUCTION

Model Predictive Control (MPC) is a popular constrained control strategy for nonlinear systems. The idea behind MPC is to solve an Optimal Control Problem (OCP) at every instant and apply the first step of the optimal control sequence to the system. Although MPC is widely used due to its stability, feasibility, and robustness properties [1], its widespread adoption is hindered by the fact that solving the optimal control problem in real time can be challenging. To address this issue, extensive research has been dedicated to the development of increasingly efficient numerical solvers [2]–[5]. This paper investigates a practical stratagem that reduces computational complexity by decoupling the prediction model (used to formulate the OCP) from the system model (used to prove closed-loop stability).

Given a continuous-time system, MPC can be formally implemented in one of three ways: the most popular approach [6] is to discretize the dynamics and use discrete-time MPC (DT-MPC). Another option is sampled-data MPC [7], where the OCP features piecewise constant control inputs and continuous-time dynamics. In both cases, the prediction model is built using the same sampling rate as the controller. Thus, increasing the sampling rate also increases the numerical complexity of the OCP. The final option is Dynamically Embedded MPC (DE-MPC), which replaces the OCP solver with a dynamic compensator that runs parallel to the system [8], [9]. In this case, the controller is continuous whereas the prediction model is discrete.

This paper provides formal stability proofs for a fourth strategy, hereafter denoted Hypersampled MPC (HMPC), which decouples the sampling rate of the controller from the timestep used to perform trajectory predictions. This stratagem, which some practitioners employ despite the absence of theoretical guarantees, reduces the computational

burden of MPC by solving a coarse OCP (thereby reducing computational effort) and running the controller at a high sample rate (thereby improving reaction times).

The paper is organized as follows. Section III summarizes the ideal continuous-time MPC formulation and details its main implementation challenges. Section IV analyzes how the discretization of the continuous-time OCP affects the stability of the closed-loop system. Section V addresses how the stability of the closed-loop system is affected by the sampling time. Section VI features numerical comparisons between the proposed method and two traditional discrete-time MPC schemes: one implemented at a high sampling rate and one implemented at a low sampling rate.

## II. NOTATION

The paper employs the following norm conventions.  $\mathbb{R}^n$  is the set of  $n$ -dimensional vectors of real numbers,  $\mathbb{N}$  is the set of natural numbers, and  $\mathbb{N}^+$  is the set of non-zero natural numbers. Given a vector  $x \in \mathbb{R}^n$ , its 2-norm is  $\|x\| := \sqrt{x^\top x}$ . Given a positive semi-definite matrix  $Q \geq 0$ , the matrix norm of vector  $x$  is  $\|x\|_Q := \sqrt{x^\top Q x}$ . Given a function  $x(t)$ , its limit superior satisfies

$$\exists t^* \geq 0 : \overline{\lim_{t \rightarrow \infty}} \|x(t)\| \geq \|x(\tau)\|, \quad \forall \tau \geq t^*, \quad (1)$$

and its infinity norm is  $\|x(t)\|_\infty := \sup_{t \in [0, \infty)} \|x(t)\|$ .

## III. PROBLEM STATEMENT

Consider a continuous-time system

$$\dot{x}(t) = f(x(t), u(t)) \quad (2)$$

where  $x \in \mathbb{R}^n$  is the state,  $u \in \mathbb{R}^m$  is the control input, and  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is the system dynamics. The system is subject to state and input constraints  $x \in \mathcal{X}$  and  $u \in \mathcal{U}$ , where  $\mathcal{X} \subseteq \mathbb{R}^n$  and  $\mathcal{U} \subseteq \mathbb{R}^m$  are closed convex sets.

This system can be controlled using continuous-time MPC (CT-MPC) by solving the optimal control problem

$$\min_{\mu(\tau)} J(\xi(T)) + \int_0^T l(\xi(\tau), \mu(\tau)) d\tau \quad (3a)$$

$$\text{s.t. } \xi(0) = x, \quad (3b)$$

$$\dot{\xi}(\tau) = f(\xi(\tau), \mu(\tau)) \quad \forall \tau \in [0, T], \quad (3c)$$

$$\mu(\tau) \in \mathcal{U}, \quad \forall \tau \in [0, T], \quad (3d)$$

$$\xi(\tau) \in \mathcal{X}, \quad \forall \tau \in [0, T], \quad (3e)$$

$$\xi(T) \in \Omega, \quad (3f)$$

where  $\xi : [0, T] \rightarrow \mathbb{R}^n$ ,  $\mu : [0, T] \rightarrow \mathbb{R}^m$  are the predicted state and input,  $T > 0$  is the prediction horizon,  $J : \mathbb{R}^n \rightarrow \mathbb{R}$  is the terminal cost,  $l : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  is the incremental

