

A Contingency Model Predictive Control Framework for Safe Learning

Merlijne Geurts*, Tren Baltussen*, *Member, IEEE*, Alexander Katriniok, *Senior Member, IEEE*, Maurice Heemels, *Fellow, IEEE*

Abstract—This research introduces a multi-horizon contingency model predictive control (CMPC) framework in which classes of robust MPC (RMPC) algorithms are combined with classes of learning-based MPC (LB-MPC) algorithms to enable safe learning. We prove that the CMPC framework inherits the robust recursive feasibility properties of the underlying RMPC scheme, thereby ensuring safety of the CMPC in the sense of constraint satisfaction. The CMPC leverages the LB-MPC to safely learn the unmodeled dynamics to reduce conservatism and improve performance compared to standalone RMPC schemes, which are conservative in nature. In addition, we present an implementation of the CMPC framework that combines a particular RMPC and a Gaussian Process MPC scheme. A simulation study on automated lane merging demonstrates the advantages of our general CMPC framework.

Index Terms—Autonomous systems, Predictive control for nonlinear systems, Uncertain systems

I. INTRODUCTION

MODEL predictive control (MPC) is a well-established control method for safety-critical systems, as MPC can ensure safety in terms of constraint satisfaction while attaining high performance. In particular, robust MPC (RMPC) can provide rigorous safety guarantees under all modeled uncertainties through robust recursive feasibility properties [1], [2]. Unfortunately, RMPC can be overly conservative and inflexible as it accounts for the worst-case disturbance at all times, which limits its performance and applicability. Alternatively, learning-based MPC (LB-MPC) methods use data-driven and machine learning techniques, for instance, to improve the system prediction model, and consequently to increase performance [3], [4]. Although LB-MPC methods can account for uncertainty, providing rigorous safety guarantees for LB-MPC methods is still an open research problem [4].

In order to ensure constraint satisfaction, LB-MPC methods are often combined with robust control methods. For example, [5] uses reinforcement learning to safely adapt the parametrization of an RMPC. Further, robust predictive safety filters can be used to correct/modify control actions when a learning-based controller will jeopardize safety [6]. In [7] an LB-MPC optimizes a learning-based cost function subject

to robust constraints. Various existing methods compute a separate high-performance plan and a robust backup plan, and use this backup plan when safety is compromised (see e.g., [6], [8], [9]). A potential drawback of these approaches is erratic intervention and switching between controllers, and possibly sub-optimal control performance. We refer to [4] and [10] for an overview on safe LB-MPC.

An alternative approach is Contingency MPC (CMPC) [11], which uses two separate MPC horizons to compute both a performance and a contingency input sequence. Instead of computing a single input sequence as, e.g., in [7], [12], CMPC simultaneously computes two input sequences that are coupled by their first input. As a result, CMPC can leverage the assertive nature of LB-MPC while ensuring (safety) constraint satisfaction via the underlying RMPC, without switching between the two plans. Interestingly, there exist several relevant works in which CMPC is applied to various safety-critical systems, mostly in automated driving and traffic applications [11], [13]–[15]. However, these works either lack formal safety guarantees [11], [13], [14] or do not consider LB-MPC [15]. As CMPC presents a powerful methodology for safe learning-based MPC, a general conceptual framework with high-level conditions for robust recursive feasibility can be of great interest for safe learning.

To fill this gap and as the main contribution of this paper, we propose a conceptual CMPC framework which allows the integration of various classes of LB-MPC and RMPC schemes (Sec. III), resulting in a framework of which [7] and [15] are special cases. We show that the CMPC inherits the safety guarantees from the underlying RMPC by formally proving robust recursive feasibility of the CMPC framework (Sec. IV). This will highlight the basic assumptions required by the underlying MPC schemes to enable safe learning. Moreover, we demonstrate how the CMPC framework can be used by presenting a specific implementation using the RMPC in [16] and a Gaussian Process MPC (GP-MPC) [17] (Sec. V–VI). Finally, the CMPC is applied to a lane-merging use case with an autonomous vehicle and is compared to the baseline RMPC and GP-MPC (Sec. VII). CMPC shows improved performance over the standalone RMPC while guaranteeing safety as opposed to the standalone GP-MPC.

II. PROBLEM FORMULATION

We consider the discrete-time non-linear system \mathcal{S} subject to uncertainty, of the form

$$x_{k+1} = f(x_k, u_k) + \underbrace{g(x_k, u_k) + v_k}_{=:w_k}, \quad (1)$$

Received March 17, 2025; revised May 5, 2025; accepted May 19, 2025. Date of publication XXX XX 2025, date of current version XXX XX 2025. This research has received funding from the Dutch Research Council (NWO) via AMADeUS, project no. 18489.

*Merlijne Geurts and Tren Baltussen contributed equally to this work. (Corresponding author: Tren Baltussen).

All authors are with the Control Systems Technology Section, Eindhoven University of Technology, The Netherlands {m.e.geurts, t.m.j.t.baltussen, a.katriniok, m.heemels}@tue.nl.

Digital Object Identifier 10.1109/LCSYS.2025.3575191

where $x_k \in \mathbb{R}^{n_x}$ and $u_k \in \mathbb{R}^{n_u}$ denote the state and control input, respectively, of \mathcal{S} at time step $k \in \mathbb{N} := \{0, 1, 2, \dots\}$. The dynamics of \mathcal{S} are composed of known, nominal dynamics denoted by $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$, of unknown, unmodeled dynamics $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$, and process noise $v_k \in \mathbb{V} \subset \mathbb{R}^{n_x}$. The system is subject to state constraints $x_k \in \mathbb{X}$ and input constraints $u_k \in \mathbb{U}$, $k \in \mathbb{N}$. The disturbance $w_k = g(x_k, u_k) + v_k$ is assumed to satisfy $w_k \in \mathbb{W} \subset \mathbb{R}^{n_x}$, for all $k \in \mathbb{N}$, where \mathbb{W} is a bounded set and $0 \in \mathbb{W}$. In this paper, we aim to address the problem of controlling system \mathcal{S} and learning the unmodeled dynamics g to improve the performance while providing theoretical safety guarantees in the sense of constraint satisfaction.

