2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Stochastic MPC Approach to Drift Counteraction

Robert A. E. Zidek[1], Ilya V. Kolmanovsky[1], Alberto Bemporad[2]

***Abstract*— The contribution of this paper is a novel tree-based stochastic model predictive control (SMPC) approach to solve the optimal exit-time control problem for stochastic systems, that is to maximize the expected value of the first time instant at which prescribed constraints are violated. A scenario tree with a specified number of tree nodes is used to encode the most likely system behavior, where each path on the tree corresponds to a distinct disturbance scenario. For linear discrete-time systems with an additive random disturbance, a mixed-integer linear program (MILP) obtains solutions arbitrarily close to the optimal solution for a sufficient number of tree nodes. In order to compensate for an incomplete scenario tree and/or unmodeled effects, feedback is provided by recomputing the MILP solution over a receding time horizon based on the current state and disturbance / scenario tree. Two numerical case studies, including an adaptive cruise control problem, demonstrate the effectiveness of the proposed SMPC scheme compared to dynamic programming solutions.**

I. Introduction

In this paper, we consider stochastic linear discrete-time systems of the form,

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

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 $t \in \mathbb{Z}_{\geq 0}$ and A_t and B_t are time-dependent matrices. The non-empty time-dependent sets $G_t \in \mathbb{R}^n$ and U_t define state and control constraints, respectively.

The variable w denotes a measured random disturbance that is modeled by a Markov chain and takes values in a finite set $W = \{w^1, w^2, ..., w^{|W|}\}$ of cardinality $|W| > 0$. The transition probabilities for w are given by $P_W(w^j|w^i) = P_W(w_{t+1} = w^j | w_t = w^i) \in [0,1]$ for all $w^i, w^j \in W$ and $t \in \mathbb{Z}_{\geq 0}$. A control policy is denoted by $\pi : G_t \times W \times \mathbb{Z}_{\geq 0} \to U_t$ for all $t \in \mathbb{Z}_{\geq 0}$, i.e., $u_t = \pi(x_t, w_t, t)$, and Π is the set of admissible (i.e., U_t-valued) control policies. For a given control policy $\pi \in \Pi$ and initial $x_0 \in G_0$ 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,

$$\tau(x_0, w_0, \pi) = \inf\{t \in \mathbb{Z}_{\geq 0} : x_t \notin G_t\}, \tag{2}$$

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 $\{w_t\}$.

This research is supported by the National Science Foundation Award Number EECS 1404814.

[1]Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI. Emails: robzidek@umich.edu, ilya@umich.edu

[2]IMT – Institute for Advanced Studies Lucca, Italy. Email: alberto.bemporad@imtlucca.it

The average (i.e., the expected value of the) first exit-time is given by

$$\bar{\tau}(x_0, w_0, \pi) = \mathrm{E}\{\tau(x_0, w_0, \pi)\}, \tag{3}$$

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

$$\max_{\pi \in \Pi} \bar{\tau}(x_0, w_0, \pi). \tag{4}$$

Problem (4) can be found in many engineering applications, in particular, those with finite resources (fuel, energy, component life, etc.) or where large persistent disturbances (e.g., wind gusts) are present. Furthermore, driving policies for autonomous vehicles can be generated based on problem (4) as demonstrated in [1]. A solution to problem (4) may be viewed as providing drift counteraction in order to delay constraint violation and is therefore also referred to as drift counteraction optimal control.

A model predictive control (MPC) scheme was proposed for the deterministic version of problem (4) in [2]. For non-exit time problems, stochastic MPC (SMPC) that accounts for the uncertainty in the disturbance has been considered in [3], [4], [5], [6]. At the same time, recent developments in hardware and numerical methods [7] may facilitate practical use of SMPC.

The main contribution of this paper is a novel SMPC scheme to solve problem (4). Similar to the developments in [8], our approach uses a tree structure to encode the most likely disturbance scenarios. We show that a mixed-integer linear program (MILP) yields a control policy that maximizes the average first exit-time for a given scenario tree. Moreover, as the number of tree nodes goes to infinity, the average first exit-time achieved by the MILP solution approaches the optimal average first exit-time of problem (4). The SMPC policy is given by recomputing the MILP solution over a receding time horizon at each time instant based on the current state and an updated scenario tree (based on the current disturbance). Thus, feedback is provided to counteract unmodeled effects and incomplete scenario trees (since a scenario tree only includes a subset of all possible scenarios).

The structure of the paper is as follows. In Section II, a scenario tree is discussed and an algorithm for constructing one is provided. The MILP that maximizes the average first exit-time for a given scenario tree is formulated in Section III. Based on the MILP, the SMPC strategy is stated in Section IV. Section V presents two numerical case studies, including an adaptive cruise control (ACC) problem. A conclusion is provided in Section VI. Throughout this paper, we make the following two assumptions.