cost and  $\Omega \subseteq \mathcal{X}$  is the terminal constraint set. Specifically, the CT-MPC feedback law is  $u(x) = \mu^*(0|x)$ , where  $\mu^*(\tau|x)$  denotes the solution mapping of (3) for a given  $x$ . As detailed in [1], the following assumption is sufficient to ensure that the optimal control problem (3) is recursively feasible and that solving it leads to a stabilizing control action.

**Assumption 1.** *The dynamics  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  are Lipschitz continuous, stabilizable, and admit the origin as a constraint-admissible equilibrium point. The cost functions  $J : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $l : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  are convex, twice continuously differentiable, zero at the origin, and upper/lower bounded by quadratic functions. In addition,  $\Omega \subseteq \mathcal{X}$  is a closed and convex set such that there exists a terminal control law  $\kappa : \mathbb{R}^n \rightarrow \mathbb{R}^m$  that satisfies*

$$\kappa(x) \in \mathcal{U}, \quad \forall x \in \Omega, \quad (4a)$$

$$f(x, \kappa(x)) \in T_\Omega(x), \quad \forall x \in \Omega, \quad (4b)$$

$$\nabla J(x)f(x, \kappa(x)) + l(x, \kappa(x)) \leq 0, \quad \forall x \in \Omega, \quad (4c)$$

where  $T_\Omega(x)$  is the tangent cone (defined in [10]) of  $\Omega$  evaluated in  $x$ .

Under these standard assumptions, the following theorem is easily deduced from existing literature [1].

**Theorem 1.** *Under Assumption 1, there exists a class  $\mathcal{K}$  function  $\gamma_1$  and positive scalars  $\chi_1, \Delta_1$  such that*

$$\dot{x} = f(x, \mu^*(0|x) + \delta_1) \quad (5)$$

satisfies

$$\overline{\lim}_{t \rightarrow \infty} \|x(t)\| \leq \gamma_1 \left( \overline{\lim}_{t \rightarrow \infty} \|\delta_1(t)\| \right), \quad (6)$$

whenever  $\|x(0)\| \leq \chi_1$  and  $\|\delta_1\|_\infty \leq \Delta_1$ .

*Proof.* As detailed in [1, Section 3.6], the origin of (5) is exponentially stable for  $\delta_1 = 0$ . Therefore, it is locally Input-to-State Stable (ISS) for sufficiently small  $\delta_1 \neq 0$ .  $\square$

This result states that CT-MPC is robust with respect to additive input disturbances. Unfortunately, its implementation poses two major challenges: i) the OCP (3) is formulated in a Banach function space and requires sophisticated solvers, e.g. [11], to compute  $\mu^*(\tau|x)$ . ii) Implementing the controller in continuous-time requires solving (3) “instantaneously”, which is not realistic in practice.

Due to the limitations of CT-MPC, it is common practice to discretize the system dynamics and employ DT-MPC. This approach assumes that the dynamic model of the system matches the prediction model of the OCP. An unfortunate consequence is that, for a fixed prediction horizon, reducing the sampling time inevitably leads to an increased number of prediction steps. This has the combined negative effect of increasing the numerical complexity of the OCP while also decreasing the allocated time for solving it.

This paper seeks to formalize the distinction between *discretization time*  $t_d$ , i.e., the step size used to discretize the dynamic model (2), and *sampling time*  $t_s$ , i.e., the time at which the controller is implemented. Doing so will require

substantially different stability proofs from traditional MPC literature (e.g. [1], [6], [7]), which heavily relies on having a prediction model that matches the system dynamics.