III. CMPC FRAMEWORK FOR SAFE LEARNING

To address the problem formulated above, we propose a general multi-horizon CMPC framework that uses two prediction models, based on the following optimal control problem (OCP) for a given state x_k at time $k \in \mathbb{N}$:

$$\min_{\bar{x}_{k|k}, \bar{u}_k, \bar{U}_k} J(x_k, \bar{x}_{k|k}, \bar{U}_k, \hat{U}_k) \quad (2a)$$

$$\text{s.t. } \bar{x}_{k+j+1|k} = f(\bar{x}_{k+j|k}, \bar{u}_{k+j|k}), \quad j = 0, 1, \dots, N-1, \quad (2b)$$

$$\hat{x}_{k+j+1|k} = f(\hat{x}_{k+j|k}, \hat{u}_{k+j|k}) + d(\hat{x}_{k+j|k}, \hat{u}_{k+j|k}, \mathcal{D}_k) \\ j = 0, 1, \dots, N-1, \quad (2c)$$

$$\bar{x}_{k|k} = \hat{x}_{k|k} \in x_k \oplus X, \quad (2d)$$

$$\bar{u}_{k|k} = \hat{u}_{k|k}, \quad (2e)$$

$$\bar{u}_{k+j|k} \in \bar{\mathbb{U}}_j, \quad j = 0, 1, \dots, N-1, \quad (2f)$$

$$\hat{u}_{k+j|k} \in \hat{\mathbb{U}}_j, \quad j = 0, 1, \dots, N-1, \quad (2g)$$

$$\bar{x}_{k+j|k} \in \bar{\mathbb{X}}_j, \quad j = 0, 1, \dots, N, \quad (2h)$$

$$\hat{x}_{k+j|k} \in \hat{\mathbb{X}}_j, \quad j = 0, 1, \dots, N. \quad (2i)$$

The OCP features two separate input sequences (also referred to as two *horizons*), namely a robust input sequence $\bar{U}_k = (\bar{u}_{k|k}, \bar{u}_{k+1|k}, \dots, \bar{u}_{k+N-1|k})$ used in a nominal prediction model (2b) of \mathcal{S} with $w_k = 0$, $k \in \mathbb{N}$, and the performance input sequence $\hat{U}_k = (\hat{u}_{k|k}, \hat{u}_{k+1|k}, \dots, \hat{u}_{k+N-1|k})$ used in the LB prediction model (2c) of \mathcal{S} . Here, the notation $u_{k+j|k}$ is used to denote the prediction of the variable u at time $k+j$ made at time step k . The LB prediction model in (2c) consists of the nominal dynamics f and the residual dynamics estimated by $d(\hat{x}_{k+j|k}, \hat{u}_{k+j|k}, \mathcal{D}_k)$, using a data set \mathcal{D}_k available at time step k . The data set \mathcal{D}_k can consist of past collections of state and input samples. Note that in various applications this data is not available a priori and, hence, the model must be adapted online. If no data are available a priori, then $\mathcal{D}_0 = \emptyset$. The two horizons are coupled by sharing the initial state (2d), i.e., $\bar{x}_{k|k} = \hat{x}_{k|k}$ and the first control input (2e), i.e., $\bar{u}_{k|k} = \hat{u}_{k|k}$. Note that in (2d) the initial state $\hat{x}_{k|k} = \bar{x}_{k|k}$ must lie in the Minkowski sum $x_k \oplus X = \{x_k + \xi \mid \xi \in X\}$, where X can be a robust positive invariant set for \mathcal{S} , as sometimes used in tube-based MPC [18], or $X = \{0\}$ as used in classical robust MPC. Both input sequences are constrained according to (2f) and (2g), where the (possibly tightened) input constraint sets $\bar{\mathbb{U}}_j \subseteq \mathbb{U}$ can be selected as, e.g., in tube-MPC [18]. Similarly, we consider (tightened) constraint sets $\bar{\mathbb{X}}_j \subseteq \mathbb{X}$, $j = 0, 1, \dots, N$ in (2h)

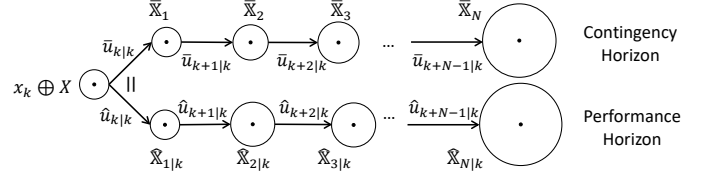


Fig. 1. A schematic of CMPC framework with circles indicating the admissible state constraint sets for each horizon.

as is common in various RMPC approaches [16], [18]. The LB horizon uses the data set \mathcal{D}_k to adapt the tightened state constraint set $\hat{\mathbb{X}}_j \subseteq \mathbb{X}$ in (2i) over time $k \in \mathbb{N}$. Finally, let $(\bar{x}_{k|k}^*, \bar{U}_k^*, \hat{U}_k^*)$ be a minimizer of (2) for x_k , which we assume exists, at time $k \in \mathbb{N}$ for cost function J . For simplicity we assume that the minimizer is unique, but all the results hold also in case of non-uniqueness.

The CMPC can be perceived as a combination of two separate MPCs, namely, an RMPC

$$\min_{\bar{x}_{k|k}, \bar{U}_k} J_R(x_k, \bar{x}_{k|k}, \bar{U}_k) \quad (3a)$$

$$\text{s.t. } (2b), (2f), (2h), \bar{x}_{k|k} \in x_k \oplus X \quad (3b)$$

and an LB-MPC

$$\min_{\hat{x}_{k|k}, \hat{U}_k} J_L(x_k, \hat{x}_{k|k}, \hat{U}_k) \quad (4a)$$

$$\text{s.t. } (2c), (2g), (2i), \hat{x}_{k|k} \in x_k \oplus X \quad (4b)$$

that are coupled by (2d) and (2e). The cost function J can be chosen, for instance, as a convex combination of J_R and J_L . The CMPC (2) can balance the performance of the LB-MPC, while ensuring safety through the RMPC under appropriate assumptions, as we will demonstrate in this paper. Fig. 1 depicts the CMPC with the robust *contingency horizon* and the learning-based *performance horizon*. Note that CMPC (2) can accommodate various RMPC and LB-MPC methods by appropriately designing its components (3) and (4).