Assumption 1: A solution $\pi^* \in \Pi$ (may not be unique) to problem (4) exists for each $x_0 \in G_0$ and $w_0 \in W$.

Assumption 2: The sets G_t and U_t are polytopes for all $t \in \mathbb{Z}_{\geq 0}$, where G_t is given by

$$G_t = \{x : C_t x \leq b_t\}. \tag{5}$$

II. SCENARIO TREE

In order to optimize over a subset of all possible disturbance scenarios, similar to the work in [8], 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

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

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 $\mathrm{pre}(\eta)$. The set of successors of a node $\eta \in \mathcal{T}_N$ is denoted by

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

and the set of leaf nodes of $\mathcal{T}_N$ has the form,

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

Figure 1 shows an example scenario tree $\mathcal{T}_{11} = \{\eta_0, \eta_1, ..., \eta_{11}\}$ for a given Markov chain with $|W| = 3$. For example, $\mathrm{succ}(\eta_1) = \{\eta_2, \eta_6, \eta_{11}\}$ and $\eta_1^{\mathrm{succ}(\eta_1)} = \eta_2$, $\eta_2^{\mathrm{succ}(\eta_1)} = \eta_6$, and $\eta_3^{\mathrm{succ}(\eta_1)} = \eta_{11}$ in Figure 1. The set of leaf nodes is $\mathcal{S}_{11} = \{\eta_5, \eta_7, \eta_8, \eta_9, \eta_{10}, \eta_{11}\}$ in Figure 1.

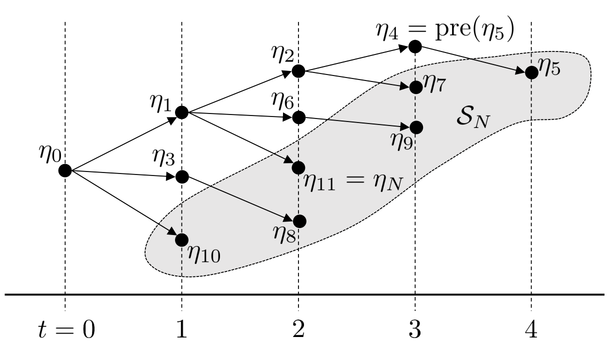

Fig. 1: Scenario tree example for 12 nodes, including $|\mathcal{S}_N| = 6$ leaf nodes.

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,

$$x^\eta = A_{t^{\mathrm{pre}(\eta)}} x^{\mathrm{pre}(\eta)} + B_{t^{\mathrm{pre}(\eta)}} u^{\mathrm{pre}(\eta)} + w^{\mathrm{pre}(\eta)}. \tag{6}$$

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

$$\rho^\eta = \rho^{\mathrm{pre}(\eta)} P_W(w^\eta | w^{\mathrm{pre}(\eta)}) \in [0, 1], \tag{7}$$

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 $j \in \{1, 2, ..., |W|\}$ **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

$$\begin{aligned}\{w_t\}^\eta &= \{w_t : t \in \mathbb{Z}_{[0,t^\eta]}\}^\eta \\ &= (w_0, ..., w^{\mathrm{pre}(\mathrm{pre}(\eta))}, w^{\mathrm{pre}(\eta)}, w^\eta),\end{aligned} \tag{8}$$

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 Figure 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 (8), is defined by

$$\begin{aligned}\tau_N^\eta(x_0, w_0, \pi_N) = \min\{&\min\{t \in \mathbb{Z}_{[0,t^\eta]} : x_t \notin G_t\} \\ &\cup \{t^\eta + 1\}\},\end{aligned} \tag{9}$$

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 (9). The average first exit-time for a given scenario tree $\mathcal{T}_N$ and a control policy $\pi_N \in \Pi$ is given by

$$\bar{\tau}_N(x, w, \pi_N) = \sum_{\eta \in \mathcal{S}_N} \tau_N^\eta(x, w, \pi_N) \rho^\eta. \tag{10}$$

In analogy to problem (4), the optimal control problem of maximizing the average first exit-time over a subset of disturbance scenarios defined by $\mathcal{T}_N$ can be expressed as

$$\max_{\pi_N \in \Pi} \bar{\tau}_N(x, w, \pi_N). \tag{11}$$

The following sets are defined

$$\mathcal{H}_N^{\eta} = \{\eta_0, ..., \text{pre}(\text{pre}(\eta)), \text{pre}(\eta), \eta\}, \text{ for all } \eta \in \mathcal{S}_N, \tag{12}$$

$$\mathcal{K}_N^{\xi} = \{\eta \in \mathcal{S}_N : \xi \in \mathcal{H}_N^{\eta}\}, \text{ for all } \xi \in \mathcal{T}_N, \tag{13}$$

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

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

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

$$\mathcal{Z}_N(\pi, i) = \{\eta \in \mathcal{S}_N : \tau_N^{\eta}(x, w, \pi) = i\}. \tag{14}$$

The next result (Theorem 1) shows that, in terms of the average first exit-time, a solution to (11) is arbitrarily close to a solution (if one exists) of problem (4) for sufficiently large N. Theorem 1 is based on Lemma 1.