#### IV. DISCRETIZED OPTIMAL CONTROL PROBLEM

This section addresses the first challenge of CT-MPC, which is that (3) is an infinite dimensional OCP. Given  $N \in \mathbb{N}^+$ , let  $t_d = T/N$  be the discretization time and let  $f_d : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  be a discrete-time model such that

$$\lim_{t_d \rightarrow 0} \frac{f_d(x, u) - x}{t_d} = f(x, u). \quad (7)$$

Then, the continuous-time optimal control problem (3) can be approximated as

$$\min_{\mu_j} J(\xi_N) + \sum_{j=0}^{N-1} l_d(\xi_j, \mu_j) \quad (8a)$$

$$\text{s.t. } \xi_0 = x, \quad (8b)$$

$$\xi_{j+1} = f_d(\xi_j, \mu_j) \quad \forall j \in [0, N-1], \quad (8c)$$

$$\mu_j \in \mathcal{U}, \quad \forall j \in [0, N-1], \quad (8d)$$

$$\xi_j \in \mathcal{X}, \quad \forall j \in [1, N-1], \quad (8e)$$

$$\xi_N \in \Omega, \quad (8f)$$

with  $l_d(x, u) = t_d l(x, u)$ . Let  $\mu_j^*(x)$  denote the solution mapping of (8). The following assumption ensures that  $\mu_j^*(x)$  is a suitable approximation of  $\mu^*(jt_d|x)$ .

**Assumption 2.** *The parametrized optimal control problem (3) is strongly regular. Moreover, its discrete approximation (8) is strongly regular and admits  $T_d > 0$  and  $\chi_0 > 0$  such that the discretization error  $\Delta\mu = \mu_0^*(x) - \mu^*(0|x)$  satisfies*

$$\|\Delta\mu\| \leq L(t_d)\|x\|, \quad \forall \|x\| \leq \chi_0, \quad (9)$$

where  $L(t_d) > 0$ ,  $\forall t_d \in (0, T_d]$  is a Lipschitz constant, and

$$\lim_{t_d \rightarrow 0} L(t_d) = 0. \quad (10)$$

Although Assumption 2 is formally difficult to verify (see [12] for the case  $f_d(x, u) = x + t_d f(x, u)$  and  $\mathcal{X}, \Omega = \mathbb{R}^n$ ), it is reasonable to expect that (3) and (8) converge to the same solution as the discretization step  $t_d$  goes to zero. The following theorem states that, given a sufficiently small  $t_d$ , the CT-MPC law  $u(x) = \mu^*(0|x)$ , which relies on the solution to (3), can be replaced by the approximate control law  $u(x) = \mu_0^*(x)$ , obtained by solving (8).

**Theorem 2.** *Under Assumptions 1-2, given a sufficiently small discretization step  $t_d > 0$ , there exist positive scalars  $\chi_2, \Delta_2$  such that the origin of the closed-loop system*

$$\dot{x} = f(x, \mu_0^*(x) + \delta_2) \quad (11)$$

is locally ISS with state and input restrictions  $\|x(0)\| \leq \chi_2$  and  $\|\delta_2\|_\infty \leq \Delta_2$ .

*Proof.* System (11) can be rewritten as

$$\dot{x} = f(x, \mu^*(0|x) + \Delta\mu + \delta_2). \quad (12)$$

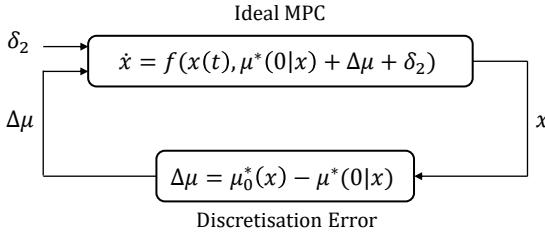


Fig. 1. Discretized Model Predictive Control law applied to the continuous system. The interconnection is ISS with respect to the error introduced by discretization.

Given  $\delta_2 = 0$ , it follows from Theorem 1 that system (12) is locally ISS with asymptotic gain

$$\overline{\lim_{t \rightarrow \infty}} \|x(t)\| \leq \gamma_1 \left( \overline{\lim_{t \rightarrow \infty}} \|\Delta\mu(t)\| \right). \quad (13)$$

Moreover, it follows from Assumption 2 that the solution error between (3) and (8) satisfies

$$\overline{\lim_{t \rightarrow \infty}} \|\Delta\mu(t)\| \leq L(t_d) \overline{\lim_{t \rightarrow \infty}} \|x(t)\|. \quad (14)$$