The resulting control input applied to \mathcal{S} at time k is $u_k = \pi(x_k) := \kappa(x_k, \bar{x}_{k|k}^*) + \bar{u}_{k|k}^* = \kappa(x_k, \hat{x}_{k|k}^*) + \hat{u}_{k|k}^*$, where an ancillary control law κ is used as, e.g., in some tube-based MPC setups [18]. Note that for classical robust MPC, where $X = \{0\}$, the ancillary control law κ is only used for constraint tightening and is not used in the closed-loop control law as κ is typically zero for $x_k = \bar{x}_{k|k} = \hat{x}_{k|k}$. In tube-based MPC the nominal state is relaxed to lie in a robust positive invariant set X around the nominal trajectory [18].

IV. SAFETY GUARANTEES

In this section, we present the assumptions on the adopted RMPC (3) and LB-MPC (4) that ensure robust recursive feasibility of the CMPC (2), i.e., if the CMPC is feasible at time $k \in \mathbb{N}$ then it is also feasible at the next time step $k+1$.

Assumption 1: The RMPC (3) is robustly recursively feasible in the sense that, if there is a pair $(\bar{x}_{k|k}, \bar{U}_k)$ that is feasible for (3), for state x_k at time $k \in \mathbb{N}$, then there is, for all $w_k \in \mathbb{W}$, a pair $(\bar{x}_{k+1|k+1}, \bar{U}_{k+1})$ that is feasible for (3), for the state $x_{k+1} = f(x_k, \kappa(x_k, \bar{x}_{k|k}) + \bar{u}_{k|k}) + w_k$ at time $k+1$.

Assumption 2: The learned residual dynamics satisfy $d(x_k, u_k, \mathcal{D}_k) \in \mathbb{W}$, and it holds that $\hat{\mathbb{X}}_j \supseteq \bar{\mathbb{X}}_j$, $j \in \{0, 1, \dots, N\}$, for all $k \in \mathbb{N}$.

Note that Ass. 1 can be guaranteed for various robust MPC schemes, such as tube-based MPC, using proper constraint tightening and terminal ingredients [2], [18]. Based on these assumptions, we can show that the CMPC (2) inherits the robust recursive feasibility of the RMPC (3):

Theorem 1: Let Ass. 1 and 2 hold. If OCP (2) has a feasible solution $(\bar{x}_{k|k}, \bar{U}_k, \hat{U}_k)$ for x_k at time $k \in \mathbb{N}$, then (2) is feasible for the next state $x_{k+1} = f(x_k, \kappa(x_k, \bar{x}_{k|k}) + \bar{u}_{k|k}) + w_k$, for any $w_k \in \mathbb{W}$, at time $k + 1$.

Proof: Given a feasible initial nominal state $\bar{x}_{k|k}$ and a pair of input sequences (\bar{U}_k, \hat{U}_k) , for state x_k at time $k \in \mathbb{N}$, we have that the corresponding predicted state trajectories satisfy $(\bar{x}_{k|k}, \bar{x}_{k+1|k}, \dots, \bar{x}_{k+N|k}) \in ([x_k \oplus X] \cap \bar{\mathbb{X}}_0) \times \bar{\mathbb{X}}_1 \times \dots \times \bar{\mathbb{X}}_{N-1} \times \bar{\mathbb{X}}_N$ and $(\hat{x}_{k|k}, \hat{x}_{k+1|k}, \dots, \hat{x}_{k+N|k}) \in ([x_k \oplus X] \cap \hat{\mathbb{X}}_0) \times \hat{\mathbb{X}}_1 \times \dots \times \hat{\mathbb{X}}_{N-1|k} \times \hat{\mathbb{X}}_{N|k}$. Due to Ass. 1, this implies that there exist a nominal initial state $\bar{x}_{k+1|k+1}$ and an input sequence \bar{U}_{k+1} that satisfy (3b) at time $k + 1$ for the state $x_{k+1} = f(x_k, \kappa(x_k, \bar{x}_{k|k}) + \bar{u}_{k|k}) + w_k$. Thus, for the corresponding state sequence, we have that $(\bar{x}_{k+1|k+1}, \bar{x}_{k+2|k+1}, \dots, \bar{x}_{k+N|k+1}, \bar{x}_{k+N+1|k+1}) \in ([x_{k+1} \oplus X] \cap \bar{\mathbb{X}}_0) \times \bar{\mathbb{X}}_1 \times \dots \times \bar{\mathbb{X}}_{N-1} \times \bar{\mathbb{X}}_N$. It remains to show that there exist a feasible initial nominal state and an input sequence for the performance horizon of problem (2). Due to Ass. 2, we have that $d(x_k, u_k, \mathcal{D}_{k+1}) \in \mathbb{W}$ and $\hat{\mathbb{X}}_{j|k+1} \supseteq \bar{\mathbb{X}}_j$. This implies that we can select $\hat{x}_{k+1|k+1} := \bar{x}_{k+1|k+1}$ and $\hat{U}_{k+1} := \bar{U}_{k+1}$ that leads to a state sequence $(\hat{x}_{k+1|k+1}, \hat{x}_{k+2|k+1}, \dots, \hat{x}_{k+N|k+1}, \hat{x}_{k+N+1|k+1}) \in ([x_{k+1} \oplus X \cap \hat{\mathbb{X}}_{0|k+1}]) \times \hat{\mathbb{X}}_{1|k+1} \times \dots \times \hat{\mathbb{X}}_{N-1|k+1} \times \hat{\mathbb{X}}_{N|k+1}$. Hence, there exists a feasible solution $(\bar{x}_{k+1|k+1}, \bar{U}_{k+1}, \bar{U}_{k+1})$ for x_{k+1} . ■

Theorem 1 shows that under Ass. 1 and 2, the CMPC inherits the robust recursive feasibility of the RMPC. When the estimated residual dynamics d fail to satisfy Ass. 2, we can use constraint softening for the LB-MPC to preserve robust recursive feasibility. Thereto, we replace (2) by

$$\min_{\bar{x}_{k|k}, \bar{U}_k, \hat{U}_k, \mathcal{E}_k} J(x_k, \bar{x}_{k|k}, \bar{U}_k, \hat{U}_k) + J_\varepsilon(\mathcal{E}_k) \quad (5a)$$

$$\text{s.t. (2b) – (2h),} \quad (5b)$$

$$\hat{x}_{k+j|k} \in \hat{\mathbb{X}}_{j|k}(\varepsilon_{k+j|k}), \quad \varepsilon_{k+j|k} \geq 0, \quad j = 0, 1, \dots, N, \quad (5c)$$

where $\mathcal{E}_k = (\varepsilon_{k|k}, \dots, \varepsilon_{k+N|k}) \in \mathbb{R}_{\geq 0}^{N+1}$ is a sequence of nonnegative slack variables and a corresponding penalty term $J_\varepsilon : \mathbb{R}_{\geq 0}^{N+1} \rightarrow \mathbb{R}_{\geq 0}$ is added to the cost function, with $J_\varepsilon(\mathcal{E}) > 0$, if $\mathcal{E} \neq 0$, and $J_\varepsilon(0) = 0$. The state constraints $\hat{\mathbb{X}}_{j|k}$ in (2) are assumed to be softened such that $\hat{\mathbb{X}}_{j|k}(0) = \bar{\mathbb{X}}_{j|k}$ and $\hat{\mathbb{X}}_{j|k}(\varepsilon) \rightarrow \mathbb{R}^{n_x}$ when $\varepsilon \rightarrow \infty$.