Lemma 1:

$$\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) = \bar{\tau}(x, w, \pi), \tag{15}$$

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 (10),

$$\begin{aligned}\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) &= \lim_{N\to\infty} \sum_{\eta\in\mathcal{S}_N} \tau_N^{\eta}(x, w, \pi)\rho^{\eta} \\ &= \lim_{N\to\infty} \left(\sum_{i=1}^{t_N} i \sum_{\eta\in\mathcal{Z}_N(\pi,i)} \rho^{\eta} \right),\end{aligned} \tag{16}$$

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 $i \in \mathbb{Z}_+$,

$$\lim_{N\to\infty} \sum_{\eta\in\mathcal{Z}_N(\pi,i)} \rho^{\eta} = \text{Prob}(\tau(x, w, \pi) = i). \tag{17}$$

Moreover, $t_N \to \infty$ as $N \to \infty$. Consequently, (16) and (17) imply that

$$\begin{aligned}\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) &= \sum_{i=1}^{\infty} i\text{Prob}(\tau(x, w, \pi) = i) \\ &= \bar{\tau}(x, w, \pi).\end{aligned} \tag{18}$$

■

Theorem 1: Suppose Assumption 1 holds. Then, for each $x \in G_0$, $w \in W$, and $\varepsilon > 0$, there exists $\bar{N} > 0$ such that

$$\bar{\tau}(x, w, \pi_N^*) + \varepsilon \geq \max_{\pi\in\Pi} \bar{\tau}(x, w, \pi), \tag{19}$$

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 $x \in G_0$ 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 (4), which exists by Assumption 1, and let $\pi_N^* \in \Pi$ be a control policy that maximizes the average first exit-time for $\mathcal{T}_N$ according to (11), which exists due to the existence of a solution to (4). It follows that

$$\bar{\tau}_N(x, w, \pi_N^*) \geq \bar{\tau}_N(x, w, \pi^*). \tag{20}$$

The optimal average first exit-time of problem (4) may be written as follows

$$\bar{\tau}(x, w, \pi^*) = \bar{\tau}_N(x, w, \pi^*) + \bar{\tau}_{\text{Rest},N}(x, w, \pi^*), \tag{21}$$

where $\bar{\tau}_{\text{Rest},N}$ is the average first exit-time 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 $\bar{\tau}_{\text{Rest},N} \to 0$. This implies that for every $\varepsilon > 0$, there exists $\bar{N} > 0$ such that

$$\bar{\tau}(x, w, \pi^*) \leq \bar{\tau}_N(x, w, \pi^*) + \varepsilon, \tag{22}$$

for all $N \geq \bar{N}$. It follows from (20) and (22) that

$$\bar{\tau}_N(x, w, \pi_N^*) + \varepsilon \geq \bar{\tau}(x, w, \pi^*), \tag{23}$$

for all $N \geq \bar{N}$. In analogy to (21), it follows from adding $\bar{\tau}_{\text{Rest},N}(x, w, \pi_N^*)$ to (23) that

$$\bar{\tau}(x, w, \pi_N^*) + \varepsilon \geq \bar{\tau}(x, w, \pi^*), \tag{24}$$

for all $N \geq \bar{N}$, which proves (19). ■

III. MILP Formulation

In this section, an MILP is proposed that solves (11), where, by Theorem 1, the average first exit-time of a solution to (11) is arbitrarily close to the average first exit-time of a solution to problem (4) for a sufficiently large N.

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

$$\mathcal{U}_N = \{u^{\eta} \in U_{t^{\eta}} : \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}. \tag{25}$$

Moreover, a given $\mathcal{U}_N$ defines a control policy $\pi_{\mathcal{U}_N}$ according to

$$\pi_{\mathcal{U}_N}(x^{\eta}, w^{\eta}, t^{\eta}) = u^{\eta} \in \mathcal{U}_N, \tag{26}$$

for each $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$ and x^{η} satisfying (6) where $u^{\text{pre}(\eta)} \in \mathcal{U}_N$. Likewise, a control policy $\pi_N^* \in \Pi$ defines a set of control inputs for a given tree $\mathcal{T}_N$ by

$$\mathcal{U}_N(\pi_N^*) = \{u^{\eta} = \pi_N^*(x^{\eta}, w^{\eta}, t^{\eta}) : \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}, \tag{27}$$

where x^{η} satisfies (6) for $u^{\text{pre}(\eta)} \in \mathcal{U}_N(\pi_N^*)$.

