Automatica 123 (2021) 109304

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Model predictive control for drift counteraction of stochastic constrained linear systems☆

Robert A.E. Zidek [a,*], Ilya V. Kolmanovsky [a], Alberto Bemporad [b]

[a] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, United States of America
[b] IMT – Institute for Advanced Studies, Lucca, Italy

ARTICLE INFO

Article history:
Received 29 November 2019
Received in revised form 3 May 2020
Accepted 23 August 2020
Available online xxxx

Keywords:
Stochastic control
Mixed-integer programming
Model predictive control

ABSTRACT

The paper presents a stochastic MPC (SMPC) formulation suitable for maximizing the average time until a discrete-time linear system with additive random disturbance violates prescribed constraints. The SMPC procedure is based on a scenario tree that encodes the most likely system behavior for a given tree density, where each branch of the tree represents a specific evolution of the system that occurs with a certain probability. A mixed-integer linear program (MILP) is developed that maximizes the average time until constraint violation for a given scenario tree. Feedback is provided by reconstructing the scenario tree and recomputing the MILP solution over a receding time horizon based on the current state of the system. The average time until constraint violation achieved by the SMPC strategy approaches the optimal value as the scenario tree density is increased. Two numerical case studies, including an adaptive cruise control problem, demonstrate the effectiveness of the proposed SMPC strategy compared to dynamic programming solutions.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

This paper is concerned with the control problem of maximizing the time before a specified set of constraints is violated by a given system. Such problems arise in many engineering applications, in particular, those constrained by finite resources (fuel, energy, or, component life) or where limited control authority is available to counteract large persistent disturbances, which is common in underactuated systems as, for example, discussed by Kolmanovsky and Zidek (2018).

In optimal control theory, such problems are known as exit-time problems. The properties of exit-time problems were studied by Barles and Rouy (1998), Bayraktar et al. (2010), Buckdahn and Nie (2016), Crandall et al. (1984), Lions (1983), and Malisoff (2001), and in the references therein. The optimal solution satisfies the Hamilton–Jacobi–Bellman (HJB) equation in the so-called viscosity or weak sense, where the notion of viscosity solutions to partial differential equations was described by Crandall and Lions (1983). However, explicit solutions to the HJB equation only exist for special problems, see, for example, Clark and Vinter (2012) or Kolmanovsky and Maizenberg (2002).

The discrete-time formulation of the problem, on the other hand, is computationally more tractable. In this context, model predictive control (MPC) appears attractive as an effective method for solving high-dimensional constrained control problems. At each sampling time, MPC exploits a solution to an open-loop optimal control problem based on a prediction model of the system and its current state. Only the first element of the solution sequence is applied to the system as the control input. The process is repeated at the next sampling time. This approach provides feedback to compensate for unmodeled effects.

A systematic MPC approach to exit-time problems for deterministic linear systems was developed by Zidek et al. (2017). In this paper the approach is extended to stochastic systems where the objective is to maximize the average time before constraint violation. We focus on linear systems with additive random disturbance, but the developments can readily be applied to exit-time problems for other classes of stochastic linear systems. Initial results were published in the conference paper by Zidek et al. (2018). Compared to the previous publication, in this paper we extend the theoretical analysis of the problem and develop additional theoretical results. Moreover, we provide additional practical considerations and present a comprehensive numerical case study on stochastic adaptive cruise control.

In the uncertain system case, robust MPC techniques may be used where the control law only considers the worst-case

☆ This research was supported by the National Science Foundation Award Number EECS 1404814. The material in this paper was partially presented at the 2018 American Control Conference, June 27–29, 2018, Milwaukee, WI, USA. This paper was recommended for publication in revised form by Associate Editor Alessandro Abate under the direction of Editor Ian R. Petersen.

* Corresponding author.
E-mail addresses: robzidek@umich.edu (R.A.E. Zidek), ilya@umich.edu (I.V. Kolmanovsky), alberto.bemporad@imtlucca.it (A. Bemporad).

https://doi.org/10.1016/j.automatica.2020.109304
0005-1098/© 2020 Elsevier Ltd. All rights reserved.

disturbance scenario; examples of robust MPC are given in the papers by Lee and Yu (1997) and Mayne et al. (2005). On the other hand, a less conservative variant of robust MPC is Stochastic MPC (SMPC) which explicitly accounts for the uncertainty of the disturbances as demonstrated by Cannon et al. (2009), Couchman et al. (2006), Di Cairano et al. (2014), Primbs and Sung (2009), and Mesbah (2016). At the same time, recent developments in hardware and numerical methods may facilitate practical use of SMPC. We further discuss practical considerations and the potential for fast embedded applications in Section 5.2 of this paper.

Similar to the developments by Bernardini and Bemporad (2012), we propose an SMPC scheme that uses a tree structure to encode the most likely disturbance scenarios. A mixed-integer linear program (MILP) is developed that obtains a control policy that maximizes the average time until constraint violation for a given scenario tree. In order to obtain the current control input, the MILP solution is recomputed at each sampling time over a receding time horizon based on the current state and an updated scenario tree which in turn is based on the current disturbance values. The average time until constraint violation for this SMPC law is shown to approach the optimal value as the number of tree nodes goes to infinity.

The paper is structured as follows. In Section 2, we formulate the problem and reason about the existence of a solution. In Section 3, the scenario tree structure is discussed and an algorithm for constructing a scenario tree is presented. The MILP that maximizes the average time until constraint violation for a given scenario tree is developed in Section 4. Based on the MILP, the SMPC strategy is stated in Section 5, which also includes a discussion on the practical applicability of the proposed approach. Two examples, including a car following problem, are presented in Section 6, where we also compare the SMPC solutions against dynamic programming solutions. A conclusion is provided in Section 7.

2. Problem formulation

2.1. Stochastic linear system

We consider a stochastic linear discrete-time system represented by

$$x_{t+1} = A_t x_t + B_t u_t + w_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in U_t \subset \mathbb{R}^p$ denote the state and control input vectors, respectively, at a time instant $t \in \mathbb{Z}_{\geq 0}$, A_t and B_t are time-dependent matrices, and the set U_t represents prescribed constraints on the control input vector at time instant t. The variable w denotes a measured random disturbance that is modeled by a Markov chain. The disturbance can take values in the finite set

$$W = \{w^1, w^2, \ldots, w^{|W|}\}, \tag{2}$$

of cardinality $|W| > 0$.

The approach of modeling the disturbance as a Markov chain with a finite number of states is consistent with the one adopted in the stochastic dynamic programming literature, an overview of which is provided by Puterman (2014). Markov chains are used in a wide area of applications, such as physics, chemistry, biology, manufacturing, and economics. For example, a Markov chain was used by Lin et al. (2004) to model the driver demand for hybrid vehicles, Ikonen et al. (2016) employed a Markov chain model for chemical process control, and Zidek and Kolmanovsky (2017) used Markov chains to model the traffic on a two-lane road.

A key characteristic of a Markov chain is that the next state only depends on the current state, but not on previous transitions that led to the current state. In this paper, we denote the transition probabilities by

$$P_W(w^j|w^i) = P_W(w_{t+1} = w^j | w_t = w^i) \in [0, 1], \tag{3}$$

for all $w^i, w^j \in W$ and $t \in \mathbb{Z}_{\geq 0}$.

2.2. Stochastic optimal control problem

In addition to U_t, we introduce the time-dependent set $G_t \in \mathbb{R}^n$ that represents prescribed state constraints. Furthermore, we denote a control policy by

$$\pi : G_t \times W \times \mathbb{Z}_{\geq 0} \to U_t, \tag{4}$$

for all $t \in \mathbb{Z}_{\geq 0}$, i.e., the control input vector at a time instant t is obtained by $u_t = \pi(x_t, w_t, t)$, where Π is the set of admissible (U_t-valued) control policies. For a given control policy $\pi \in \Pi$ and initial condition $x_0 \in G_0$ and $w_0 \in W$, the random variable τ, also referred to as the first exit-time, denotes the time instant at which constraint violation occurs for the first time,

$$\tau(x_0, w_0, \pi) = \inf\{t \in \mathbb{Z}_{\geq 0} : x_t \notin G_t\}, \tag{5}$$

where x_t is the response of (1) to the initial condition x_0 and w_0 when using the control policy π. Note that the value of τ is random as it depends on the random realization of $\{w_t\}$. The average (i.e., the expected value of the) first exit-time is denoted by

$$\bar{\tau}(x_0, w_0, \pi) = \mathrm{E}\{\tau(x_0, w_0, \pi)\}. \tag{6}$$

Then the optimal control problem of maximizing the average first exit-time is as follows:

$$\max_{\pi \in \Pi} \bar{\tau}(x_0, w_0, \pi). \tag{7}$$

Throughout the paper we make the following assumption about the sets G_t and U_t.

Assumption 1. The sets G_t and U_t are polytopes for all $t \in \mathbb{Z}_{\geq 0}$, where the set of state constraints, G_t, is given by

$$G_t = \{x : C_t x \leq b_t\}. \tag{8}$$

2.3. Existence of maximizing sequence

The following theorem provides conditions under which $\bar{\tau}$ is bounded. We adopt the following assumption in this regard.

Assumption 2. There exists $T > 0$ and $\bar{w} \in W$ such that $\bar{w}$ overpowers any admissible control and the deterministic system,

$$x_{t+1} = A_t x_t + B_t \pi(x_t, \bar{w}, t) + \bar{w},$$

exits G_t in at most T steps for all $x_0 \in G_0$ and $\pi \in \Pi$. In addition, $P_W(\bar{w}|\bar{w}) > 0$ and $\bar{w}$ is accessible from each $w \in W$, meaning $\mathrm{Prob}(w_n = \bar{w}, \text{given } w_0 = w) > 0$, for some $n > 0$.

Theorem 1. *Suppose Assumption 2 holds. Then there exists $\bar{T} > 0$ such that*

$$\bar{\tau}(x, w, \pi) \leq \bar{T},$$

for all $x \in G_0$, $w \in W$, and $\pi \in \Pi$.