Theorem 2: Let Ass. 1 hold. If $(\bar{x}_{k|k}, \bar{U}_k, \hat{U}_k, \mathcal{E}_k)$ is a feasible quadruple for OCP (5) for x_k at time $k \in \mathbb{N}$, then (5) is feasible for the state $x_{k+1} = f(x_k, \kappa(x_k, \bar{x}_{k|k}) + \bar{u}_{k|k}) + w_k$ for any $w_k \in \mathbb{W}$ at time $k + 1$.

Proof: Similar to the proof of Thm. 1, based on Ass. 1, there exist a feasible nominal initial state $\bar{x}_{k+1|k+1}$ and an input sequence \bar{U}_{k+1} that satisfies (3b) at time $k+1$ for $x_{k+1} = f(x_k, \kappa(x_k, \bar{x}_{k|k}) + \bar{u}_{k|k}) + w_k$. Due to $\lim_{\varepsilon \rightarrow \infty} \hat{\mathbb{X}}_{j|k}(\varepsilon) = \mathbb{R}^{n_x}$ for $k \in \mathbb{N}$, there always exists a feasible $\mathcal{E}_{k+1} \geq 0$ large enough such that $\hat{\mathbb{X}}_j \subseteq \hat{\mathbb{X}}_{j|k+1}(\varepsilon_{k+1+j|k+1})$. It follows directly from the proof of Thm. 1 that $\hat{x}_{k+1|k+1} := \bar{x}_{k+1|k+1}$

and $\hat{U}_{k+1} := \bar{U}_{k+1}$ satisfy (5). Hence, there exists a feasible solution $(\bar{x}_{k+1|k+1}, \bar{U}_{k+1}, \bar{U}_{k+1}, \mathcal{E}_{k+1})$ for x_{k+1} . ■

Note that the theoretical results of this general CMPC framework can be leveraged for various choices of the RMPC and LB-MPC as (2) – (5) are formulated on a high, conceptual level. Below, we demonstrate a specific implementation of CMPC for safe learning. Firstly, we introduce a relevant case study in the field of automated driving.

V. AUTOMATED LANE MERGING APPLICATION

We consider a lane merging application, in which the ego vehicle should execute the following scenario, depicted in Fig. 2: (i) Initially, the ego vehicle (Agent 1, gray vehicle) is driving in the ego lane. The target vehicle (Agent 2, white vehicle) is driving in the target lane, driving with an unknown but bounded acceleration. (ii) The ego lane is closing off, requiring Agent 1 to merge into the target lane, before the merging point *MP*. (iii) After the merge, Agent 1 maintains a safe distance to Agent 2. When Agent 1 is in front, Agent 2 maintains a safe distance to Agent 1. During the scenario, it is assumed that the paths of Agent 1 and 2 are fixed (but not the speed along the path); and the current position and velocity of Agent 2 are known, but its current and future acceleration profile are unknown.

The dynamics of Agent 1 and 2 are formulated in terms of a one-dimensional path coordinate s_k^i and the forward velocity v_k^i for $i \in \{1, 2\}$, as well as the relative distance $\Delta s_k = s_k^2 - s_k^1$ and the relative velocity $\Delta v_k = v_k^2 - v_k^1$ of Agents 1 and 2. Here, $s_k^i < 0$ and $s_k^i > 0$ indicate a position before and after the *MP*, respectively, as depicted in Fig. 2. The acceleration of Agent 1 u_k^1 is controlled by the CMPC, while the acceleration of Agent 2 u_k^2 is modeled as an unknown, bounded disturbance. The discrete time $k \in \mathbb{N}$ corresponds to real time $t = kT_s$ with the sampling period T_s . We select the state as $x_k = [\Delta s_k \quad \Delta v_k \quad s_k^1 \quad v_k^1]^\top \in \mathbb{R}^4$ with

$$x_{k+1} = Ax_k + B_1 u_k^1 + B_2 u_k^2, \quad \text{where} \quad (6)$$

$$A = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} -\frac{1}{2}T_s^2 \\ -T_s \\ \frac{1}{2}T_s^2 \\ T_s \end{bmatrix} \quad \text{and} \quad B_2 = \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \\ 0 \\ 0 \end{bmatrix}.$$

We assume that the process noise $v_k = 0$ and the bounded disturbance $w_k = B_2 u_k^2 = g(x_k) \in \mathbb{W}$, $u_k^2 \in [u_{\min}^2, u_{\max}^2]$ is a state-dependent policy for Agent 2 and $v_{\min}^2 \leq v_k^2 \leq v_{\max}^2$ for $k \in \mathbb{N}$. The acceleration and velocity of Agent 1 are bounded, yielding the input constraint set $\mathbb{U} := [u_{\min}^1, u_{\max}^1]$ and $0 \leq v_k^1 \leq v_{\max}^1$ for all $k \in \mathbb{N}$. Finally, there is a smooth state-dependent safety distance function D_{safe} , which depends on the relative distance Δs_k , velocity v_k^1 and position s_k^1 of Agent 1, that ensures a safe distance to Agent 2:

$$D_{\text{safe}}(s_k^1, \Delta s_k, v_k^1) \leq 0, \quad k \in \mathbb{N}. \quad (7)$$

We refer to [16] for details on the design of the safety distance function D_{safe} . This leads to the state constraint set

$$\mathbb{X} := \{x \in \mathbb{R}^4 \mid 0 \leq v^1 \leq v_{\max}^1, D_{\text{safe}}(s^1, \Delta s, v^1) \leq 0\}. \quad (8)$$

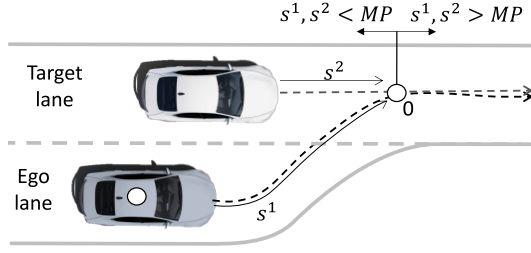


Fig. 2. Set-up of the lane-merging scenario.