Referring to Figure 1, it follows from the small gain theorem that the interconnection between the dynamic system (12) and the static system  $\Delta\mu = \mu_0^*(x) - \mu^*(0|x)$ , is locally ISS with respect to  $\delta_2 \neq 0$  if the condition

$$L(t_d) \gamma_1(s) < s \quad (15)$$

is satisfied for all  $s \leq \Delta_1$ . Since Assumption 2 guarantees  $\lim_{t_d \rightarrow 0} L(t_d) = 0$ , it is always possible to select a sufficiently small  $t_d$  such that (15) holds. Given an input restriction  $\Delta_2 \in (0, \Delta_1)$ , it is then possible to select a state restriction  $\chi_2 > 0$  such that  $\|x(0)\| \leq \chi_2$  ensures  $\|x(t)\| \leq \min(\chi_0, \chi_1)$  and  $\|\Delta\mu(t)\| \leq \Delta_1 - \Delta_2$ ,  $\forall t \geq 0$ .  $\square$

Theorem 2 states that it is possible to implement an approximate CT-MPC law by solving (8) instead of (3). Although (8) is a standard Non-Linear Program (NLP), which is arguably simpler to solve compared to (3), it would be unrealistic to assume that the NLP solution can be computed ‘instantaneously’.

## V. HYPERSAMPLED MODEL PREDICTIVE CONTROL

This section addresses the second challenge of CT-MPC, which is the fact that solving (8) requires a finite amount of time. To address this issue, let  $t_s > 0$  be the sampling time of the control law, let

$$x_k(t) = x(kt_s), \quad \forall t \in [kt_s, (k+1)t_s). \quad (16)$$

with  $k \in \mathbb{N}$ , be the sampled state of the system, and let  $u(t) = \mu_0^*(x_k(t))$  be a zero-order hold control input.

The following theorem states that, given a sufficiently small sampling time, the closed-loop system maintains all its local ISS properties.

**Theorem 3.** *Under Assumptions 1-2, given a sufficiently small discretization step  $t_d > 0$  and sampling time  $t_s > 0$ ,*

*there exist positive scalars  $\chi_3, \Delta_3$  such that the origin of the closed-loop system*

$$\dot{x} = f(x, \mu_0^*(x_k) + \delta_3) \quad (17)$$

*is locally ISS with state and input restrictions  $\|x(0)\| \leq \chi_3$  and  $\|\delta_3\|_\infty \leq \Delta_3$ .*

*Proof.* Due to Theorem 2, there exists a sufficiently small  $t_d > 0$  such that (11) is locally ISS with respect to input disturbances. It then follows from [13, Theorem 2] that there exists a sufficiently small  $t_s > 0$  such that the sampled-data implementation (17) is also locally ISS.  $\square$

Figure 2 clarifies the various components of HMPC. Theorem 3 highlights the fact that the discretization time  $t_d$  and the sampling time  $t_s$  can be treated as two entirely different entities. This distinction introduces an additional degree of freedom that can be leveraged for HMPC design. In particular, we note the following properties:

- **Benefits of Decreasing  $t_s$ :** Reducing the sampling time tends to improve the performance of the controller by making it more reactive to external disturbances. Generally speaking, one would like  $t_s$  to be as small as possible to mimic continuous-time behaviour.
- **Limit for Decreasing  $t_s$ :** Due to real-time implementation requirements,  $t_s$  is lower-bounded by the computational time required to solve (8).
- **Benefits of Increasing  $t_d$ :** Given a fixed prediction horizon  $T$ , a larger discretization time step  $t_d$  leads to less prediction steps  $N$ . Since the computational complexity of (8) scales with  $N$ , it is generally beneficial for  $t_d$  to be as large as possible.
- **Limit for Increasing  $t_d$ :** Since larger values of  $t_d$  increase the discrepancy between  $\mu^*(0|x)$  and  $\mu_0^*(x)$ , an upper bound on the maximal admissible error will translate into an upper bound on  $t_d$ .

Since traditional MPC inherently assumes  $t_s = t_d$ , it can be challenging to reduce the sampling time because doing so increases the computational complexity while tightening the real-time requirements. The HMPC framework overcomes this preconceived trade-off by decoupling the two effects: given a fixed discretization time  $t_d$ , the OCP complexity is unaffected by the sampling time  $t_s$ .

In practice, any discrete-time MPC algorithm can be implemented as HMPC by selecting a coarse discretization step  $t_d$  and then running the controller at a faster sampling rate  $t_s$ . As a result, the HMPC framework can be adopted without any additional effort from the user. Moreover, since Theorem 3 proves that the closed-loop system is locally ISS with respect to input disturbances  $\delta_3$ , the proposed scheme can be seamlessly combined with other MPC schemes that rely on ISS-based proofs. Notably, HMPC can be combined with DE-MPC [9] by replacing the numerical solver used to compute  $\mu_0^*(x)$  with a dynamic compensator that tracks the solution of (8) with a bounded error.