Proof. Let $x \in G_0$, $w \in W$, and $\pi \in \Pi$ be a given initial condition and admissible control policy. The corresponding average first exit-time may be expressed as follows:

$$\begin{aligned}\bar{\tau}(x, w, \pi) &= \sum_{i=1}^{\infty} iP(\tau(x, w, \pi) = i) \\ &\leq \sum_{i=1}^{\infty} iP(\tau(x, w, \pi) \geq i),\end{aligned} \tag{9}$$

where $P(\tau(x, w, \pi) = i)$ and $P(\tau(x, w, \pi) \geq i)$ denote the probabilities that the first exit-time is equal to i or greater than or equal to i, respectively. Using Assumption 2 and

$$\begin{aligned}\rho_{\bar{w},T,i} = \text{Prob}(&\bar{w} \text{ occurs } T \text{ times in a row} \\ &\text{prior to } t = i - 1),\end{aligned} \tag{10}$$

$P(\tau(x, w, \pi) \geq i)$ is bounded according to

$$P(\tau(x, w, \pi) \geq i) \leq 1 - \rho_{\bar{w},T,i}, \tag{11}$$

where $\bar{w}$ and T are defined in Assumption 2. Using Assumption 2 (in particular: $\bar{w}$ is accessible from every $w^i \in W$) and denoting

$$\begin{aligned}q_T = (P_W(\bar{w}|\bar{w}))^T \times \text{Prob}(&\bar{w} \text{ is reached from } w \\ &\text{in at most } |W| \text{ steps}),\end{aligned} \tag{12}$$

which is greater than zero due to the accessibility of $\bar{w}$ and $P_W(\bar{w}|\bar{w}) > 0$ by Assumption 2, it follows that

$$\begin{aligned}\rho_{\bar{w},T,i} &\geq \sum_{k=0}^{\left\lfloor \frac{i-1}{T+|W|} \right\rfloor - 1} (1 - q_T)^k q_T \\ &= q_T \left(\frac{1 - (1 - q_T)^{\left\lfloor \frac{i-1}{T+|W|} \right\rfloor}}{1 - (1 - q_T)} \right) \\ &= 1 - (1 - q_T)^{\left\lfloor \frac{i-1}{T+|W|} \right\rfloor},\end{aligned} \tag{13}$$

where $\lfloor \cdot \rfloor$ is the floor operator. Hence, (11) becomes

$$P(\tau(x, w, \pi) \geq i) \leq (1 - q_T)^{\left\lfloor \frac{i-1}{T+|W|} \right\rfloor}, \tag{14}$$

and (9) may be written as follows:

$$\begin{aligned}\bar{\tau}(x, w, \pi) &\leq \sum_{i=1}^{\infty} i(1 - q_T)^{\left\lfloor \frac{i-1}{T+|W|} \right\rfloor} \\ &\leq \sum_{k=0}^{\infty} (k + 1)(T + |W|)^2 (1 - q_T)^k \\ &= (T + |W|)^2 \left(\frac{1 - q_T}{q_T^2} + \frac{1}{q_T} \right) \\ &= \left(\frac{T + |W|}{q_T} \right)^2 = \bar{T}.\end{aligned} \tag{15}$$

Remark 1. Theorem 1 guarantees the existence of a maximizing sequence for all initial conditions, $x \in G_0$ and $w \in W$, i.e., a sequence $\{\pi_n\}$ in Π such that $\bar{\tau}(x, w, \pi_n) \to \sup_{\pi \in \Pi} \bar{\tau}(x, w, \pi)$.

For the following developments, it is assumed that the sequence in Remark 1 converges, meaning a solution exists, for all initial conditions.

Assumption 3. A solution $\pi^* \in \Pi$ (may not be unique) to problem (7) exists for each $x_0 \in G_0$ and $w_0 \in W$.

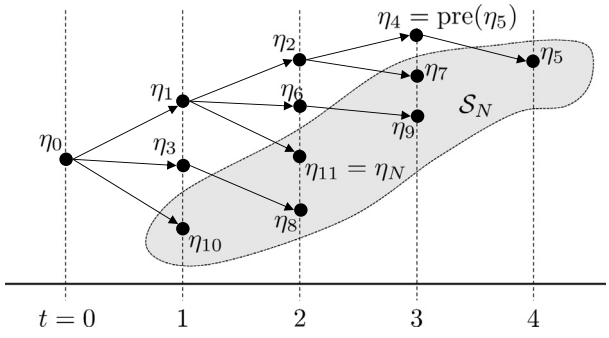

Fig. 1. Scenario tree example for 12 nodes, including $|\mathcal{S}_N| = 6$ leaf nodes.

3. Scenario tree

In order to optimize over a subset of all possible disturbance scenarios, similar to the work by Bernardini and Bemporad (2012), a scenario tree is constructed that contains the most likely disturbance scenarios for a given number of tree nodes. A tree node is denoted by $\eta \in \mathcal{T}_N$, where

$$\mathcal{T}_N = \{\eta_0, \eta_1, \ldots, \eta_N\},$$

denotes a tree with $N + 1$ nodes. The node η_0 is the root node of the tree. The predecessor of a node $\eta \in \mathcal{T}_N$ is given by $\text{pre}(\eta)$. The set of successors of a node $\eta \in \mathcal{T}_N$ is denoted by

$$\text{succ}(\eta) = \{\eta_1^{\text{succ}(\eta)}, \eta_2^{\text{succ}(\eta)}, \ldots, \eta_{|W|}^{\text{succ}(\eta)}\}.$$

The nodes that do not have a successor node in $\mathcal{T}_N$ form the set of leaf nodes, which is given by

$$\mathcal{S}_N = \{\eta \in \mathcal{T}_N : \text{succ}(\eta) \cap \mathcal{T}_N = \emptyset\}.$$

Fig. 1 shows an example scenario tree,

$$\mathcal{T}_{11} = \{\eta_0, \eta_1, \ldots, \eta_{11}\},$$

for a given Markov chain with $|W| = 3$. For example, $\text{succ}(\eta_1) = \{\eta_2, \eta_6, \eta_{11}\}$, in Fig. 1, i.e., $\eta_1^{\text{succ}(\eta_1)} = \eta_2$, $\eta_2^{\text{succ}(\eta_1)} = \eta_6$, and $\eta_3^{\text{succ}(\eta_1)} = \eta_{11}$. The set of leaf nodes in Fig. 1 is given by

$$\mathcal{S}_{11} = \{\eta_5, \eta_7, \eta_8, \eta_9, \eta_{10}, \eta_{11}\}.$$

With each $\eta \in \mathcal{T}_N$, we associate a disturbance w_η as well as a state vector x_η, control input u_η, and time instant t_η, where $w_{\eta_0} = w_0$, $x_{\eta_0} = x_0$, and $t_{\eta_0} = 0$ for the root node. Moreover, for each $\eta \in \mathcal{T}_N \setminus \{\eta_0\}$, x_η satisfies the dynamics in (1). Consequently,

$$x_\eta = A_{t_{\text{pre}(\eta)}} x_{\text{pre}(\eta)} + B_{t_{\text{pre}(\eta)}} u_{\text{pre}(\eta)} + w_{\text{pre}(\eta)}. \tag{16}$$

The probability of reaching a node $\eta \in \mathcal{T}_N$, starting from the root node, is given by

$$\rho_\eta = \rho_{\text{pre}(\eta)} P_W(w_\eta | w_{\text{pre}(\eta)}) \in [0, 1], \tag{17}$$

where $\rho_{\eta_0} = 1$. Algorithm 1 implements the scenario tree generation suitable for either offline or online use. The set $\mathcal{C}$ contains the candidate nodes that are considered when adding a node to the tree. At each iteration, the node $\eta \in \mathcal{C}$ with the greatest probability ρ_η is chosen from the set of candidate nodes, and the successors of η are added to the list of candidate nodes. Thus, the tree is intended to capture most likely scenarios subject to the total number of nodes constrained to be $N + 1$.

Algorithm 1 Design of scenario tree $\mathcal{T}_N$

$\mathcal{T}_N \leftarrow \{\eta_0\}$; $\mathcal{C} \leftarrow \emptyset$; $\rho_{\eta_0} \leftarrow 1$
$t_{\eta_0} \leftarrow 0$; $x_{\eta_0} \leftarrow x_0$; $w_{\eta_0} \leftarrow w_0$
$i \leftarrow 0$
while $i < N$ **do**
  for $j \in \{1, 2, \dots, |W|\}$ **do**
    $w_{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow w^j$ $(w^j \in W)$
    $t_{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow t_{\eta_i} + 1$
    $\rho_{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow \rho_{\eta_i} P_W(w^j | w_{\eta_i})$
  end for
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathrm{succ}(\eta_i)$
  $\eta_{i+1} \leftarrow \arg\max_{\eta \in \mathcal{C}} \rho_\eta$ (pick any maximizer)
  $\mathcal{T}_N \leftarrow \mathcal{T}_N \cup \{\eta_{i+1}\}$
  $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\eta_{i+1}\}$
  $i \leftarrow i + 1$
end while

In general, a scenario tree $\mathcal{T}_N$ contains $|\mathcal{S}_N| \geq 1$ unique disturbance trajectories/scenarios that are denoted by

$$\begin{aligned}\{w_t\}_\eta &= \{w_t : t \in \mathbb{Z}_{[0,t_\eta]}\}_\eta \\ &= (w_0, \dots, w_{\mathrm{pre}(\mathrm{pre}(\eta))}, w_{\mathrm{pre}(\eta)}, w_\eta),\end{aligned} \tag{18}$$

for each leaf node $\eta \in \mathcal{S}_N$. For example, $\{w_t\}_{\eta_9} = (w_0, w_{\eta_1}, w_{\eta_6}, w_{\eta_9})$ in Fig. 1.