Using (5) [see (29e)], (6) [see (29b)], and (25) [see (29a)], the MILP for a given tree $\mathcal{T}_N$ is stated in (29) below, where

$$\mathcal{D}_N = \{\delta^{\eta} \in \{0, 1\} : \eta \in \mathcal{T}_N\}, \tag{28}$$

denotes the set of δ^{η} values for the tree $\mathcal{T}_N$, M is a large positive number, $\mathbf{1}$ denotes the n-dimensional row vector of ones, and the control constraints $u^{\eta} \in U_{t^{\eta}}$ for all $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$ are satisfied due to (25). The next result states conditions for the existence of a solution to MILP (29).

Lemma 2: For a given $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, suppose $M > 0$ is sufficiently large such that $C_{t^{\eta}} x^{\eta} \leq b_{t^{\eta}} + \mathbf{1}M$ for all $\eta \in \mathcal{T}_N$

and x^η according to (29b) for any $\mathcal{U}_N$. Then a solution to MILP (29) 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 $\delta^\eta \in \{0,1\}$, the number of possible sets $\mathcal{D}_N$ is finite. Furthermore, $\rho^\xi \in [0,1]$ for all $\xi \in \mathcal{T}_N$. Thus, a feasible solution exists for at least one of the $\mathcal{D}_N$ sets and the existence of a solution to MILP (29) follows. ∎

$$\min_{\mathcal{U}_N,\mathcal{D}_N} \sum_{\eta\in\mathcal{T}_N} \sum_{\xi\in\mathcal{K}_N^\eta} \delta^\eta \rho^\xi \quad \text{s.t.} \tag{29a}$$

$$x^\eta = A_{t^{\mathrm{pre}(\eta)}} x^{\mathrm{pre}(\eta)} + B_{t^{\mathrm{pre}(\eta)}} u^{\mathrm{pre}(\eta)} + w^{\mathrm{pre}(\eta)}, \quad \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{29b}$$

$$\delta^\eta \geq \delta^{\mathrm{pre}(\eta)}, \text{ for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{29c}$$

$$\delta^\eta \in \{0,1\} \subset \mathbb{Z}, \text{ for all } \eta \in \mathcal{T}_N \tag{29d}$$

$$C_{t^\eta} x^\eta \leq b_{t^\eta} + \mathbf{1} M \delta^\eta, \text{ for all } \eta \in \mathcal{T}_N. \tag{29e}$$

The following theorem shows that, under suitable assumptions and based on (26) and (27), a solution to MILP (29) is equivalent to a solution to (11).

Theorem 2: Suppose Assumptions 1 and 2 hold and M is sufficiently large as in Lemma 2. Then $\mathcal{U}_N^*$ is a solution to MILP (29) if the control policy $\pi_{\mathcal{U}_N^*}$ according to (26) is a solution to (11). Likewise, $\pi_N^* \in \Pi$ is a solution to (11) if $\mathcal{U}_N(\pi_N^*)$ according to (27) is a solution to MILP (29).

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 (11). Thus,

$$\bar{\tau}_N(x,w,\pi_N^*) \geq \bar{\tau}_N(x,w,\pi_N^\#), \tag{30}$$

for all $\pi_N^\# \in \Pi$. A solution to MILP (29) exists due to the assumptions and Lemma 2. Using (27), fix $\mathcal{U}_N = \mathcal{U}_N(\pi_N^*)$ in MILP (29) 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 (29c)–(29e), 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 (9), it follows that

$$\tau_N^\eta(x,w,\pi_N^*) = t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi*} \tag{31a}$$

$$\tau_N^\eta(x,w,\pi_N^\#) = t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi\#}, \tag{31b}$$

for all $\eta \in \mathcal{S}_N$. Then, using (10), (30), and (31), one obtains

$$\begin{aligned}\sum_{\eta\in\mathcal{S}_N} (t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi*})\rho^\eta &= \bar{\tau}_N(x,w,\pi_N^*) \\ \geq \bar{\tau}_N(x,w,\pi_N^\#) &= \sum_{\eta\in\mathcal{S}_N} (t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi\#})\rho^\eta.\end{aligned} \tag{32}$$

Consequently,

$$\sum_{\eta\in\mathcal{S}_N} \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi*}\rho^\eta \leq \sum_{\eta\in\mathcal{S}_N} \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi\#}\rho^\eta. \tag{33}$$

By (12) and (13), $\eta \in \mathcal{S}_N$ and $\xi \in \mathcal{H}_N^\eta$ iff $\xi \in \mathcal{T}_N$ and $\eta \in \mathcal{K}_N^\xi$. Therefore, (33) is equivalent to

$$\sum_{\xi\in\mathcal{T}_N} \sum_{\eta\in\mathcal{K}_N^\xi} \delta^{\xi*}\rho^\eta \leq \sum_{\xi\in\mathcal{T}_N} \sum_{\eta\in\mathcal{K}_N^\xi} \delta^{\xi\#}\rho^\eta, \tag{34}$$

which shows that $\mathcal{U}_N(\pi_N^*), \mathcal{D}_N^*$ is a solution to MILP (29). 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 (29), which exists by Lemma 2, where $\mathcal{D}_N^* = \{\delta^{\eta*} \in \{0,1\} : \eta \in \mathcal{T}_N\}$. Hence,