The objective is to use CMPC to control Agent 1, completing steps (i) to (iii) of the scenario, learning the acceleration policy g of Agent 2 and adhering to all the safety constraints.

VI. CMPC FOR AUTOMATED LANE MERGING

In this section, we apply the CMPC framework proposed in Sec. III to the automated lane merging problem (see Sec. V) by combining the RMPC from [16] for safety guarantees and a GP-MPC as in [17] for performance. We use the soft-constrained LB-MPC variant of CMPC described in (5).

A. Robust MPC - Contingency Horizon

For the contingency horizon, we use a classical robust MPC with a nominal prediction model (2b) as

$$\bar{x}_{k+1|k} = f(\bar{x}_{k|k}, \bar{u}_{k|k}^1) = A\bar{x}_{k|k} + B_1\bar{u}_{k|k}^1, \quad (9)$$

predicted state $\bar{x}_{k|k} := [\Delta\bar{s}_{k|k} \ \Delta\bar{v}_{k|k} \ \bar{s}_{k|k}^1 \ \bar{v}_{k|k}^1]^\top$ and $\kappa(x, \bar{x}) = 0$ for all x, \bar{x} and $X = \{0\}$. We construct the tightened constraint set $\bar{\mathbb{X}}_j := \mathbb{X} \ominus \sum_{i=0}^{j-1} A^i \mathbb{W}$, for $j = 1, 2, \dots, N-1$ in (2h) (see details in [16]), where the Pontryagin set difference of $\mathbb{A} \subseteq \mathbb{R}^n$ and $\mathbb{B} \subseteq \mathbb{R}^n$ is $\mathbb{A} \ominus \mathbb{B} := \{a \mid \forall b \in \mathbb{B}, a + b \in \mathbb{A}\}$. We consider two disjoint robust control invariant sets Ω_1 (for merging behind Agent 2) and Ω_2 (for merging in front of Agent 2) from [16]. The terminal constraint set of the contingency horizon $\bar{\mathbb{X}}_N$ in (2h) is defined as the union of Ω_1 and Ω_2 , such that $\bar{\mathbb{X}}_N = \Omega_1 \cup \Omega_2$. In [16], the details of Ω_1 and Ω_2 are given and it is shown that the adopted constraint tightening and choice for $\bar{\mathbb{X}}_N$ leads to the robust recursive feasibility of the RMPC as in Ass. 1.

B. Gaussian Process MPC - Performance Horizon

The prediction model (2c) of the performance horizon consists of the nominal dynamics f as in (9) and augmented dynamics d to account for the unmodeled dynamics $w_k = B_2 u_k^2 = g(x_k)$, using GP regression [19] and Bayesian inference based on collected observations of the uncertainty

$$y_k = B_2^\dagger (x_{k+1} - f(x_k, u_k^1)) = u_k^2, \quad (10)$$

and the regressor $\mathbf{z}_k = x_k$, where B_2^\dagger denotes the Moore-Penrose pseudo-inverse of B_2 . We use a data set $\mathcal{D} = \{\mathbf{Z}, \mathbf{y}\}$ with the collected training data $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{n_D}] \in \mathbb{R}^{n_z \times n_D}$ and $\mathbf{y} = [y_1, y_2, \dots, y_{n_D}] \in \mathbb{R}^{1 \times n_D}$. At each time k , we append the data set with the collected training data: $\mathcal{D}_{k+1} = \mathcal{D}_k \cup \{\mathbf{z}_k, y_k\}$. We use a GP $\tilde{d} : \mathbb{R}^{n_z} \rightarrow \mathbb{R}$ to approximate the policy of Agent 2:

$$B_2^\dagger g(x) \approx \tilde{d}(\mathbf{z}) \sim \mathcal{N}(d(\mathbf{z}), \Sigma^d(\mathbf{z})), \quad (11)$$

where $d(\mathbf{z})$ and $\Sigma^d(\mathbf{z})$ denote the mean and covariance function of the posterior of the GP at a given test point \mathbf{z} :

$$d(\mathbf{z}) = \mathbf{k}_{\mathbf{z}\mathbf{z}} K_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{y}, \quad \Sigma^d(\mathbf{z}) = k(\mathbf{z}, \mathbf{z}) - \mathbf{k}_{\mathbf{z}\mathbf{z}} K_{\mathbf{z}\mathbf{z}}^{-1} \mathbf{k}_{\mathbf{z}\mathbf{z}} \quad (12)$$

where k denotes the user-defined kernel function, $\mathbf{k}_{\mathbf{z}\mathbf{z}} \in \mathbb{R}^{n_D}$ is the concatenation of the kernel function evaluated at the test point \mathbf{z} and the training set \mathbf{Z} , where $[\mathbf{k}_{\mathbf{z}\mathbf{z}}]_i = k(\mathbf{z}_i, \mathbf{z})$ and $\mathbf{k}_{\mathbf{z}\mathbf{z}}^\top = \mathbf{k}_{\mathbf{z}\mathbf{z}}$, and $K_{\mathbf{z}\mathbf{z}} \in \mathbb{R}^{n_D \times n_D}$ is a Gram matrix which satisfies $[K_{\mathbf{z}\mathbf{z}}]_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$. In this work, we employ a squared exponential kernel function:

$$k(\mathbf{z}, \mathbf{z}') = \sigma_d^2 \exp\left(-\frac{1}{2}(\mathbf{z} - \mathbf{z}')^\top L_d^{-2}(\mathbf{z} - \mathbf{z}')\right), \quad (13)$$