For a given tree $\mathcal{T}_N$ with initial $x_0 \in G_0$ and $w_0 \in W$ and control policy $\pi_N \in \Pi$, the deterministic first exit-time corresponding to the disturbance trajectory $\{w_t\}_\eta$, see (18), is defined by

$$\begin{aligned}\tau_{N,\eta}(x_0, w_0, \pi_N) = \min\{&\min\{t \in \mathbb{Z}_{[0,t_\eta]} : x_t \notin G_t\} \\ &\cup \{t_\eta + 1\}\},\end{aligned} \tag{19}$$

for each $\eta \in \mathcal{S}_N$, where x_t is the deterministic response of (1) under $\{w_t\}_\eta$ when using the control policy $\pi_N \in \Pi$. Note that for some $\{w_t\}_\eta$, x_t may not exit G_t for $t \in \mathbb{Z}_{[0,t_\eta]}$; in this case, $\tau_{N,\eta}(x_0, w_0, \pi_N) = t_\eta + 1$ in line with (19). The average first exit-time for a given scenario tree $\mathcal{T}_N$ and a control policy $\pi_N \in \Pi$ is given by

$$\bar{\tau}_N(x, w, \pi_N) = \sum_{\eta \in \mathcal{S}_N} \tau_{N,\eta}(x, w, \pi_N)\rho_\eta. \tag{20}$$

In analogy to problem (7), the optimal control problem of maximizing the average first exit-time over a subset of disturbance scenarios defined by tree $\mathcal{T}_N$ can be expressed as

$$\max_{\pi_N \in \Pi} \bar{\tau}_N(x, w, \pi_N). \tag{21}$$

The following two sets are defined:

$$\begin{aligned}\mathcal{H}_{N,\eta} = \{&\eta_0, \dots, \mathrm{pre}(\mathrm{pre}(\eta)), \\ &\mathrm{pre}(\eta), \eta\}, \text{ for all } \eta \in \mathcal{S}_N,\end{aligned} \tag{22}$$

$$\mathcal{K}_{N,\xi} = \{\eta \in \mathcal{S}_N : \xi \in \mathcal{H}_{N,\eta}\}, \text{ for all } \xi \in \mathcal{T}_N. \tag{23}$$

$\mathcal{H}_{N,\eta}$ is the set of nodes of the disturbance scenario associated with leaf node $\eta \in \mathcal{S}_N$. $\mathcal{K}_{N,\xi}$ is the set of leaf nodes whose associated disturbance scenarios contain the node $\xi \in \mathcal{T}_N$. For example, in Fig. 1,

$\mathcal{H}_{11,\eta_7} = \{\eta_0, \eta_1, \eta_2, \eta_7\}$ and $\mathcal{K}_{11,\eta_1} = \{\eta_5, \eta_7, \eta_9, \eta_{11}\}$.

Moreover, for a given control policy $\pi \in \Pi$ and scenario tree $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, with initial condition $x = x_0 \in G_0$ and $w = w_0 \in W$, the set of leaf nodes $\eta \in \mathcal{S}_N$ with associated first exit-time $\tau_{N,\eta}(x, w, \pi) = i \in \mathbb{Z}_+$ is given by

$$\mathcal{Z}_N(\pi, i) = \{\eta \in \mathcal{S}_N : \tau_{N,\eta}(x, w, \pi) = i\}. \tag{24}$$

To simplify the notations, we drop the dependence on x and w on the left hand side.

The next result (Theorem 2) shows that, in terms of the average first exit-time, a solution to problem (21) is arbitrarily close to a solution (if one exists) of problem (7) for sufficiently large N. Theorem 2 is based on Lemma 1.

Lemma 1.

$$\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) = \bar{\tau}(x, w, \pi), \tag{25}$$

for all $x \in G_0$, $w \in W$, and $\pi \in \Pi$.

Proof. Let $\pi \in \Pi$ be a given control policy and $x \in G_0$ and $w \in W$ be a given initial condition. Then, by (20),

$$\begin{aligned}\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) &= \lim_{N\to\infty} \sum_{\eta \in \mathcal{S}_N} \tau_{N,\eta}(x, w, \pi)\rho_\eta \\ &= \lim_{N\to\infty} \left(\sum_{i=1}^{t_N} i \sum_{\eta \in \mathcal{Z}_N(\pi,i)} \rho_\eta \right),\end{aligned} \tag{26}$$

where $t_N = \max\{t_\eta : \eta \in \mathcal{T}_N\} + 1$. Since W is a finite set, it follows from the tree generation procedure (Algorithm 1) that eventually every branch corresponding to non-zero probability of next disturbance value continues. Thus, for each $i \in \mathbb{Z}_+$,

$$\lim_{N\to\infty} \sum_{\eta \in \mathcal{Z}_N(\pi,i)} \rho_\eta = \mathrm{Prob}(\tau(x, w, \pi) = i). \tag{27}$$

Moreover, $t_N \to \infty$ as $N \to \infty$. Consequently, from (26) and (27) it can be shown that

$$\begin{aligned}\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) &= \sum_{i=1}^{\infty} i\mathrm{Prob}(\tau(x, w, \pi) = i) \\ &= \bar{\tau}(x, w, \pi).\end{aligned} \tag{28}$$

Theorem 2. *Suppose Assumption 3 holds. Then, for each $x \in G_0$, $w \in W$, and $\varepsilon > 0$, there exists $\bar{N} > 0$ such that*

$$\max_{\pi_N \in \Pi} \bar{\tau}_N(x, w, \pi_N) + \varepsilon \geq \max_{\pi \in \Pi} \bar{\tau}(x, w, \pi), \tag{29}$$

$$\bar{\tau}(x, w, \pi_N^*) + \varepsilon \geq \max_{\pi \in \Pi} \bar{\tau}(x, w, \pi), \tag{30}$$

where $\pi_N^* \in \arg\max_{\pi_N \in \Pi} \bar{\tau}_N(x, w, \pi_N)$, for all $N \geq \bar{N}$.

Proof. For a given initial condition, $x \in G_0$ and $w \in W$, let $\mathcal{T}_N$ be the scenario tree for a given $N \in \mathbb{Z}_+$. Moreover, let $\pi^* \in \Pi$ be a solution to problem (7), which exists by Assumption 3, and let $\pi_N^* \in \Pi$ be a control policy that maximizes the average first exit-time for $\mathcal{T}_N$ according to problem (21), which exists due to the existence of a solution to (7). It follows that

$$\bar{\tau}_N(x, w, \pi_N^*) \geq \bar{\tau}_N(x, w, \pi^*). \tag{31}$$

The optimal average first exit-time of problem (7) may be written as follows:

$$\bar{\tau}(x, w, \pi^*) = \bar{\tau}_N(x, w, \pi^*) + \bar{\tau}_{\mathrm{Rest},N}(x, w, \pi^*), \tag{32}$$

where $\bar{\tau}_{\mathrm{Rest},N}$ represents the portion of all scenarios not described by $\mathcal{T}_N$. By Lemma 1, $\bar{\tau}_N(x, w, \pi^*)$ approaches $\bar{\tau}(x, w, \pi^*)$ as $N \to \infty$ and thus $\bar{\tau}_{\mathrm{Rest},N}(x, w, \pi^*) \to 0$. This implies that for every $\varepsilon > 0$, there exists $\bar{N} > 0$ such that

$$\bar{\tau}(x, w, \pi^*) \leq \bar{\tau}_N(x, w, \pi^*) + \varepsilon, \tag{33}$$

for all $N \geq \bar{N}$. It follows from (31) and (33) that

$$\bar{\tau}_N(x, w, \pi_N^*) + \varepsilon \geq \bar{\tau}(x, w, \pi^*), \tag{34}$$

for all $N \geq \bar{N}$. In analogy to (32), it follows from adding $\bar{\tau}_{\text{Rest},N}(x, w, \pi_N^*)$ to (34) that

$$\bar{\tau}(x, w, \pi_N^*) + \varepsilon \geq \bar{\tau}(x, w, \pi^*), \tag{35}$$

for all $N \geq \bar{N}$, which proves (30).

4. Mixed-integer linear program

4.1. Formulation

In this section, a mixed-integer linear program (MILP) is proposed that solves problem (21). By Theorem 2, the average first exit-time of a solution to problem (21) is arbitrarily close to the average first exit-time of a solution to problem (7) for a sufficiently large N.

In what follows, for a given tree $\mathcal{T}_N$, a set of control inputs that satisfy the control constraints is denoted by

$$\mathcal{U}_N = \{u_\eta \in U_{t_\eta} \,:\, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}. \tag{36}$$

Likewise, a set of states, x_η, for each node of the tree is denoted by $\mathcal{X}_N$.

Furthermore, for each node $\eta \in \mathcal{T}_N$, we use a binary variable δ_η to indicate the condition that the state constraints are violated, i.e., $x_{t_\eta} \notin G_{t_\eta}$. $\mathcal{D}_N$ denotes a set of δ_η values for a tree $\mathcal{T}_N$,

$$\mathcal{D}_N = \{\delta_\eta \in \{0, 1\} \,:\, \eta \in \mathcal{T}_N\}. \tag{37}$$

The MILP for a given tree $\mathcal{T}_N$ is as follows:

$$\min_{\mathcal{X}_N, \mathcal{U}_N, \mathcal{D}_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_{N,\eta}} \delta_\eta \rho_\xi \quad \text{s.t.} \tag{38a}$$

$$x_\eta = A_{t_{\text{pre}(\eta)}} x_{\text{pre}(\eta)} + B_{t_{\text{pre}(\eta)}} u_{\text{pre}(\eta)} + w_{\text{pre}(\eta)}, \quad \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{38b}$$

$$u_\eta \in U_{t_\eta}, \ \text{for all } \eta \in \mathcal{T}_N \setminus \mathcal{S}_N \tag{38c}$$

$$\delta_\eta \geq \delta_{\text{pre}(\eta)}, \ \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{38d}$$

$$\delta_\eta \in \{0, 1\} \subset \mathbb{Z}, \ \text{for all } \eta \in \mathcal{T}_N \tag{38e}$$

$$C_{t_\eta} x_\eta \leq b_{t_\eta} + \mathbf{1} M \delta_\eta, \ \text{for all } \eta \in \mathcal{T}_N, \tag{38f}$$

where $\mathbf{1}$ denotes the column vector of ones and M is a large positive number consistent with the "Big-M" approach described in Section 9.1.3 of the textbook by Williams (2013). The dynamics of the system are captured by (38b) which follows from (16). C_{t_η} and b_{t_η} in (38f) represent the state constraints as defined in (8).