$$\sum_{\eta\in\mathcal{T}_N} \sum_{\xi\in\mathcal{K}_N^\eta} \delta^{\eta*}\rho^\xi \leq \sum_{\eta\in\mathcal{T}_N} \sum_{\xi\in\mathcal{K}_N^\eta} \delta^{\eta\#}\rho^\xi, \tag{35}$$

for any $\mathcal{U}_N = \mathcal{U}_N^\#$ fixed in MILP (29) with corresponding solution $\mathcal{D}_N^\# = \{\delta^{\eta\#} \in \{0,1\} : \eta \in \mathcal{T}_N\}$. Now define $\pi_{\mathcal{U}_N^*}$ according to (26). Since the dynamics in (1) and (29b) are equivalent, it follows from (9) and (29c)–(29e) that, for each $\eta \in \mathcal{S}_N$,

$$\begin{aligned}\tau_N^\eta(x,w,\pi_{\mathcal{U}_N^*}) &= \min\{\min\{t^\xi \in \mathbb{Z}_{[0,t^\eta]} : \\ &\qquad \delta^{\xi*} = 1, \xi \in \mathcal{H}_N^\eta\} \cup \{t^\eta + 1\}\} \\ &= t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi*}.\end{aligned} \tag{36}$$

Thus, by (10), the average first exit-time of tree $\mathcal{T}_N$ with control policy $\pi_{\mathcal{U}_N^*}$ is given by

$$\bar{\tau}_N(x,w,\pi_{\mathcal{U}_N^*}) = \sum_{\eta\in\mathcal{S}_N} (t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi*})\rho^\eta. \tag{37}$$

In analogy, define $\pi_{\mathcal{U}_N^\#}$ according to (26). Hence,

$$\bar{\tau}_N(x,w,\pi_{\mathcal{U}_N^\#}) = \sum_{\eta\in\mathcal{S}_N} (t^\eta + 1 - \sum_{\xi\in\mathcal{H}_N^\eta} \delta^{\xi\#})\rho^\eta. \tag{38}$$

Using (12) and (13), it follows from (35), (37), and (38) that

$$\begin{aligned}&\bar{\tau}_N(x,w,\pi_{\mathcal{U}_N^*}) - \bar{\tau}_N(x,w,\pi_{\mathcal{U}_N^\#}) \\ &\quad = \sum_{\eta\in\mathcal{S}_N} \sum_{\xi\in\mathcal{H}_N^\eta} (\delta^{\xi\#} - \delta^{\xi*})\rho^\eta \\ &\quad = \sum_{\xi\in\mathcal{T}_N} \sum_{\eta\in\mathcal{K}_N^\xi} (\delta^{\xi\#} - \delta^{\xi*})\rho^\eta \geq 0,\end{aligned} \tag{39}$$

implying that $\pi_{\mathcal{U}_N^*}$ is a solution to (11). ∎

IV. SMPC Strategy

A. 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 (26), maximizes the average first exit-time $\bar{\tau}_N$ for a given $\mathcal{T}_N$ (Theorem 2) and achieves average first exit-times $\bar{\tau}$ arbitrarily close to the optimal value of problem (4) for sufficiently large N (Theorem 1). 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 included in $\mathcal{T}_N$.

Therefore, an SMPC scheme is proposed using MILP (29), 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 this case, the stochastic linear model in (1) and the Markov chain for w_t serve 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 SMPC scheme defines the following control policy $\pi_{\mathrm{SMPC},N} \in \Pi$,

$$\pi_{\mathrm{SMPC},N}(x, w, t_0) = u^{\eta_0} \in \mathcal{U}_N^*, \tag{40}$$

where $\mathcal{U}_N^*$ is a solution to MILP (29) for the scenario tree $\mathcal{T}_N$ with root node η_0 and $t^{\eta_0} \leftarrow t_0$, $x^{\eta_0} \leftarrow x$, and $w^{\eta_0} \leftarrow w$ in Step 2 of Algorithm 1. It follows from Theorems 1 and 2 that, in terms of first exit-time performance, $\pi_{\mathrm{SMPC},N}$ in (40) is arbitrarily close to a solution (assuming one exists) of problem (4) for sufficienlty large N. This is summarized in Theorem 3.

Theorem 3: Suppose Assumptions 1 and 2 hold, $\pi_{\mathrm{SMPC},N}$ is as in (40), 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 $\bar{\tau}(x, w, \pi_{\mathrm{SMPC},N}) + \varepsilon \geq \max_{\pi \in \Pi} \bar{\tau}(x, w, \pi)$ for all $N \geq \bar{N}$.