where σ_d denotes the prior covariance and L_d is the length-scale matrix. The GP formulation in (11) is generally intractable for multi-step predictions. To overcome this issue, we make several commonly used approximations in GP-MPC. Similarly to [3], (i) we assume that the predicted state \hat{x} and the posterior GP d are jointly Gaussian, and (ii) the consecutive GP evaluations are independent. In addition, (iii) we approximate the consecutive means, covariances and cross-covariances of the GP with a first-order Taylor approximation. (iv) We use a Sparse Pseudo-Input GP with M inducing points that are equally spaced over the prediction horizon. For further details, we refer the reader to [17]. The resulting prediction model (2c) of the performance horizon is composed of the mean and covariance of the posterior GP:

$$\hat{x}_{k+j+1|k} = A\hat{x}_{k+j|k} + B_1\hat{u}_k^1 + B_2d(\hat{x}_{k+j|k}, \mathcal{D}_k), \quad (14a)$$

$$\Sigma_{k+j+1|k}^x = \begin{bmatrix} A & B_2 \end{bmatrix} \Sigma_{k+j|k} \begin{bmatrix} A & B_2 \end{bmatrix}^\top, \quad (14b)$$

for $j = 0, 1, \dots, N-1$, where the initial prediction equals the current state $\hat{x}_{0|k} = x_k$ with $\Sigma_{0|k}^x = \mathbf{0}$, and Σ is the predicted joint covariance matrix of the state x and $\tilde{d}(\mathbf{z})$ [17].

We use the GP prediction model to obtain an estimate of the lower and upper bound of Δs based on the predicted sequence \hat{U}_k and the observed data set \mathcal{D}_k . In the CMPC (5), we predict the future states through the mean function (14a) and use the state covariance (14b) to adapt the set $\bar{\mathbb{X}}_{j|k}$. This constraint is softened in the performance horizon by an ℓ_1 penalty method:

$$\bar{\mathbb{X}}_{j|k} := \{x_{k+j|k} \in \mathbb{R}^4 \mid 0 \leq v_{k+j|k}^1 \leq v_{\max}, \quad (15)$$

$$D_{\text{safe}}(s_{k+j|k}^1, \Delta s_{k+j|k} \pm 2\sigma_{s^2, k+j|k}, v_{k+j|k}^1) \leq \varepsilon_{k+j|k}\},$$

where $\sigma_{s^2, k+j}$ denotes the approximate standard deviation of the position of Agent 2 at prediction step $k+j$.

C. Integration into CMPC Framework

We choose the primary cost function J in (5a) as

$$\begin{aligned} J(x_k, \bar{x}_{k|k}, \bar{U}_k, \hat{U}_k) &:= PH(x_k, \bar{U}_k) + (1-P)H(x_k, \hat{U}_k), \\ H(x_k, U_k) &= \sum_{j=0}^N \|v_{\text{ref}}^1 - v_{k+j|k}^1\|_Q^2 + \sum_{j=0}^{N-1} \|u_{k+j|k}^1\|_R^2 \\ &\quad + \|\Delta u_{k+j|k}^1\|_S^2, \end{aligned} \quad (16)$$

where $\|z\|_Q^2 = z^\top Q z$, and $Q, R, S > 0$ are tuning parameters, $\Delta u_{k+j|k}^1 := u_{k+j|k}^1 - u_{k+j-1|k}^1$ is the change in control input with $u_{k-1|k}^1 := u_{k-1}^1$, and v_{ref}^1 denotes the desired velocity

of Agent 1. The parameter $P \in [0, 1]$ balances the cost for the contingency and the performance horizon. Further, we use an ℓ_1 penalty function $J_e(\mathcal{E}) = \rho \|\mathcal{E}\|_1$ in (5a) with a penalty weight $\rho > 0$. While the GP-MPC is soft constrained over the performance horizon, the recursive feasibility and closed-loop safety of the CMPC are ensured by Thm. 2.

VII. NUMERICAL RESULTS

The behavior of the CMPC detailed in Sec. V-VI is verified in a numerical case study, which compares CMPC with the lane-merging controllers using standalone RMPC and GP-MPC, and shows the benefits of CMPC. The following parameters are used in the case study¹: $v_{\text{ref}}^1 = \frac{50}{3.6} \frac{\text{m}}{\text{s}}$, $v_{\text{max}} = 1.1 \cdot v_{\text{ref}}^1$, $T_s = 0.25 \text{ s}$, $N = 20$, $Q = 10$, $R = 1$, $S = 10$, $\rho = 10^4$, $u_{\text{min}}^1 = -3 \frac{\text{m}}{\text{s}^2}$, $u_{\text{max}}^1 = 5 \frac{\text{m}}{\text{s}^2}$, $u_{\text{min}}^2 = -0.5 \frac{\text{m}}{\text{s}^2}$, $u_{\text{max}}^2 = 0.5 \frac{\text{m}}{\text{s}^2}$, $v_{\text{min}}^2 = \frac{25}{3.6} \frac{\text{m}}{\text{s}}$, $s_0^1 = -200 \text{ m}$, $s_0^2 = -180 \text{ m}$, $P = 0.5$, $M = 4$, $\sigma_d = 0.7$ and $L_d = \text{diag}(5, 100, 500, 100)$. Parameters related to the safety function D_{safe} are the same as in [16]. The GP-MPC starts with $n_D = 0$ data points and the size of \mathcal{D}_k increments at every time step. Agent 2's acceleration u_k^2 adheres to the state-dependent cooperative driving policy

$$u_k^2 = \min(\max(k_v(v_{\text{ref}}^2 - v_k^2) + u_{\Delta s}, u_{\text{min}}^2), u_{\text{max}}^2), \quad (17)$$

with a deceleration dependent on the relative distance where

$$u_{\Delta s} = \begin{cases} u_{\Delta s}^*, & \text{if } s_k^2 \geq -200, s_k^1 < 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$u_{\Delta s}^* = \begin{cases} k_{\Delta s}(\Delta s_{\text{ref}} - \Delta s_k), & \text{if } \Delta s_{\text{react}} \geq \Delta s_k \geq 0, \\ k_{\Delta s}(-\Delta s_{\text{ref}} - \Delta s_k), & \text{if } -\Delta s_{\text{react}} \leq \Delta s_k \leq 0, \\ 0, & \text{otherwise,} \end{cases}$$

with a reference velocity $v_{\text{ref}}^2 = v_0^2$, reference following distance $\Delta s_{\text{ref}} = 10 \text{ m}$, reaction distance $\Delta s_{\text{react}} = 10 \text{ m}$, velocity gain $k_v = 1.0461$ and a relative distance gain $k_{\Delta s} = 0.4472$.