4.2. Theoretical results

The following result states conditions for the existence of a solution to MILP (38).

Lemma 2. *For a given $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, suppose $M > 0$ is sufficiently large such that (38f) is satisfied for all $\eta \in \mathcal{T}_N$ and x_η according to (38b) for any $\mathcal{U}_N$. Then a solution to MILP (38) exists.*

Proof. Because M is assumed to be sufficiently large, for a given $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, $\delta_\eta = 1$ for all $\eta \in \mathcal{T}_N$ satisfies the constraints of the MILP for any $\mathcal{U}_N$. Since $\delta_\eta \in \{0, 1\}$ and $N + 1$ is the number of tree nodes, the number of possible sets $\mathcal{D}_N$ is 2^{N+1}. Furthermore, $\rho_\xi \in [0, 1]$ for all $\xi \in \mathcal{T}_N$. Thus, a feasible solution exists for at least one of the 2^{N+1} $\mathcal{D}_N$ sets, i.e., the existence of a solution to MILP (38) follows.

Section 4.3 further discusses how to select a proper M, in particular: how to choose M sufficiently large as required by Lemma 2.

Theorem 3 shows that, under suitable assumptions, a solution to MILP (38) is equivalent to a solution to problem (21). This result is based on the fact that a solution, $\mathcal{U}_N$, of MILP (38) defines a control policy $\pi_{\mathcal{U}_N}$ according to

$$\pi_{\mathcal{U}_N}(x_\eta, w_\eta, t_\eta) = u_\eta \in \mathcal{U}_N, \tag{39}$$

for each $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$ and x_η satisfying (16) with $u_{\text{pre}(\eta)} \in \mathcal{U}_N$. Likewise, a control policy $\pi_N^* \in \Pi$ defines a set of control inputs for a given tree $\mathcal{T}_N$ as

$$\mathcal{U}_N(\pi_N^*) = \{u_\eta = \pi_N^*(x_\eta, w_\eta, t_\eta) \,:\, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}, \tag{40}$$

where x_η satisfies (16) for $u_{\text{pre}(\eta)} \in \mathcal{U}_N(\pi_N^*)$.

Theorem 3. *Suppose Assumptions 1 and 3 hold and M is sufficiently large as in Lemma 2. Then $\mathcal{U}_N^*$ is a solution to MILP (38) if the control policy $\pi_{\mathcal{U}_N^*}$ according to (39) is a solution to problem (21). Likewise, $\pi_N^* \in \Pi$ is a solution to problem (21) if $\mathcal{U}_N(\pi_N^*)$ according to (40) is a solution to MILP (38).*

Proof. Let $x = x_0 \in G_0$ and $w = w_0 \in W$ be a given initial condition and $\mathcal{T}_N$ be the corresponding scenario tree, $N \in \mathbb{Z}_+$. For the first part of the proof, suppose π_N^* is a solution to problem (21). Thus,

$$\bar{\tau}_N(x, w, \pi_N^*) \geq \bar{\tau}_N(x, w, \pi_N^\#), \tag{41}$$

for all $\pi_N^\# \in \Pi$. A solution to MILP (38) exists due to the assumptions and Lemma 2. Using (40), fix $\mathcal{U}_N = \mathcal{U}_N(\pi_N^*)$ in MILP (38) and denote the resulting $\mathcal{D}_N$ by $\mathcal{D}_N^* = \{\delta_{\eta *} \in \{0, 1\} \,:\, \eta \in \mathcal{T}_N\}$. Similarly, let $\mathcal{D}_N^\# = \{\delta_{\eta \#} \in \{0, 1\} \,:\, \eta \in \mathcal{T}_N\}$ denote the MILP solution when $\mathcal{U}_N = \mathcal{U}_N(\pi_N^\#)$ is fixed. Hence, by (38d)–(38f), for each $\eta \in \mathcal{S}_N$, $\delta_{\xi *} = 1$ iff $t_\xi \geq \tau_{N,\eta}(x, w, \pi_N^*)$, $\delta_{\xi \#} = 1$ iff $t_\xi \geq \tau_{N,\eta}(x, w, \pi_N^\#)$, $\delta_{\xi *} = 0$ iff $t_\xi < \tau_{N,\eta}(x, w, \pi_N^*)$, and $\delta_{\xi \#} = 0$ iff $t_\xi < \tau_{N,\eta}(x, w, \pi_N^\#)$ for all $\xi \in \mathcal{H}_{N,\eta}$. Consequently, according to (19), it follows that

$$\tau_{N,\eta}(x, w, \pi_N^*) = t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi *} \tag{42a}$$

$$\tau_{N,\eta}(x, w, \pi_N^\#) = t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi \#}, \tag{42b}$$

for all $\eta \in \mathcal{S}_N$. Then, using (20), (41), and (42), one obtains

$$\begin{aligned} &\sum_{\eta \in \mathcal{S}_N} (t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi *}) \rho_\eta = \bar{\tau}_N(x, w, \pi_N^*) \\ &\geq \bar{\tau}_N(x, w, \pi_N^\#) = \sum_{\eta \in \mathcal{S}_N} (t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi \#}) \rho_\eta. \end{aligned} \tag{43}$$

Consequently,

$$\sum_{\eta \in \mathcal{S}_N} \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi *} \rho_\eta \leq \sum_{\eta \in \mathcal{S}_N} \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi \#} \rho_\eta. \tag{44}$$

By (22) and (23), $\eta \in \mathcal{S}_N$ and $\xi \in \mathcal{H}_{N,\eta}$ iff $\xi \in \mathcal{T}_N$ and $\eta \in \mathcal{K}_{N,\xi}$. Therefore, (44) is equivalent to

$$\sum_{\xi \in \mathcal{T}_N} \sum_{\eta \in \mathcal{K}_{N,\xi}} \delta_{\xi *} \rho_\eta \leq \sum_{\xi \in \mathcal{T}_N} \sum_{\eta \in \mathcal{K}_{N,\xi}} \delta_{\xi \#} \rho_\eta, \tag{45}$$

which shows that $\mathcal{U}_N(\pi_N^*)$, $\mathcal{D}_N^*$ is a solution to MILP (38). This completes the first part of the proof.

For the second part of the proof, let $\mathcal{U}_N^*$, $\mathcal{D}_N^*$ be a solution to MILP (38), which exists by Lemma 2, where $\mathcal{D}_N^* = \{\delta_{\eta *} \in \{0, 1\} \,:\, \eta \in \mathcal{T}_N\}$. Hence,

$$\sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_{N,\eta}} \delta_{\eta *} \rho_\xi \leq \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_{N,\eta}} \delta_{\eta \#} \rho_\xi, \tag{46}$$

for any $\mathcal{U}_N = \mathcal{U}_N^{\#}$ fixed in MILP (38) with corresponding solution $\mathcal{D}_N^{\#} = \{\delta_{\eta\#} \in \{0, 1\} : \eta \in \mathcal{T}_N\}$. Now define $\pi_{\mathcal{U}_N^*}$ according to (39). Since the dynamics in (1) and (38b) are equivalent, it follows from (19) and (38d)–(38f) that, for each $\eta \in \mathcal{S}_N$,

$$\begin{aligned}\tau_{N,\eta}(x, w, \pi_{\mathcal{U}_N^*}) &= \min\{\min\{t_\xi \in \mathbb{Z}_{[0,t_\eta]} : \\ &\qquad \delta_{\xi *} = 1, \xi \in \mathcal{H}_{N,\eta}\} \cup \{t_\eta + 1\}\} \\ &= t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi *}.\end{aligned} \tag{47}$$

Thus, by (20), the average first exit-time of tree $\mathcal{T}_N$ with control policy $\pi_{\mathcal{U}_N^*}$ is given by

$$\bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^*}) = \sum_{\eta \in \mathcal{S}_N} (t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi *})\rho_\eta. \tag{48}$$

In analogy, define $\pi_{\mathcal{U}_N^{\#}}$ according to (39). Hence,

$$\bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^{\#}}) = \sum_{\eta \in \mathcal{S}_N} (t_\eta + 1 - \sum_{\xi \in \mathcal{H}_{N,\eta}} \delta_{\xi \#})\rho_\eta. \tag{49}$$

Using (22) and (23), it follows from (46), (48), and (49) that

$$\begin{aligned}&\bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^*}) - \bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^{\#}}) \\ &\quad = \sum_{\eta \in \mathcal{S}_N} \sum_{\xi \in \mathcal{H}_{N,\eta}} (\delta_{\xi \#} - \delta_{\xi *})\rho_\eta \\ &\quad = \sum_{\xi \in \mathcal{T}_N} \sum_{\eta \in \mathcal{K}_{N,\xi}} (\delta_{\xi \#} - \delta_{\xi *})\rho_\eta \geq 0,\end{aligned} \tag{50}$$

implying that $\pi_{\mathcal{U}_N^*}$ is a solution to problem (21).

4.3. Choosing M

In practice, choosing M for MILP (38) to be sufficiently large as required by Lemma 2 may be achieved by setting M to the largest number that can be represented by a given computer, which may, however, lower the computational performance. On the other hand, underestimating M may also be tolerated in practice since it effectively tightens the state constraints which may result in potentially more conservative (sub-optimal) solutions.