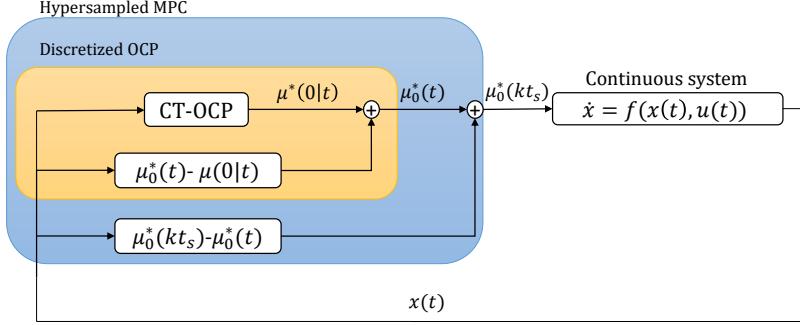


Fig. 2. Hypersampled Model Predictive Control. The continuous system is sampled at timestep  $t_s$  to generate  $x_k$ , which is then used to compute the optimal control law to the discretized MPC problem.

## VI. NUMERICAL EXAMPLES

This section illustrates the benefits of distinguishing between  $t_d$  and  $t_s$  using a linear and a nonlinear example. All timing data is obtained using FBstab [5] to solve the OCP.

### A. Double Integrator

Consider the double integrator system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d. \quad (18)$$

where  $d$  is an external disturbance and the state and input constraint sets are  $\mathcal{X} = \{x \in \mathbb{R}^2 : \|x_1\| \leq 2, \|x_2\| \leq 0.4\}$  and  $\mathcal{U} = \{u \in \mathbb{R} : -4 \leq u \leq 10\}$ . Given the initial condition  $x_0 = [2, 0]^\top$  and control horizon  $T = 2$ , consider the OCP

$$\min \|\xi(T)\|_P^2 + \int_0^T \|\xi(\tau)\|_Q^2 + \|\mu(\tau)\|_R^2 d\tau \quad (19a)$$

$$\text{s.t. } \xi(0) = x, \quad (19b)$$

$$\dot{\xi} = A\xi + B\mu, \quad \forall \tau \in [0, T] \quad (19c)$$

$$\mu(\tau) \in \mathcal{U}, \quad \forall \tau \in [0, T], \quad (19d)$$

$$\xi(\tau) \in \mathcal{X}, \quad \forall \tau \in [0, T], \quad (19e)$$

where  $Q = \text{diag}(1, 0)$ ,  $R = 0.04$ ,  $P$  is the solution to the algebraic Riccati equation  $A^\top P + PA - PBR^{-1}B^\top P + Q = 0$ , and  $\mathcal{O}_\infty$  is the maximal output admissible set [14] associated to the corresponding linear-quadratic regulator.

To study the effect of discretization, consider the OCP

$$\min \|\xi_N\|_P^2 + \sum_{i=0}^{N-1} \|\xi_i\|_{Q_d}^2 + \|\mu_i\|_{R_d}^2 \quad (20a)$$

$$\text{s.t. } \xi_{i+1} = A_d \xi_i + B_d \mu_i, \quad \forall i \in [0, N-1] \quad (20b)$$

$$\xi_0 = x, \quad (20c)$$

$$\mu_i \in \mathcal{U}, \quad \forall i \in [0, N-1], \quad (20d)$$

$$\xi_i \in \mathcal{X}, \quad \forall i \in [1, N] \quad (20e)$$

where  $A_d, B_d$  are the discretized version of the continuous dynamics (18) and  $Q_d = t_d Q$ ,  $R_d = t_d R$ ,  $N = T/t_d \in \mathbb{N}^+$ .