A. Merging scenario comparison

We consider a scenario in which Agent 1 is driving at $v_0^1 = \frac{46}{3.6} \frac{\text{m}}{\text{s}}$ behind Agent 2, driving at $v_0^2 = \frac{35}{3.6} \frac{\text{m}}{\text{s}}$. The results are shown in Fig. 3. In all MPCs, Agent 1 first accelerates towards the desired velocity v_{ref}^1 . Then, Agent 1 needs to decide to merge in front or behind Agent 2. The LB-MPCs use an advanced prediction model to have a better understanding of the environment compared to the nominal model of the RMPC. The proposed CMPC framework leverages this LB-MPC to make more informed decisions through the cost function (i.e., select better control inputs) than the standalone RMPC. In case of GP-MPC and CMPC, the GP estimates based on past observations that Agent 2 will decelerate when Agent 1 tries to merge in front. Consequently, the GP-MPC and CMPC policies control Agent 1 to merge in front. Hence, the GP-MPC and CMPC show more assertive behavior and improved performance over the standalone RMPC. Around 9.25 s, the CMPC slightly accelerates as the MP enters the prediction horizon, to ensure a sufficient safety distance to Agent 2 and retain the feasibility of the CMPC, see Fig. 3.

However, the RMPC does not consider the interactions with Agent 2 and shows more conservative behavior, resulting in a merge behind Agent 2. All MPCs maintain sufficient distance towards Agent 2. Finally, Agent 1 converges to either v_{ref}^1 or to Agent 2's velocity. In summary, the CMPC is less conservative than RMPC by leveraging the GP while, as opposed to GP-MPC, preserving robust recursive feasibility and thus robust safety guarantees. The CMPC results are obtained with a maximum computation time of 2.517 s and average computation time of 0.699 s per MPC iteration on a standard laptop PC¹, showing that real-time tractability is reachable with code optimization and high-performance hardware.

B. Sensitivity analysis

We further verify the CMPC (5) by comparing it to the standalone RMPC and GP-MPC with respect to safety and performance in a larger sensitivity analysis, based on the following key performance indicators (KPIs) averaged over $n = 231$ scenarios and $m = 161$ time steps:

- 1) **Slack**: The amount of constraint violation measured by the slack over the complete performance horizon $\frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{k=0}^{m-1} \|\mathcal{E}_k^i\|_1$.
- 2) **Cost**: The merging progress and effort measured by the closed-loop stage cost $\frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{k=0}^{m-1} Q(v_{\text{ref}}^1 - v_{k,i}^1)^2 + R(u_{k,i}^1)^2 + S(\Delta u_{k,i}^1)^2$, with $u_{-1,i}^1 := 0$.
- 3) **Merging time**: Performance measured with the merging time, the time when Agent 1's position is $s^1 > 0 \text{ m}$.
- 4) **Result**: Number of scenarios in which Agent 1 merges in front or behind Agent 2 as a measure of assertiveness.

In the sensitivity analysis, the CMPC, RMPC and GP-MPC are tested for the parameterization defined at the beginning of this section, Agent 2's control policy in (17) and initial conditions $v_0^1 \in \{40, 41, \dots, 50\} \frac{\text{km}}{\text{h}}$ and $v_0^2 \in \{30, 31, \dots, 50\} \frac{\text{km}}{\text{h}}$. The resulting KPIs are shown in Table I. In this study, the CMPC is more assertive than RMPC with 88 vs. 72 front merges (22% increase) while having a reduced cost of 2.14 vs. 2.89 (26% reduction) with comparable merging times. The performance when merging behind Agent 2 is comparable for RMPC and CMPC. However, in some cases, Agent 1 optimistically tries to merge in front, but eventually has to merge behind to guarantee safety, causing the CMPC to incur a higher cost. Both the RMPC and CMPC never exceed the safety constraint in closed loop due to their recursive feasibility properties. The GP-MPC is the least conservative in the sense that it merges in front more often than RMPC and CMPC. Similarly, when GP-MPC merges behind, the cost is lowest as it takes less caution than RMPC and CMPC. However, the lack of safety guarantees leads to an increased use of soft constraints, and hence, constraint violation. The GP-MPC exceeds the safety constraint by at most 0.98 m. However, no collision occurred due to the presence of a safety margin in D_{safe} (see [16]). The CMPC combines the strengths of GP-MPC and RMPC, by learning the residual dynamics in (1), hence, improving performance while remaining safe.

Note that for 12 and 11 initial conditions for the RMPC and CMPC, respectively, the robust controlled invariant set $\bar{\mathbb{X}}_N$ is not reachable from the initial condition, and, hence, safety cannot be guaranteed under all possible disturbances. As a

¹The numerical experiments are performed in Matlab R2023a with IPOPT 3.14.11 [20] and CasADi 3.6.5 [21] on an Intel Core i7-9750H at 2.60 GHz.

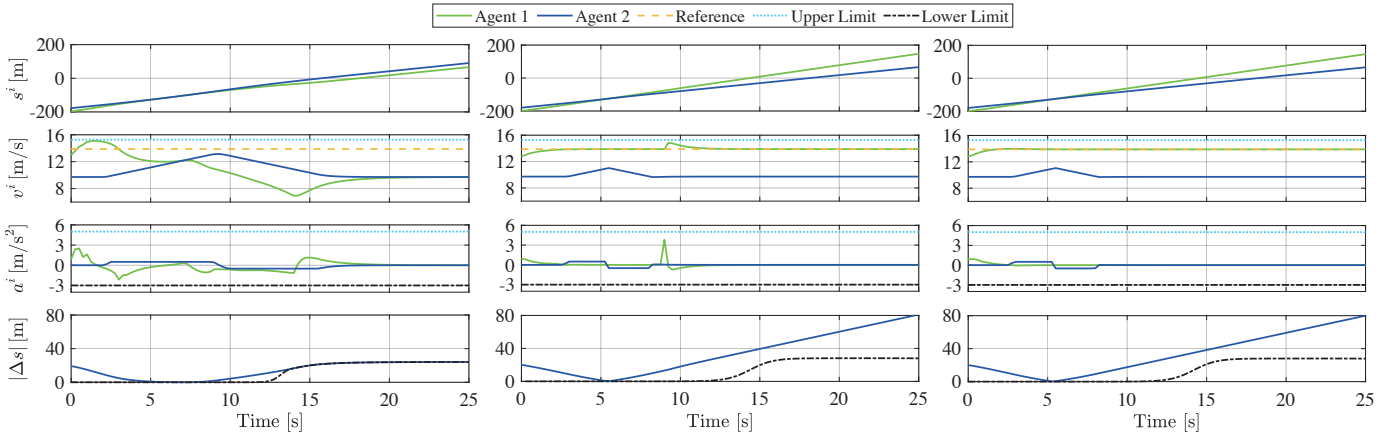


Fig. 3. Comparison of RMPC (left), CMPC (middle) and GP-MPC (right) for one lane merging scenario, see Sec. VII-A. Shown are the trajectories of the path coordinate s^i , velocity v^i , acceleration a^i and relative distance $|\Delta s|$ of Agent 1 and 2. The lower limit of $|\Delta s|$ is imposed by D_{safe} .