Alternatively, the least upper bound for M to be sufficiently large as required by Lemma 2 can be determined by solving a linear program (LP) for each node of the tree. The LP for a node $\xi \in \mathcal{T}_N$ has the following form:

$$\mathbf{M}_\xi = \max_{x_N, \mathcal{U}_N} C_{t_\xi} x_\xi - b_{t_\xi} \quad \text{s.t.} \tag{51a}$$

$$x_\eta = A_{t_{\text{pre}(\eta)}} x_{\text{pre}(\eta)} + B_{t_{\text{pre}(\eta)}} u_{\text{pre}(\eta)} + w_{\text{pre}(\eta)}, \quad \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{51b}$$

$$u_\eta \in U_{t_\eta}, \ \text{for all } \eta \in \mathcal{T}_N \setminus \mathcal{S}_N, \tag{51c}$$

where $\mathbf{M}_\xi = [M_{\xi,1}, \ldots, M_{\xi,n}]^\top$ is a column vector. The least upper bound (LUB) for M to be sufficiently large is obtained from the solutions to LPs (51) as

$$M_{\text{LUB}} = \max_{\xi \in \mathcal{T}_N} \max_{i \in \{1,\ldots,n\}} M_{\xi,i}. \tag{52}$$

5. Stochastic MPC

5.1. Theoretical results

For a given scenario tree $\mathcal{T}_N$ with initial $w \in W$ and root node $w_{\eta_0} = w$, the control policy $\pi_{\mathcal{U}_N^*}$, derived from the MILP solution $\mathcal{U}_N^*$ according to (39), maximizes the average first exit-time $\bar{\tau}_N$ for a given $\mathcal{T}_N$ (Theorem 3) and achieves average first exit-times $\bar{\tau}$ arbitrarily close to the optimal value of problem (7) for sufficiently large N (Theorem 2). However, $\pi_{\mathcal{U}_N^*}$ is only defined for the disturbance scenarios encoded by tree $\mathcal{T}_N$, which are the most likely scenarios for the specified N according to Algorithm 1. Thus, starting at $w_0 = w$, $w_t \notin \{w_\eta : \eta \in \mathcal{T}_N, t_\eta = t\}$ may occur at some $t \in \mathbb{Z}_+$, i.e., a disturbance scenario may occur that is not represented by $\mathcal{T}_N$.

Therefore, a stochastic MPC (SMPC) strategy is proposed based on MILP (38), where the solution of the MILP is recomputed at each time instant for an updated tree $\mathcal{T}_N$ based on the current state vector. This approach furthermore provides feedback to compensate for unmodeled effects and can be effective in the context of controlling a nonlinear system and/or when the exact disturbance model is unknown. In such a case, the stochastic linear model in (1) and the Markov chain for w_t serves as an approximation of the nonlinear system and/or the unknown disturbance model.

For a given $x \in G_{t_0}$, $w \in W$, and $t_0 \in \mathbb{Z}_{\geq 0}$, the proposed SMPC scheme establishes the following control policy $\pi_{\text{SMPC},N} \in \Pi$:

$$\pi_{\text{SMPC},N}(x, w, t_0) = u_{\eta_0} \in \mathcal{U}_N^*, \tag{53}$$

where $\mathcal{U}_N^*$ is a solution to MILP (38) for the scenario tree $\mathcal{T}_N$ with root node η_0 and

$$t_{\eta_0} \leftarrow t_0, x_{\eta_0} \leftarrow x, \ \text{and } w_{\eta_0} \leftarrow w,$$

in Step 2 of Algorithm 1. It follows from Theorems 2 and 3 that, in terms of first exit-time performance, the control policy $\pi_{\text{SMPC},N}$ in (53) is arbitrarily close to a solution (assuming one exists, i.e., Assumption 3) of problem (7) for sufficiently large N. This is summarized in Theorem 4, the proof of which follows from the proofs of Theorems 2 and 3.

Theorem 4. *Suppose Assumptions 1 and 3 hold, $\pi_{\text{SMPC},N}$ is as in (53), and M is sufficiently large as in Lemma 2. Then, for each $x \in G_0$, $w \in W$, and $\varepsilon > 0$, there exists $\bar{N} > 0$ such that*

$$\bar{\tau}(x, w, \pi_{\text{SMPC},N}) + \varepsilon \geq \max_{\pi \in \Pi} \bar{\tau}(x, w, \pi),$$

for all $N \geq \bar{N}$.

5.2. Practical implementation

Mixed-integer programming, including MILP, has non-polynomial complexity in the worst-case, which makes practical mixed-integer MPC (MI-MPC) applications with strict real-time constraints challenging. However, since MI-MPC was introduced about two decades ago, see e.g., Bemporad and Morari (1999), and initial applications were presented, such as by Richards and How (2005), a lot of progress has been made on efficient solvers suitable for embedded real-time applications. For example, Bemporad and Naik (2018), Hespanhol et al. (2019), Marcucci and Tedrake (2019), and Stellato et al. (2018) have developed branch-and-bound-based algorithms and efficient warm start strategies for solving mixed-integer programs for fast embedded MI-MPC by exploiting the specific structure of the optimal control problem.

In addition to the improvements in software, computer processing speeds have increased by several orders of magnitude due to hardware advancements in field programmable arrays, multi-core processors, and graphics processing units as discussed, for example, by Abughalieh and Alawneh (2019), Grubmüller et al. (2018), Phung et al. (2017), Rogers and Slegers (2013), and Sampathirao et al. (2018).

While the recent hardware improvements and solver developments are encouraging for reducing worst-case computation times of mixed-integer programs, we also introduce a relaxed version of MILP (38), a standard LP, that is used as a fallback to enforce computation time limits if an allocated time for the MILP

solution is exceeded. The LP for a given tree $\mathcal{T}_N$ is obtained by replacing the integer variables δ_η in MILP (38) by non-negative real variables ε_η for all $\eta \in \mathcal{T}_N$. Thus, the LP is as follows:

$$\min_{\mathcal{X}_N, \mathcal{U}_N, \mathcal{E}_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_{N,\eta}} \varepsilon_\eta \rho_\xi \quad \text{s.t.} \tag{54a}$$

$$x_\eta = A_{t_{\text{pre}(\eta)}} x_{\text{pre}(\eta)} + B_{t_{\text{pre}(\eta)}} u_{\text{pre}(\eta)} + w_{\text{pre}(\eta)}, \quad \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{54b}$$

$$u_\eta \in U_{t_\eta}, \ \text{for all } \eta \in \mathcal{T}_N \setminus \mathcal{S}_N \tag{54c}$$

$$\varepsilon_\eta \geq \varepsilon_{\text{pre}(\eta)} \geq 0, \ \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{54d}$$

$$C_{t_\eta} x_\eta \leq b_{t_\eta} + \mathbf{1}\varepsilon_\eta, \ \text{for all } \eta \in \mathcal{T}_N, \tag{54e}$$

where $\mathcal{X}_N$ and $\mathcal{U}_N$, see (36), are sets of states and control values, respectively, corresponding to the nodes of a tree $\mathcal{T}_N$. A set of ε_η values for a tree $\mathcal{T}_N$ is denoted by $\mathcal{E}_N = \{\varepsilon_\eta \geq 0 : \eta \in \mathcal{T}_N\}$. A solution to LP (54) always exists because $\varepsilon_\eta \geq 0$ can always be chosen sufficiently large such that (54e) is satisfied for all $\eta \in \mathcal{T}_N$.

Our numerical case studies in Section 6 suggest that using the relaxed LP as a fallback in case the MILP computation exceeds a prescribed maximum time is effective. However, the connections between solutions of MILP and relaxed LP need to be better understood in general. Further research in this direction is left to future work.

Algorithm 2 SMPC implementation

1: $t \leftarrow 0$
2: $x, w \leftarrow$ obtain current $x(t)$ and $w(t)$
3: $\mathcal{T}_N \leftarrow$ output of Algorithm 1 with $t_{\eta_0} \leftarrow t$, $x_{\eta_0} \leftarrow x$, and $w_{\eta_0} \leftarrow w$ in Step 2 of Algorithm 1
4: $t_{\text{comp}} \leftarrow 0$
5: **while** computing solution of MILP (38) **do**
6: **if** $t_{\text{comp}} > t_{\max}$ **then**
7: Go to Step 12
8: **end if**
9: $t_{\text{comp}} \leftarrow$ update t_{comp}
10: **end while**
11: $\mathcal{U}_N^* \leftarrow$ solution of MILP (38); go to Step 13
12: $\mathcal{U}_N^* \leftarrow$ solution of LP (54)
13: $u(t) \leftarrow$ apply control $u_{\eta_0} \in \mathcal{U}_N^*$ to the system
14: $t \leftarrow t + 1$
15: Go to Step 2

Algorithm 2 outlines the practical implementation of the SMPC strategy. At each time instant t, the current state vector and disturbance are obtained in Step 2 of Algorithm 2. Based on these values, a new scenario tree is constructed in Step 3 using Algorithm 1. Then a solution $\mathcal{U}_N^*$ of MILP (38) is computed. If the MILP computation time t_{comp} exceeds a specified upper bound $t_{\max}$, solving the MILP is terminated in Steps 6–8, and LP (54) is solved instead. The root node control input u_{η_0} of the MILP solution $\mathcal{U}_N^*$ (or the LP solution in case $t_{\text{comp}} > t_{\max}$) is applied to the system in Step 13 and the procedure is repeated at the next time instant $t + 1$.

5.3. Selecting the number of tree nodes

The problem of choosing the number of tree nodes is similar to the problem of selecting the prediction horizon in conventional MPC. Thus, similar to conventional MPC, N can be selected during the algorithm calibration and tuning phase based on the trade-off between performance and computation time. In the offline calibration phase, numerical experiments may be used to assess the required N for a specific problem. Typically, the optimal value of N is where the average first exit-time dependence on N obtained from simulations begins to flatten out as N grows. Figs. 3 and 7 demonstrate this pattern for the two numerical case studies in this paper.

While some general problem-agnostic guidelines exist for conventional MPC, for example, based on settling time and relaxed dynamic programming inequality as discussed by Grüne and Pannek (2017), the development of similar guidelines in our case is left to future work.

6. Numerical case studies

This section provides numerical case studies of applications of the proposed SMPC strategy in Algorithm 2. In order to reduce computation times, the scenario trees $\mathcal{T}_N(w^i)$ with $w_{\eta_0} = w^i$ are precomputed offline and stored for each $w^i \in W$ instead of constructing $\mathcal{T}_N$ online at each time instant. All computations involving the SMPC strategy are performed in MATLAB 2016 on a laptop, where the Hybrid Toolbox developed by Bemporad (2004) (with default settings) is used to solve LPs and MILPs.

6.1. Case study 1

In this case study, we investigate the influence of the number of tree nodes on the solution for a stochastic linear time-varying system of the form

$$\begin{bmatrix} r_{1,t+1} \\ r_{2,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 1.2 \end{bmatrix} \begin{bmatrix} r_{1,t} \\ r_{2,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \sin(t/2) \end{bmatrix} u_t + \begin{bmatrix} 0 \\ w_t \end{bmatrix}, \tag{55}$$

where $x = [r_1, r_2]^\top$ denotes the state vector and the control input is $u \in [-1, 1]$.