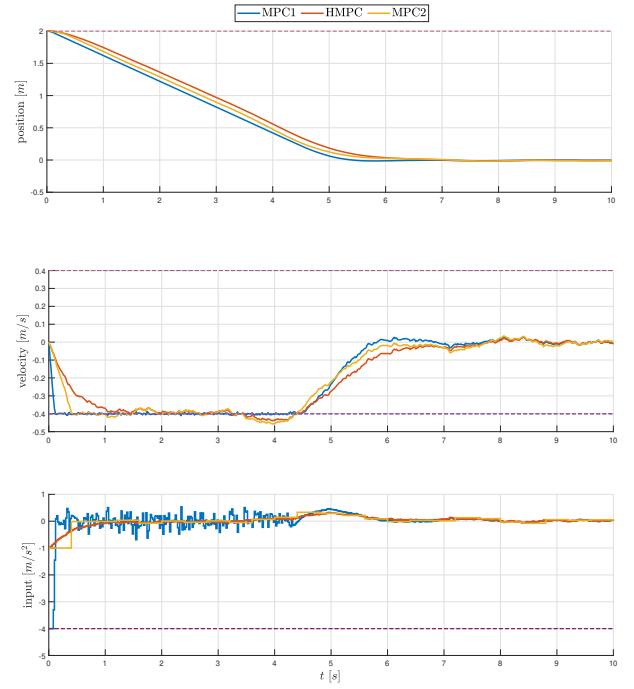


Fig. 3. Comparison of the position, velocity, and input trajectories for the three MPC schemes subject to an additive disturbance on the input.

Figure 3 compares the closed loop response obtained using three different schemes:

- MPC1:  $t_s = t_d = 0.02$ ;
- HMPC:  $t_s = 0.02$  and  $t_d = 0.4$ ;
- MPC2:  $t_s = t_d = 0.4$ .

Although the position trajectories for the three schemes perform similarly, the velocity and input trajectories highlight how the three schemes react to external disturbances in proximity to the constraint boundary. MPC1 is the only scheme that enforces the velocity constraints in the presence of disturbances. HMPC and MPC2 both violate constraints, with HMPC featuring slightly better disturbance rejection properties due to its faster response time.

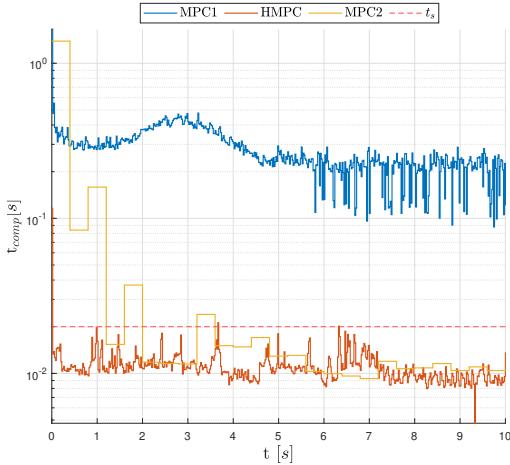


Fig. 4. The computation time for the three schemes with noise, compared to the sampling time  $t_s = 0.02$ . HMPC is the only scheme that consistently satisfies the real-time requirements.

The disadvantage of MPC1 becomes apparent by examining the computation time required to solve the underlying OCP (20). As shown in Figure 4, the time required to solve MPC1 is approximately 30 times more than HMPC and MPC2. This is because MPC1 has a prediction length of  $N = 100$  steps, whereas HMPC and MPC2 have a prediction length of only  $N = 5$  steps.

The interest in the HMPC framework is that it enables the following design choice: “Given a sampling time  $t_s$ , is it possible to reliably solve discretized OCP with  $t_d = t_s$ ?” If so, implementing the MPC1 solution is arguably the best option. If not, the computational cost can be reduced by maintaining the same sampling rate while selecting  $t_d > t_s$  to reduce the number of optimization variables. The resulting control law is preferable to increasing the sampling time. Although the advantage of HMPC over MPC2 is marginal in the case of the double integrator, the following example shows that the difference between the two schemes can be quite severe.

### B. Nonlinear Lane Change

To emphasize the advantages of HMPC with a more complex example, we consider the nonlinear lane change system detailed in [15]. The states and control inputs are  $x = [y, \psi, v, \omega, \delta_f, \delta_r]^\top$  and  $u = [\dot{\delta}_f, \dot{\delta}_r]$ , where,  $y$  is the lateral position,  $v$  is the lateral velocity,  $\psi$  is the yaw angle,  $\omega$  is the yaw rate,  $\delta_f$  is the front steering angle, and  $\delta_r$  is