B. Implementation

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: $t_{\mathrm{comp}} \leftarrow 0$
5: **while** computing solution of MILP (29) **do**
6:     **if** $t_{\mathrm{comp}} > t_{\max}$ **then**
7:         go to Step 12
8:     **end if**
9:     $t_{\mathrm{comp}} \leftarrow$ update t_{comp}
10: **end while**
11: $\mathcal{U}_N^* \leftarrow$ solution of MILP (29); go to Step 13
12: $\mathcal{U}_N^* \leftarrow$ solution of LP (41)
13: $u(t) \leftarrow$ apply control $u^{\eta_0} \in \mathcal{U}_N^*$ to the system
14: $t \leftarrow t + 1$; go to Step 2

In practice, the SMPC strategy may be implemented as in Algorithm 2. 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 (29) is computed. Since MILP is NP-complete [9], [10] and computing a solution may take considerably long in the worst-case, an upper bound $t_{\max}$ on the MILP computation time is specified. If the computation time t_{comp} is greater than $t_{\max}$, computing an MILP solution is terminated (Steps 6–8) and a relaxed version of the MILP, a standard linear program (LP), is solved instead. The LP for a given tree $\mathcal{T}_N$ is obtained by replacing the integer variables δ^η in MILP (29) by non-negative real variables ε^η for all $\eta \in \mathcal{T}_N$. Thus, the LP is as follows

$$\min_{\mathcal{U}_N, \mathcal{E}_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \varepsilon^\eta \rho^\xi \quad \text{s.t.} \tag{41a}$$

$$x^\eta = A_{t^{\mathrm{pre}(\eta)}} x^{\mathrm{pre}(\eta)} + B_{t^{\mathrm{pre}(\eta)}} u^{\mathrm{pre}(\eta)} + w^{\mathrm{pre}(\eta)}, \quad \text{for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{41b}$$

$$\varepsilon^\eta \geq \varepsilon^{\mathrm{pre}(\eta)} \geq 0, \text{ for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{41c}$$

$$C_{t^\eta} x^\eta \leq b_{t^\eta} + \mathbf{1}\varepsilon^\eta, \text{ for all } \eta \in \mathcal{T}_N, \tag{41d}$$

where $\mathcal{U}_N$ is as in (25) and a set of ε^η values for a tree $\mathcal{T}_N$ is denoted by $\mathcal{E}_N = \{\varepsilon^\eta \geq 0 : \eta \in \mathcal{T}_N\}$. Note that a solution to LP (41) always exists because $\varepsilon^\eta \geq 0$ can always be chosen sufficiently large such that (41d) is satisfied for all $\eta \in \mathcal{T}_N$.

The root node control input u^{η_0} of the MILP solution $\mathcal{U}_N^*$ (or the LP solution in case $t_{\mathrm{comp}} > t_{\max}$) is applied to the system in Step 13 of Algorithm 2 and the procedure is repeated at the next time instant $t + 1$.

V. Numerical Case Studies

Numerical case studies of problems of the form (4) are treated using the SMPC strategy given by Algorithm 2. The first case study in Section V-A considers a second-order linear system and investigates the influence of the number of tree nodes on the solution. In the second case study (Section V-B), the ACC problem that was solved with dynamic programming (DP) techniques in [11] is solved with the SMPC strategy and results are compared. In both case studies, the scenario tree for each $w^i \in W$ is precomputed offline, where $\mathcal{T}_N(w^i)$ denotes the scenario tree corresponding to the initial disturbance $w^{\eta_0} = w^i$. Hence, $\mathcal{T}_N \leftarrow \mathcal{T}_N(w)$ in Step 3 of Algorithm 2 instead of constructing $\mathcal{T}_N$ at each time instant.

All computations involving the SMPC strategy are performed in MATLAB 2015a on a laptop with an i5-6300 processor and 8 GB RAM. The Hybrid Toolbox [12] (default settings) is used to solve LPs and MILPs.

A. Influence of Number of Tree Nodes

In this case study, the influence of N on the solution is investigated, where a tree $\mathcal{T}_N$ contains $N + 1$ nodes. The following stochastic linear time-varying system is considered

$$\begin{bmatrix} r_{1,t+1} \\ r_{2,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 1.2 \end{bmatrix} \begin{bmatrix} r_{1,t} \\ r_{2,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5\sin(t/2) \end{bmatrix} u_t + \begin{bmatrix} 0 \\ w_t \end{bmatrix}, \tag{42}$$

where $x = [r_1, r_2]^\top$ denotes the state vector and the control input is $u \in [-1, 1]$. The constraints for the optimal control problem (4) are given by the set $G_t \equiv \{x : -2 \leq r_1 \leq 2, -2 \leq r_2 \leq 2\}$.