The disturbance w takes values in the set $W = \{-1, 0, 1\}$ with transition probabilities $P_W(w^i|w^j) = [P_{W,\text{Mat}}]_{j,i}$ (j = row number and i = column number), $i, j \in \{1, 2, 3\}$, given by the following matrix:

$$P_{W,\text{Mat}} = \begin{bmatrix} 0 & 0.8 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0.35 & 0.4 & 0.25 \end{bmatrix}.$$

The constraints for the optimal control problem (7) are given by the set $G_t \equiv \{x : -2 \leq r_1 \leq 2, -2 \leq r_2 \leq 2\}$.

The time limit in Algorithm 2 for solving the MILP is set to $t_{\max} = 10$ s. The following results are for an initial $x_0 = [0, 0]^\top$ and $w_0 = -1$. Fig. 2 shows a sample trajectory (for $N = 200$), where the dashed lines indicate the respective constraints.

The average first exit-time $\bar{\tau}$ (based on 1000 random simulations for each N) is plotted against N in Fig. 3. For comparison, a dynamic programming (DP) solution obtained by conventional value iteration, see Bertsekas (2005), is shown as a reference in Fig. 3 (dashed line). The DP solution, achieving $\bar{\tau} = 32.41$ s, is based on a discrete grid of the state space using linear interpolation between the grid points, where a relatively dense grid of 900000 points is used here. The set defining the control constraints is discretized as well, using an equidistant grid with 21 points. About 1.8 h are required to compute the DP control policy offline when implemented in C++ 11. Due to the dense grid representations of both G_t and U_t, the DP reference solution is expected to be close to a solution of the optimal control problem (7).

In line with Theorem 4, it can be seen in Fig. 3 that the SMPC solution improves with increasing N and approaches the DP solution (which we expect to be close to an optimal solution), where the DP reference is slightly exceeded for $N \geq 500$. This exceedance is attributed to the numerical approximation errors.

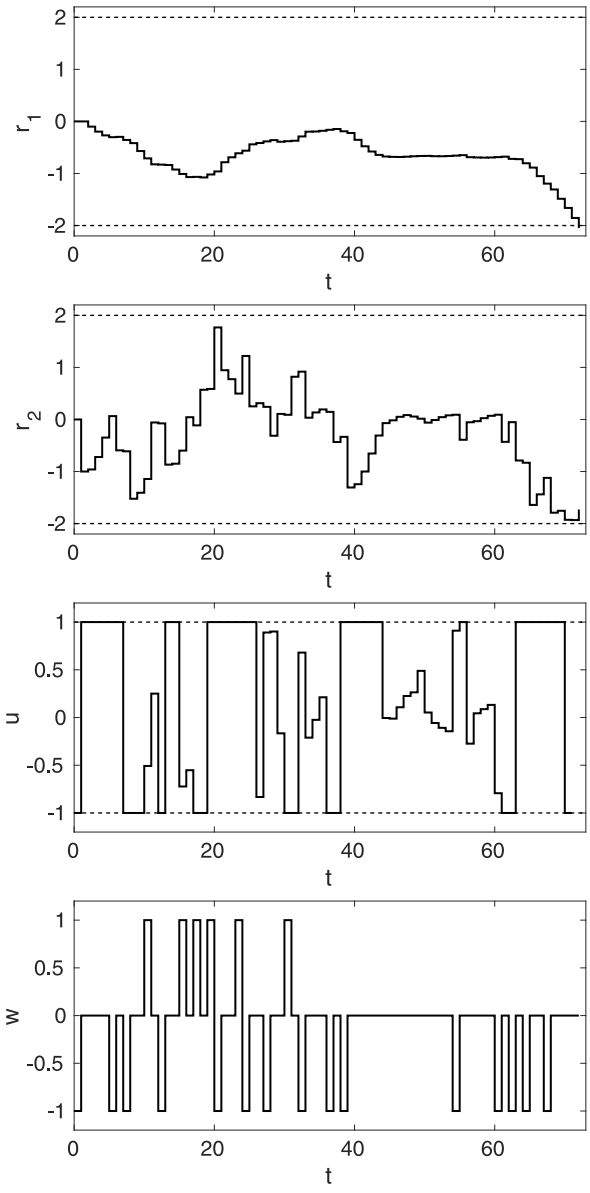

Fig. 2. Numerical case study 1: sample trajectory showing the states r_1 (top plot) and r_2 (second plot), as well as the control input u (third plot) and disturbance w (bottom plot) vs. t.

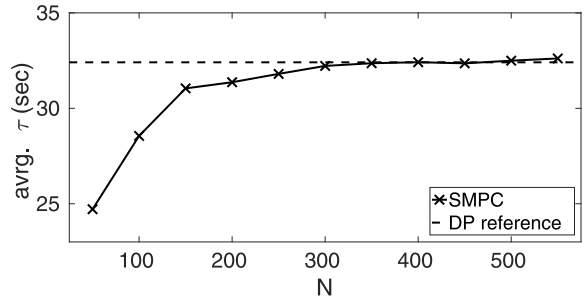

Fig. 3. Numerical case study 1: average first exit-time $\bar{\tau}$ vs. N (based on 1000 random simulations for each N).

The computation time (in MATLAB) for the SMPC scheme to compute the control input at each time instant according to Algorithm 2 (Steps 2–13) is shown in Fig. 4 for different N. The

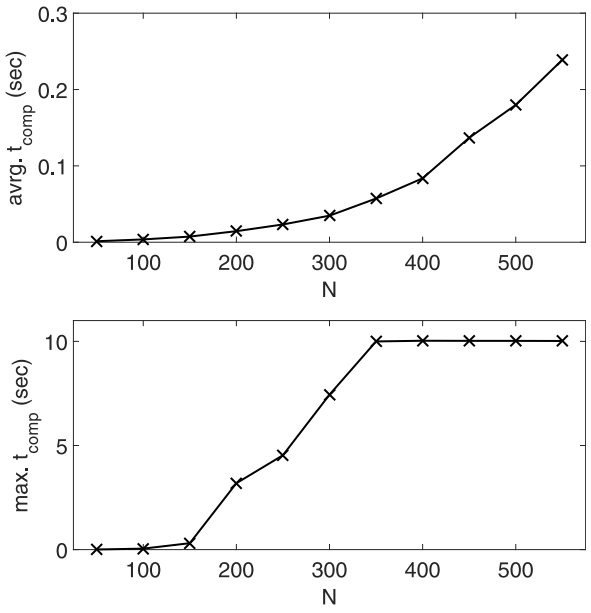

Fig. 4. Numerical case study 1: average (top) and worst-case time (bottom) to compute control u_t (Steps 2–13 in Algorithm 2) vs. N (based on 1000 random simulations for each N).

top plot in Fig. 4 shows the average computation time, which increases nearly exponentially with N. The worst-case/maximum computation time is shown in Fig. 4 (bottom), where the prescribed limit on the MILP computation time $t_{\max} = 10$ s is reached at least once for $N \geq 400$.

6.2. Case study 2: Car following

A car following or adaptive cruise control problem is treated in this case study. The problem involves two vehicles: the lead vehicle and the follower vehicle. With s denoting the distance between the two vehicles, the objective is to control the speed of the follower vehicle v_{f}, such that the time gap between the two vehicles,

$$T_{\mathrm{g}} = s/v_{\mathrm{f}}, \tag{56}$$

stays within prescribed bounds for as long as possible.

The speed of the lead vehicle v_{l} is modeled by a Markov chain that takes values in the set

$$W = \{27, 27.25, 27.5, \ldots, 32\} \text{ m/s},$$

which contains 21 elements. The system is represented by the following model:

$$\begin{aligned} s_{t+1} &= s_t + \Delta t(v_{\mathrm{l},t} - v_{\mathrm{f},t}), \\ v_{\mathrm{f},t+1} &= v_{\mathrm{f},t} + \Delta t a_t. \end{aligned} \tag{57}$$

We consider a sampling time of $\Delta t = 1$ s. The control input at a time instant t is the acceleration of the follower vehicle,

$$a_t \in [-1, 0.3]\,\mathrm{m/s^2}.$$

The state constraints for this problem are represented by the set,

$$G_t \equiv \{s, v_{\mathrm{f}} : T_{\mathrm{g}} = s/v_{\mathrm{f}} \in [0.5, 2.5] \text{ s}, \ v_{\mathrm{f}} \leq 30\,\mathrm{m/s}\}.$$

The transition probabilities of the lead vehicle velocity are similar to the values in Kolmanovsky and Filev (2009), which are based on experimental data. Moreover, as in Kolmanovsky and Filev (2009), the possibility of another vehicle cutting in upfront is taken into account by slightly modifying the model. In this case