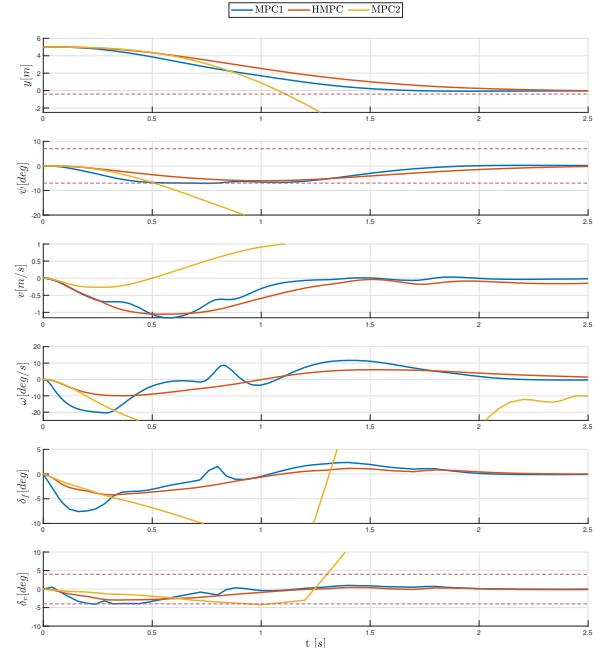


Fig. 5. The state trajectories for the nonlinear lane change model. The trajectories of MPC1 and HMPC perform similarly, whereas MPC2 leads to an unstable closed-loop response.

the rear steering angle. The system dynamics are

$$\dot{y} = s \sin(\psi) + v \cos(\psi) \quad (21a)$$

$$\dot{\psi} = \omega \quad (21b)$$

$$\dot{v} = -s \omega + \frac{F \alpha_f \cos(\delta_f) + F \alpha_r \cos(\delta_r) + F_w}{m} \quad (21c)$$

$$\dot{\omega} = \frac{F \alpha_f \cos(\delta_f) l_f - F \alpha_r \cos(\delta_r) l_r}{I_{zz}} \quad (21d)$$

$$\dot{\delta}_f = u_1, \quad \dot{\delta}_r = u_2 \quad (21e)$$

The functions  $F(\alpha), \alpha_f, \alpha_r, F_w$  along with parameters such as  $m, I_{zz}, l_f, l_r, \mu$  can be found in [15]. We take the initial condition  $x_0 = [5, 0, 0, 0, 0, 0]^\top$  and use the same cost function as (19), with  $Q = I_6$ ,  $R = I_2$ , and  $T = 2$ . The constraints on the system are

$$y \in [-0.4, 10], \quad \psi \in [-7^\circ, 7^\circ]$$

$$\delta_f \in [-35^\circ, 35^\circ], \quad \delta_r \in [-4^\circ, 4^\circ]$$

$$u_1 \in [-1.2, 1.2], \quad u_2 \in [-0.6, 0.6]$$

Figure 5 and Figure 6 compare the state and input trajectories obtained using three different schemes:

- MPC1:  $t_s = t_d = 0.04$ ;
- HMPC:  $t_s = 0.04$  and  $t_d = 0.2$ ;
- MPC2:  $t_s = t_d = 0.2$ .

The trajectories are comparable for the MPC1 and HMPC schemes, whereas implementing MPC2 causes the system to become unstable. This is because, although 0.2s is too large to be a suitable sampling time, it is still sufficiently small to ensure that (8) is a suitable approximation of (3).

Figure 7 plots the computational time taken at each step for solving the OCP for the three schemes. As with the

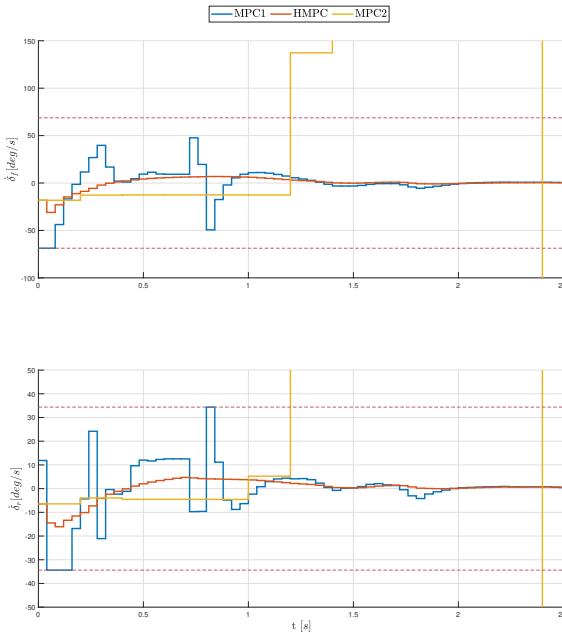


Fig. 6. The input trajectory for the nonlinear lane change model.