TABLE I

KPIs FOR THE RESULTS FRONT (F) AND BEHIND (B). NOTE THAT RMPC AND CMPC HAVE 12 AND 11 INFEASIBLE INITIAL STATES, RESPECTIVELY, THAT ARE EXCLUDED OF THESE KPIs.

MPC	Number [-]		Slack [-]		Cost [-]		Merging Time [s]	
	F	B	F	B	F	B	F	B
RMPC	72	147	0	0	2.89	62.61	14.42	16.88
CMPC	88	132	10^{-7}	10^{-6}	2.14	65.72	14.82	16.94
GP-MPC	153	78	1.36	10^{-7}	2.41	38.71	14.74	16.93

consequence, these numerical experiments were infeasible and are therefore excluded from Table I. The soft-constrained GP-MPC can handle these scenarios by using the slack variables at the risk of a potential collision. Providing formal safety guarantees may come at a cost of excluding certain initial conditions in which safety cannot be guaranteed. However, the CMPC framework can be used to analyze and flag these scenarios, either a priori or on-line.

VIII. CONCLUSIONS

To address the challenge of controlling safety-critical systems, this research focuses on achieving both assertive and flexible learning behavior while ensuring safety guarantees – a concept often referred to as “safe learning.” To this end, it introduces a framework that combines general robust MPC (RMPC) algorithms with general learning-based MPC (LB-MPC) algorithms. It is proven that this CMPC framework inherits the robust recursive feasibility properties of the RMPC while realizing improved performance by learning from data. The proposed CMPC framework is applied to an automated driving use case of lane merging. This case study shows that CMPC has improved driving performance due to the presence of the LB-MPC (i.e., merging in front more often, by learning the behavior of the other vehicle) compared to RMPC, while preserving the safety of the RMPC.

REFERENCES

- [1] M. Nezami, N.T. Nguyen, G. Mannel, H.S. Abbas, and G. Schildbach, “A safe control architecture based on robust model predictive control for autonomous driving,” American Control Conference, 2022.
- [2] I. Alvarado, P. Krupa, D. Limon, and T. Alamo, “Tractable robust MPC design based on nominal predictions,” J. Process Control, 2022.
- [3] L. Hewing, J. Kabzan and M. N. Zeilinger, “Cautious Model Predictive Control Using Gaussian Process Regression,” in IEEE Transactions on Control Systems Technology, 2020.,
- [4] L. Hewing, K.P. Wabersich, M. Menner, and M.N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” Annual Review of Control, Robotics, and Autonomous Systems, 2020.
- [5] M. Zanon and S. Gros, “Safe reinforcement learning using robust MPC,” IEEE Transactions on Automatic Control, 2021.
- [6] K.P. Wabersich and M.N. Zeilinger, “Linear Model Predictive Safety Certification for Learning-Based Control,” IEEE Conference on Decision and Control, 2018.
- [7] A. Aswani, H. Gonzalez, S.S. Sastry, and C. Tomlin, “Provably safe and robust learning-based model predictive control,” Automatica, 2013.
- [8] T. Brudigam, M. Olbrich, D. Wollherr, and M. Leibold, “Stochastic model predictive control with a safety guarantee for automated driving,” IEEE Transactions on Intelligent Vehicles, 2023.
- [9] S. Li and O. Bastani, “Robust model predictive shielding for safe reinforcement learning with stochastic dynamics,” IEEE ICRA, 2020.
- [10] A. Mesbah, K.P. Wabersich, A.P. Schoellig, M.N. Zeilinger, S. Lucia and T.A. Badgwell, “Fusion of Machine Learning and MPC under Uncertainty: What Advances Are on the Horizon?,” American Control Conference, 2022.
- [11] J.P. Alsterda, M. Brown, and J.C. Gerdes, “Contingency model predictive control for automated vehicles,” American Control Conference, 2019.
- [12] R. Soloperto, J. Kohler, and F. Allgower, “Augmenting MPC schemes with active learning: Intuitive tuning and guaranteed performance,” IEEE Control Systems Letters, 2020.
- [13] K.S. Schweidel, S.M. Koehler, V.R. Desaraju, and M. Baric, “Driver-in-the-loop contingency MPC with invariant sets,” European Control Conference, 2022.
- [14] J. Dallas, J. Wurts, J. L. Stein, and T. Ersal, “Contingent nonlinear model predictive control for collision imminent steering in uncertain environments,” IFAC-PapersOnLine, 2020.
- [15] X. Chen and J. Mårtensson, “Invariant safe contingency model predictive control for intersection coordination of mixed traffic,” IEEE Intelligent Transportation Systems Conference, 2023.
- [16] M.E. Geurts, A. Katriniok, E. Silvas, N.J. Brouwer and W.P.M.H. Heemels. “Automated Lane Merging with Safety Guarantees: A Robust Model Predictive Control Approach with Experimental Validation,” Techrxiv, 2025.
- [17] T.M.J.T. Baltussen, E. Lefeber, R. Tóth, W.P.M.H. Heemels and A. Katriniok, “Online learning of interaction dynamics with dual model predictive control for multi-agent systems using Gaussian processes,” American Control Conference 2025, arXiv:2409.00432.
- [18] J.B. Rawlings, D.Q. Mayne and M. Diehl, “Model Predictive Control: Theory, Computation, and Design”, Nob Hill Publishing, 2017.
- [19] C.E. Rasmussen and C.K. Williams, “Gaussian processes for machine learning”, MIT Press, 2006.
- [20] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Mathematical Programming, 2006.
- [21] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” Mathematical Programming Computation, 2019.