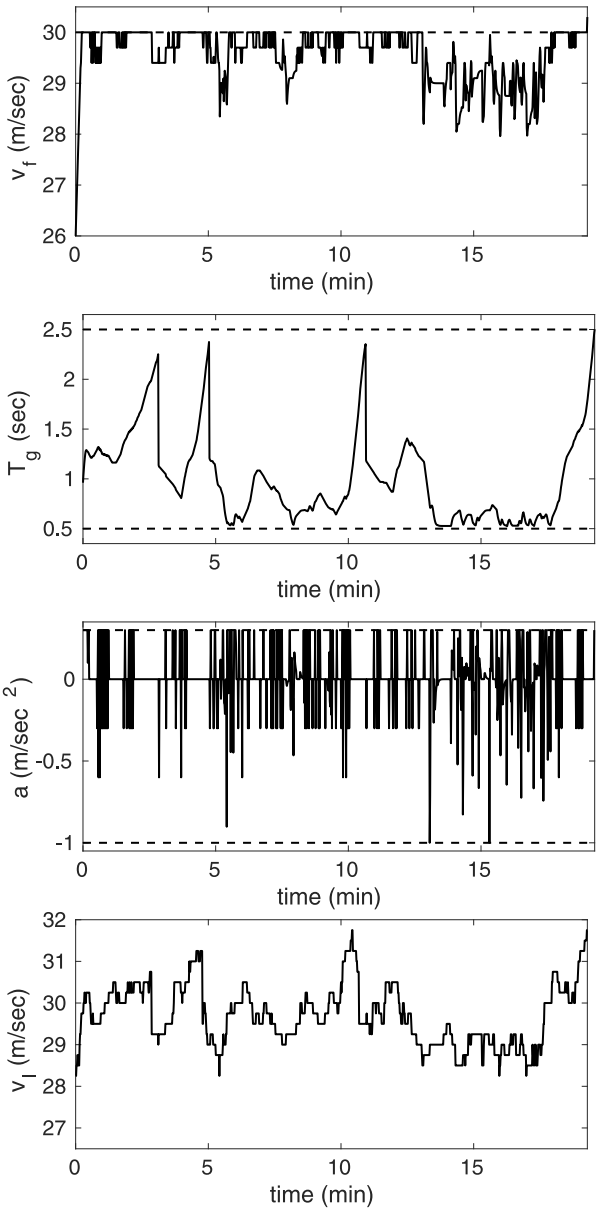

Fig. 5. Car following case study, sample trajectory using the SMPC approach without applying a control input penalty ($\beta = 0$): follower speed v_f (top plot), time gap T_g (second plot), control input a (third plot), and disturbance v_l (bottom plot) vs. time t.

study, such an event may occur with a probability of 0.05 if the time gap T_g is greater than 2.2 s In case of another vehicle cutting in upfront, the distance between the vehicles is set to half of the previous distance. Moreover, the vehicle is assumed to cut in with a speed of 29.5 m/s. The model is thus a stochastic hybrid model with state-dependent probabilities for mode switches.

As in the previous example, we compare the results of the proposed SMPC approach to a DP approach. The DP approach is able to explicitly account for the hybrid characteristics of the system. The SMPC strategy, on the other hand, uses the stochastic linear model in (57) as an approximation of the system, while neglecting the possibility of another vehicle cutting in upfront. SMPC compensates for the unmodeled effects through feedback.

Both approaches (SMPC and DP) generate frequent aggressive velocity changes which may be uncomfortable for passengers and inefficient, i.e., wasting fuel. In order to avoid this, control

Table 1
Car following case study, SMPC solution with $N = 100$: influence of control input penalty weight β on average first exit-time $\bar{\tau}$ (1500 random simulations each).

β	0	0.01	0.05	0.1
$\bar{\tau}$ (s)	858	465.1	302.9	13.4

inputs are penalized by considering the weighted sum of the absolute values of the control inputs as an additional objective to be minimized. This modification of the optimization problem (7) is a follows:

$$\max_{\pi \in \Pi} \bar{\tau}(x_0, w_0, \pi) - \beta \mathrm{E}\left\{ \sum_{t=0}^{\tau(x_0,w_0,\pi)-1} |\pi(x_t, w_t, t)| \right\}, \tag{58}$$

where x_t results from applying the control policy π to (1). The factor for weighting the control input penalty is denoted by β.

The SMPC approach is modified accordingly by penalizing the weighted sum of $|u_\eta|$. As done in Earl and D'Andrea (2005), this is achieved by introducing new variables $\gamma_\eta \geq 0$ for each $\eta \in \mathcal{T}_N \backslash \mathcal{S}_N$, and adding the following control input constraint to MILP (38) and LP (54) for each $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$:

$$-\gamma_\eta \leq u_\eta \leq \gamma_\eta. \tag{59}$$

Moreover, the weighted sum of γ_η values,

$$\beta \sum_{\eta \in \mathcal{T}_N \backslash \mathcal{S}_N} \gamma_\eta, \tag{60}$$

is added to the objective functions of MILP (38) and LP (54).

For numerical reasons, the probability of each scenario, given by ρ_η for all $\eta \in \mathcal{S}_N$, is normalized by dividing ρ_η by the sum of the probabilities of all scenarios of tree $\mathcal{T}_N$, i.e.,

$$\rho_{\eta,\mathrm{norm}} = \rho_\eta / \sum_{\xi \in \mathcal{S}_N} \rho_\xi. \tag{61}$$

which, instead of ρ_η, is used in (38a) and (54a).

Table 1 shows the average first exit-time $\bar{\tau}$ with the SMPC approach for different control input penalty weights β. As expected, $\bar{\tau}$ decreases with increasing β since a large β emphasizes less intense and less frequent acceleration/deceleration.

Note that we run 1500 random simulations to estimate $\bar{\tau}$ for each case since the result does not significantly change beyond that. The limit on the MILP computation time is set to $t_{\max} = 1$ s.

The difference between penalizing the control inputs vs. not penalizing the control inputs is furthermore demonstrated in Figs. 5 and 6, which show sample trajectories using the SMPC approach with $\beta = 0$ and $\beta = 0.01$, respectively. It is evident that without control input penalty the control inputs are frequent and large in value in order to maintain the follower vehicle speed near its maximum value. In contrast, with control penalty the control inputs are lower and smaller in value, resulting in an overall smoother, hence more comfortable, trajectory.

For the remainder of this case study, we use $\beta = 0.01$ for the SMPC approach as well as for the DP solution for comparison. For the DP comparison, we use a uniform grid of 33600 points for the state space and linearly interpolate between the grid points. The control space is uniformly discretized into 14 points between the minimum and maximum values. This setting implemented in C++ 11 requires about six hours to compute the DP control policy offline. The SMPC approach, on the other hand, is capable to be employed in an online setting. This can be seen in Fig. 8 which shows the online computation times (average and worst-case values per simulation run) for different tree sizes N. The

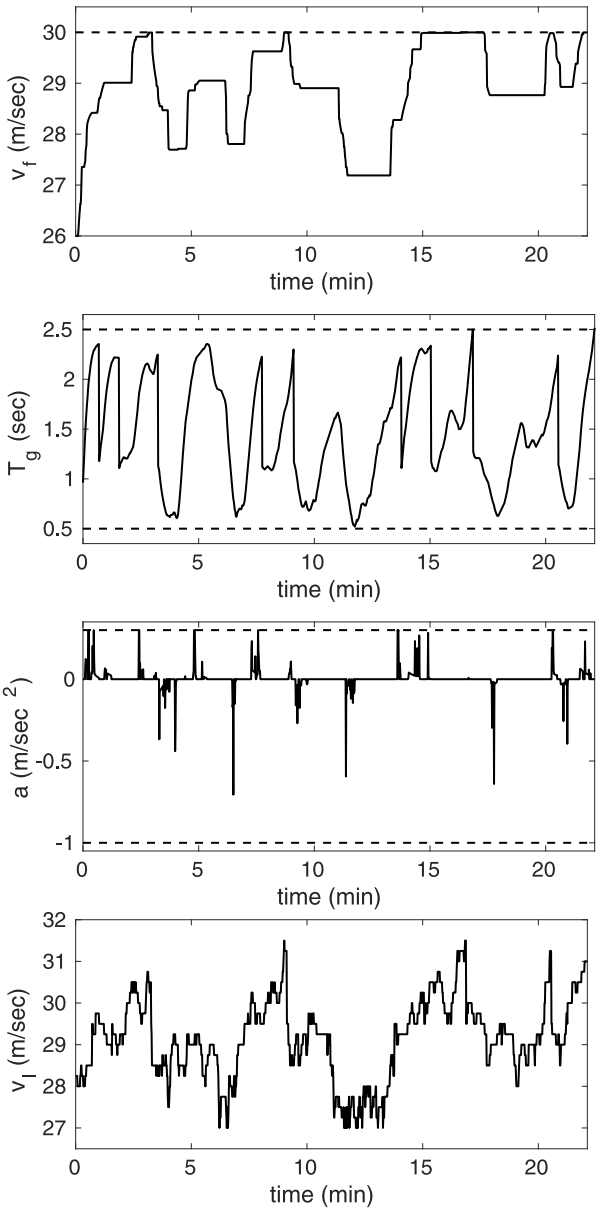

Fig. 6. Car following case study, sample trajectory using the SMPC approach with control input penalty ($\beta = 0.01$): follower speed v_f (top plot), time gap T_g (second plot), control input a (third plot), and disturbance v_l (bottom plot) vs. time t.

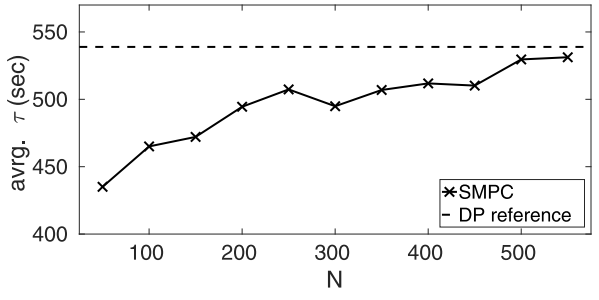

Fig. 7. Car following case study: average first exit-time $\bar{\tau}$ vs. N (1500 random simulations each).

SMPC solutions are computed in MATLAB, whereas an implementation in a lower-level programming language such as C++ is expected to further improve computation times.

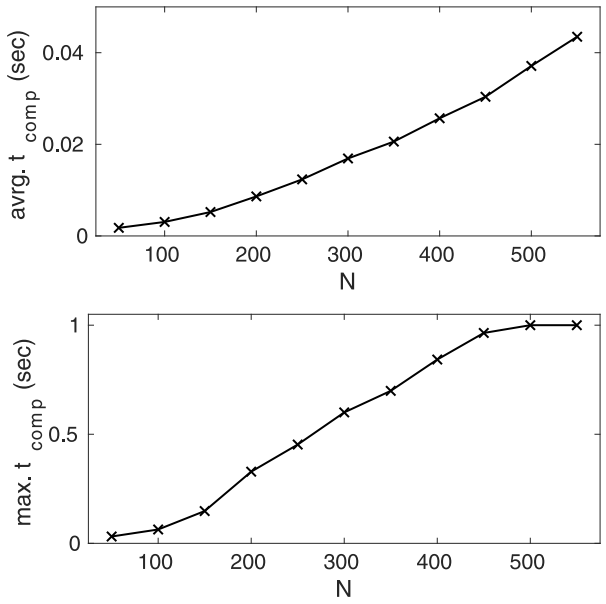

Fig. 8. Car following case study: average (top) and worst-case time (bottom) to compute control u_t (Steps 2–13 in Algorithm 2) vs. N (1500 random simulations each).