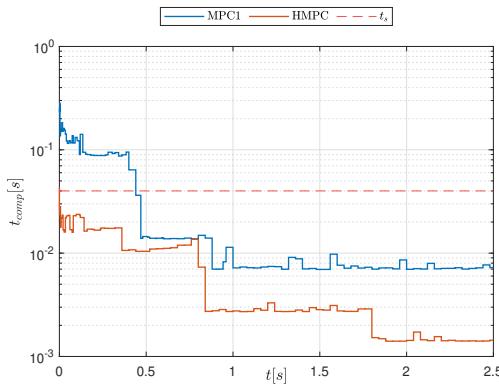


Fig. 7. Computation time utilized by MPC1 and HMPC compared to the sampling time  $t_s$ . The computation time of MPC2 is not included due to the fact that the response is unstable.

double integrator, the time taken to solve the OCP of MPC1 is consistently larger than the time used for solving the OCP of the HMPC (5 times on average). Note that the time taken to implement MPC2 was omitted from the figure due to the fact that its unstable behaviour led to significantly larger computation times. Once again, we note that MPC1 cannot be implemented in real-time because the computational time required to solve the OCP is higher than the sampling time  $t_s$ . Conversely, HMPC significantly reduces the computational effort by choosing a coarse discretization step  $t_d$  while keeping  $t_s$  unchanged.

## VII. CONCLUSION AND FUTURE WORK

This paper analyzed the stability properties of Hypersampled Model Predictive Control, which is a stratagem for

decoupling the sampling time of MPC from the computational complexity of the underlying optimal control problem. Unlike existing MPC stability proofs, the proposed stability analysis does not assume that the system dynamics match the prediction model. This important distinction enables the sampling time and discretization time to be treated as two fully independent quantities. By enabling the control designer to select each time constants based on different considerations, HMPC provides a simple yet effective way to increase the sampling rate of MPC without increasing its computational footprint. Theoretical analysis shows that the HMPC scheme is Input-to-State Stable to input disturbances. Future work will focus on ensuring robust constraint enforcement.

## REFERENCES

- [1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [2] M. J. Tenny, S. J. Wright, and J. B. Rawlings, “Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming,” *Computational Optimization and Applications*, vol. 28, pp. 87–121, 2004.
- [3] M. Diehl, H. J. Ferreau, and N. Haverbeke, “Efficient numerical methods for nonlinear MPC and moving horizon estimation,” *Nonlinear model predictive control: towards new challenging applications*, pp. 391–417, 2009.
- [4] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, “Autogenerating microsecond solvers for nonlinear MPC: a tutorial using acado integrators,” *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 685–704, 2015.
- [5] D. Liao-McPherson, M. Huang, and I. Kolmanovsky, “A regularized and smoothed Fischer–Burmeister method for quadratic programming with applications to model predictive control,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2937–2944, 2018.
- [6] J. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design (Second Edition)*. Madison, WI: Nob Hill Publishing, 2017.
- [7] L. Magni and R. Scattolini, “Model predictive control of continuous-time nonlinear systems with piecewise constant control,” *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 900–906, 2004.
- [8] C. Feller and C. Ebenbauer, “Relaxed logarithmic barrier function based model predictive control of linear systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1123–1238, 2017.
- [9] M. M. Nicotra, D. Liao-McPherson, and I. V. Kolmanovsky, “Dynamically embedded model predictive control,” in *2018 Annual American Control Conference (ACC)*, pp. 4957–4962, 2018.
- [10] F. Blanchini, “Set invariance in control,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
- [11] J. Hauser, “A projection operator approach to the optimization of trajectory functionals,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 377–382, 2002.
- [12] A. L. Dontchev, I. V. Kolmanovsky, M. I. Krastanov, M. M. Nicotra, and V. M. Veliov, “Lipschitz stability in discretized optimal control with application to SQP,” *SIAM Journal on Control and Optimization*, vol. 57, no. 1, pp. 468–489, 2019.
- [13] A. Teel, D. Nesic, and P. Kokotovic, “A note on input-to-state stability of sampled-data nonlinear systems,” in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 3, pp. 2473–2478, 1998.
- [14] E. G. Gilbert and K. T. Tan, “Linear systems with state and control constraints: The theory and application of maximal output admissible sets,” *IEEE Transactions on Automatic control*, vol. 36, no. 9, pp. 1008–1020, 1991.
- [15] D. Liao-McPherson, M. M. Nicotra, and I. V. Kolmanovsky, “Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction,” *Automatica*, vol. 117, p. 108973, 2020.