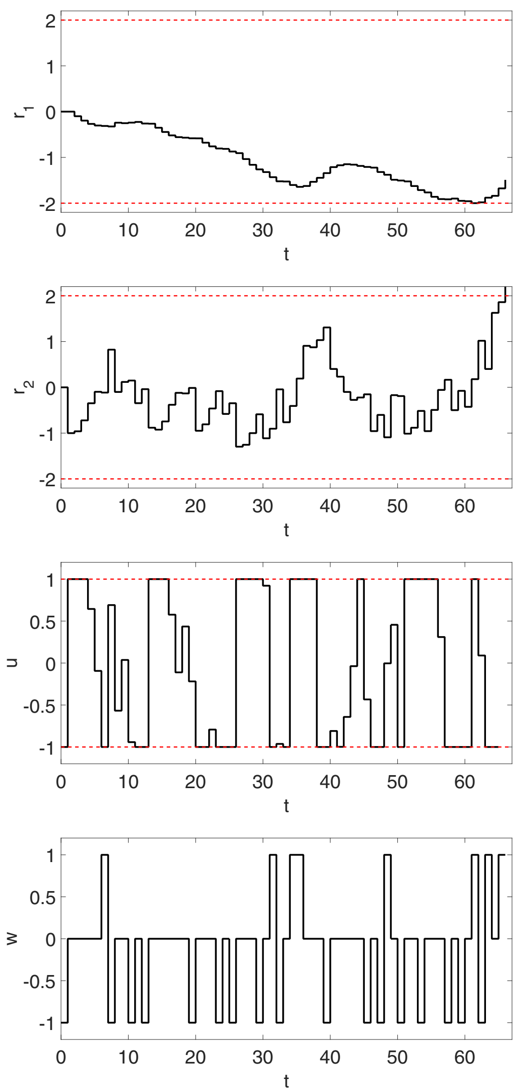

Fig. 2: Numerical case study on the influence of the number of tree nodes: sample trajectories showing the states r_1 (top plot) and r_2 (second plot) as well as the control input u (third plot) and disturbance w (bottom plot) vs. t.

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

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

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

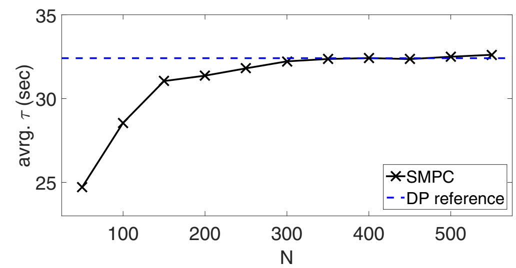

Fig. 3: Numerical case study on the influence of the number of tree nodes: average first exit-time $\bar{\tau}$ vs. N (for 1000 random simulations for each N).

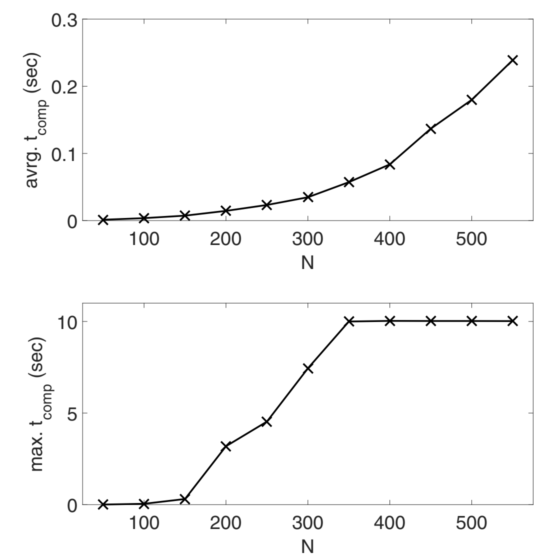

Fig. 4: Numerical case study on the influence of the number of tree nodes: average (top) and worst-case time (bottom) to compute control u_t (Steps 2–13 in Algorithm 2) vs. N (for 1000 random simulations for each N).

The average first exit-time $\bar{\tau}$ (1000 random simulations for each N) is plotted against N in Figure 3. For comparison, a DP solution with conventional value iteration [13] applied to a discrete grid of the state space using linear interpolation between the grid points (the set defining the control constraints is discretized as well, using an equidistant grid with 21 points) is shown as a reference in Figure 3 (dashed blue line), achieving $\bar{\tau} = 32.41$ sec. This DP solution is obtained for a relatively dense grid of 900000 points, which requires about 1.63 hours to compute the control policy offline when implemented in C on a desktop computer. Due to the dense grids (for both G_t and U_t), the DP reference solution is expected to be close to a solution of the optimal control problem (4).

In line with Theorem 3, it can be seen in Figure 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 value is slightly exceeded for $N \geq 500$. Note that in this case, the average first exit-time achieved by the SMPC strategy appears to be monotonically non-decreasing when increasing N, which may not hold in general. The computation time (in MATLAB) of the SMPC scheme for computing the control input at each time instant according to Algorithm 2 (Steps 2–13) is shown in Figure 4 for different N. The top plot in Figure 4 shows the average computation time, which increases nearly exponentially with N. The worst-case / maximum computation time is shown in Figure 4 (bottom), where the prescribed limit on the MILP computation time $t_{\max} = 10$ sec is reached for $N \geq 400$.