The average first exit-time $\bar{\tau}$ vs. the tree size N is shown in Fig. 7. The DP solution with conventional value iteration, yielding an average first exit-time of 539 s, is also shown for comparison. As in the previous example (see Fig. 3), the SMPC solution improves with increasing N and approaches the DP solution.

7. Conclusion

A stochastic model predictive control (SMPC) strategy was developed for solving optimal control problems with the objective of maximizing the average time until a linear system with additive random disturbance violates prescribed constraints on its state variables. The SMPC strategy is based on a tree structure with a specified number of tree nodes and a tree generation algorithm has been defined to emphasize the inclusion of the most relevant scenarios. By repeatedly solving a mixed-integer linear program over a receding time horizon based on the current state variables and disturbance, the SMPC strategy obtains solutions arbitrarily close to the optimal solution in terms of average time until constraint violation. The effectiveness of the proposed SMPC strategy was demonstrated in two numerical case studies, including a car following control problem.

References

Abughalieh, K. M., & Alawneh, S. G. (2019). A survey of parallel implementations for model predictive control. *IEEE Access*, *7*, 34348–34360.

Barles, G., & Rouy, E. (1998). A strong comparison result for the Bellman equation arising in stochastic exit time control problems and its applications. *Communications in Partial Differential Equations*, *23*(11–12), 1995–2033.

Bayraktar, E., Song, Q., & Yang, J. (2010). On the continuity of stochastic exit time control problems. *Stochastic Analysis and Applications*, *29*(1), 48–60.

Bemporad, A. (2004). Hybrid toolbox - user's guide.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*(3), 407–427.

Bemporad, A., & Naik, V. V. (2018). A numerically robust mixed-integer quadratic programming solver for embedded hybrid model predictive control. *IFAC-PapersOnLine*, *51*(20), 412–417.

Bernardini, D., & Bemporad, A. (2012). Stabilizing model predictive control of stochastic constrained linear systems. *IEEE Transactions on Automatic Control*, *57*(6), 1468–1480.

Bertsekas, D. P. (2005). *Dynamic programming and optimal control, Vol. I*. Belmont, MA: Athena Scientific.
Buckdahn, R., & Nie, T. (2016). Generalized Hamilton-Jacobi-Bellman equations with Dirichlet boundary condition and stochastic exit time optimal control problem. *SIAM Journal on Control and Optimization*, *54*(2), 602–631.
Cannon, M., Kouvaritakis, B., & Wu, X. (2009). Probabilistic constrained MPC for multiplicative and additive stochastic uncertainty. *IEEE Transactions on Automatic Control*, *54*(7), 1626–1632.
Clark, J., & Vinter, R. (2012). Stochastic exit time problems arising in process control. *Stochastics. An International Journal of Probability and Stochastic Processes*, *84*(5–6), 667–681.
Couchman, P. D., Cannon, M., & Kouvaritakis, B. (2006). Stochastic MPC with inequality stability constraints. *Automatica*, *42*(12), 2169–2174.
Crandall, M. G., Evans, L. C., & Lions, P. -L. (1984). Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, *282*(2), 487–502.
Crandall, M. G., & Lions, P. -L. (1983). Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, *277*(1), 1–42.
Di Cairano, S., Bernardini, D., Bemporad, A., & Kolmanovsky, I. (2014). Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management. *IEEE Transactions on Control Systems Technology*, *22*(3), 1018–1031.
Earl, M. G., & D'Andrea, R. (2005). Iterative MILP methods for vehicle-control problems. *IEEE Transactions on Robotics*, *21*(6), 1158–1167.
Grubmüller, S., Scharrer, M. K., Herbst, B., Tengg, A., Schmidt, H., & Watzenig, D. (2018). Predictive energy management on multi-core systems. In *Comprehensive energy management-safe adaptation, predictive control and thermal management* (pp. 47–66). Springer.
Grüne, L., & Pannek, J. (2017). Nonlinear model predictive control. In *Nonlinear model predictive control* (pp. 45–69). Springer.
Hespanhol, P., Quirynen, R., & Di Cairano, S. (2019). A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control. In *2019 18th European control conference* (pp. 2763–2768). IEEE.
Ikonen, E., Selek, I., & Najim, K. (2016). Process control using finite Markov chains with iterative clustering. *Computers & Chemical Engineering*, *93*, 293–308.
Kolmanovsky, I., & Filev, D. P. (2009). Stochastic optimal control of systems with soft constraints and opportunities for automotive applications. In *IEEE control applications & intelligent control* (pp. 1265–1270).
Kolmanovsky, I., & Maizenberg, T. L. (2002). Optimal containment control for a class of stochastic systems perturbed by Poisson and Wiener processes. *IEEE Transactions on Automatic Control*, *47*(12), 2041–2046.
Kolmanovsky, I., & Zidek, R. (2018). Drift counteraction and control of downsized and underactuated systems: What MPC has to offer? *IFAC-PapersOnLine*, *51*(20), 175–190.
Lee, J. H., & Yu, Z. (1997). Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, *33*(5), 763–781.
Lin, C. -C., Peng, H., & Grizzle, J. (2004). A stochastic control strategy for hybrid electric vehicles. In *Proceedings of the 2004 American control conference. Vol. 5* (pp. 4710–4715). IEEE.
Lions, P. L. (1983). On the Hamilton-Jacobi-Bellman equations. *Acta Applicandae Mathematicae*, *1*(1), 17–41.
Malisoff, M. (2001). Viscosity solutions of the Bellman equation for exit time optimal control problems with non-Lipschitz dynamics. *ESAIM. Control, Optimisation and Calculus of Variations*, *6*, 415–441.
Marcucci, T., & Tedrake, R. (2019). Warm start of mixed-integer programs for model predictive control of hybrid systems. arXiv preprint arXiv:1910.08251.
Mayne, D. Q., Seron, M. M., & Raković, S. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, *41*(2), 219–224.
Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, *36*(6), 30–44.
Phung, D. -K., Hérissé, B., Marzat, J., & Bertrand, S. (2017). Model predictive control for autonomous navigation using embedded graphics processing unit. *IFAC-PapersOnLine*, *50*(1), 11883–11888.
Primbs, J. A., & Sung, C. H. (2009). Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *IEEE Transactions on Automatic Control*, *54*(2), 221–230.
Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
Richards, A., & How, J. (2005). Mixed-integer programming for control. In *Proceedings of the 2005, American control conference, 2005* (pp. 2676–2683).
Rogers, J., & Slegers, N. (2013). Robust parafoil terminal guidance using massively parallel processing. *Journal of Guidance, Control, and Dynamics*, *36*(5), 1336–1345.
Sampathirao, A. K., Sopasakis, P., Bemporad, A., & Patrinos, P. P. (2018). GPU-accelerated stochastic predictive control of drinking water networks. *IEEE Transactions on Control Systems Technology*, *26*(2), 551–562.
Stellato, B., Naik, V. V., Bemporad, A., Goulart, P., & Boyd, S. (2018). Embedded mixed-integer quadratic optimization using the OSQP solver. In *2018 European control conference* (pp. 1536–1541). IEEE.
Williams, H. P. (2013). *Model building in mathematical programming*. John Wiley & Sons.
Zidek, R. A. E., Bemporad, A., & Kolmanovsky, I. V. (2017). Optimal and receding horizon drift counteraction control: linear programming approaches. In *American control conference* (pp. 2636–2641).
Zidek, R. A. E., & Kolmanovsky, I. V. (2017). Optimal driving policies for autonomous vehicles based on stochastic drift counteraction. *IFAC-PapersOnLine*, *50*(1), 290–296.
Zidek, R. A. E., Kolmanovsky, I. V., & Bemporad, A. (2018). Stochastic MPC approach to drift counteraction, In *2018 annual American control conference* (pp. 721–727).

Robert A. E. Zidek is a research scientist and manager in the planning and control team at the Toyota Research Institute. He currently focuses on developing decision making capabilities for L4 and L5 autonomous driving systems, including higher-level vehicle behavior planning and navigation, trajectory planning problem generation, and trajectory selection. Moreover, he is also working on parallel autonomy solutions that include the human driver. He received his Ph.D. degree in Aerospace Engineering from the University of Michigan in 2017. Prior to that, he worked in the automotive industry in Germany on vehicle dynamics and powertrain control systems.

Ilya V. Kolmanovsky is a professor in the department of aerospace engineering at the University of Michigan, with research interests in control theory for systems with state and control constraints, and in control applications to aerospace and automotive systems. He received his Ph.D. degree from the University of Michigan in 1995. He is a Fellow of IEEE and is named as an inventor on over 100 United States patents.

Alberto Bemporad received his Master's degree in Electrical Engineering in 1993 and his Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. In 1996/97 he was with the Center for Robotics and Automation, Department of Systems Science and Mathematics, Washington University, St. Louis. In 1997–1999 he held a postdoctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a senior researcher until 2002. In 1999–2009 he was with the Department of Information Engineering of the University of Siena, Italy, becoming an Associate Professor in 2005. In 2010–2011 he was with the Department of Mechanical and Structural Engineering of the University of Trento, Italy. Since 2011 he is Full Professor at the IMT School for Advanced Studies Lucca, Italy, where he served as the Director of the institute in 2012–2015. He spent visiting periods at Stanford University, University of Michigan, and Zhejiang University. In 2011 he cofounded ODYS S.r.l., a company specialized in developing model predictive control systems for industrial production. He has published more than 350 papers in the areas of model predictive control, hybrid systems, optimization, automotive control, and is the co-inventor of 16 patents. He is author or coauthor of various software packages for model predictive control design and implementation, including the Model Predictive Control Toolbox (The Mathworks, Inc.) and the Hybrid Toolbox for MATLAB. He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001–2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002–2010. He received the IFAC High-Impact Paper Award for the 2011–14 triennial and the IEEE CSS Transition to Practice Award in 2019. He is an IEEE Fellow since 2010.