B. Car Following – Adaptive Cruise Control

The same ACC problem (same model, constraints, initial condition, etc.) as in [11] is solved with the proposed SMPC scheme. The results are compared against the DP-based solution from [11], which achieves an average first exit-time of $\bar{\tau} = 2591$ sec (for 1000 random simulations). Note that the simulation model in this example is a stochastic hybrid model with state-dependent probabilities for mode switches since there is a 10 % chance of another vehicle cutting in upfront if the time gap T_{g} between the two vehicles is greater than 2.2 sec, see [11]. The DP approach in [11] is able to explicitly consider such hybrid models. On the other hand, the SMPC strategy assumes a linear model with an additive random disturbance modeled by a Markov chain and neglects the possibility of another vehicle cutting in upfront. It compensates for the unmodeled effects as MPC provides feedback (see Algorithm 2). With N set to 100, the SMPC strategy achieves an average first exit-time of $\bar{\tau} = 3120$ sec (1000 random simulations), which is an improvement of 20 % compared to the DP solution. The DP solution can be improved by using denser state space discretizations, which, however, would increase computation times exponentially (curse of dimensionality). For the SMPC strategy, on average, 5 msec are required to compute the control input at each time instant and 60 msec in the worst-case. Sample trajectories of the time gap between the follower and lead vehicle and of the follower vehicle velocity are shown in Figure 5, where the dashed red lines indicate the prescribed constraints.

VI. Conclusion

The optimal control problem of maximizing the average time before prescribed state and control constraints are violated for the first time was considered for stochastic linear systems with an additive disturbance. The disturbance scenarios were encoded by a tree structure with a specified number of tree nodes and the tree generation algorithm has been defined to emphasize the inclusion of the most relevant scenarios. Based on the tree structure, a stochastic model predictive control (SMPC) strategy was proposed that, for sufficiently large trees, obtains solutions arbitrarily close to the optimal solution by repeatedly solving a mixed-integer linear program over a receding time horizon based on the current state and disturbance. The effectiveness of the SMPC strategy was demonstrated in two numerical case studies, including a stochastic adaptive cruise control problem.

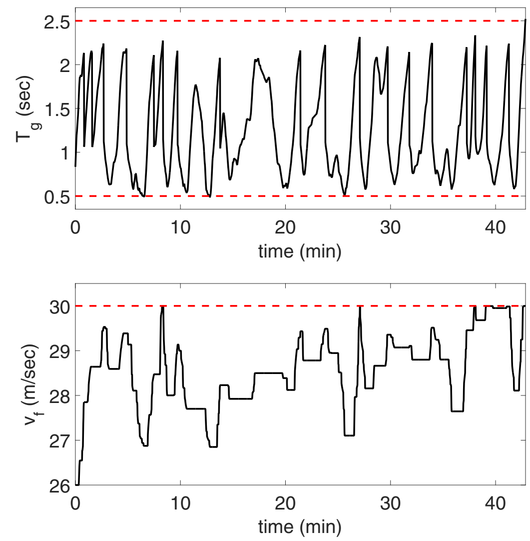

Fig. 5: ACC problem: sample trajectories over time of time gap T_{g} between follower and lead vehicle (top) and follower vehicle velocity v_{f} (bottom).

References

[1] R. A. E. Zidek and I. V. Kolmanovsky, "Optimal driving policies for autonomous vehicles based on stochastic drift counteraction," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 290–296, 2017.

[2] R. A. E. Zidek, A. Bemporad, and I. V. Kolmanovsky, "Optimal and receding horizon drift counteraction control: Linear programming approaches," in *American Control Conference (ACC)*, 2017, pp. 2636–2641.

[3] P. D. Couchman, M. Cannon, and B. Kouvaritakis, "Stochastic MPC with inequality stability constraints," *Automatica*, vol. 42, no. 12, pp. 2169–2174, 2006.

[4] M. Cannon, B. Kouvaritakis, and X. Wu, "Probabilistic constrained MPC for multiplicative and additive stochastic uncertainty," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1626–1632, 2009.

[5] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2014.

[6] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems*, vol. 36, no. 6, pp. 30–44, 2016.

[7] A. K. Sampathirao, P. Sopasakis, A. Bemporad, and P. Patrinos, "Distributed solution of stochastic optimal control problems on GPUs," in *54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 7183–7188.

[8] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2012.

[9] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.

[10] A. Richards and J. How, "Mixed-integer programming for control," in *Proceedings of the 2005, American Control Conference, 2005.*, 2005, pp. 2676–2683.

[11] R. A. E. Zidek and I. V. Kolmanovsky, "Stochastic drift counteraction optimal control and enhancing convergence of value iteration," in *55th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 1119–1124.

[12] A. Bemporad, "Hybrid Toolbox - User's Guide," 2004, http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox.

[13] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I.* Belmont, MA: Athena Scientific, 2005.