Safe High-Performance Autonomous Off-Road Driving Using Covariance Steering Stochastic Model Predictive Control

Jacob Knaup, Kazuhide Okamoto, and Panagiotis Tsiotras, *Fellow, IEEE*

***Abstract*— Autonomous racing is a high-performance, safety-critical task that inherently involves a high degree of uncertainty (especially in off-road unstructured environments), as driving conditions can vary and tire-terrain interactions are difficult to model accurately. On the one hand, the vehicle needs to drive fast while, on the other hand, it must avoid crashing, thus requiring a tradeoff between performance and safety. This work develops a stochastic model predictive controller (SMPC) for uncertain systems with additive Gaussian noise subject to state and control constraints and applies it to off-road autonomous racing. The proposed approach is based on the recently developed finite-horizon optimal covariance steering (CS) control theory, which steers the system state's mean and covariance to prescribed target values at a given terminal time. We show that the proposed CS-SMPC algorithm can deal with unbounded Gaussian additive noise while ensuring stability. The effectiveness of the proposed approach is demonstrated via both numerical and experimental tests using a scaled autonomous racing platform, as well as on an actual full-size vehicle during a global positioning system (GPS)-denied autonomous driving task.**

***Index Terms*— Autonomous racing, collision avoidance, model predictive control (MPC), stochastic optimal control.**

I. Introduction

AUTONOMOUS racing has generated a great deal of interest in the robotics research community [1], [2], [3] as it presents a uniquely challenging, yet structured, test-bed for control algorithms. The task demands fast real-time computation to control a complex dynamical system, which can be subject to multiple input and state constraints and stochastic disturbances. Model predictive control (MPC) is an attractive method for solving such control problems because it is able to provide theoretical guarantees on stability and feasibility subject to constraints, while recent advances in computer hardware have rendered MPC approaches amenable to real-time implementation [4], [5]. Within the field of MPC, the two primary methods for solving the optimal control problem are convex optimization and sampling-based methods. The first method relies on linearizing the dynamics and writing the cost function in a convex form [6], while the second method is able to work directly with nonconvex problems and general dynamics by sampling the control actions directly [7].

Since MPC is a model-based design, deterministic MPC approaches are susceptible to errors owing to modeling uncertainties and exogenous disturbances. This is especially critical to autonomous racing tasks, where the vehicle dynamics are often not known exactly because tire-road interactions are difficult to model and change with varying road conditions, and the vehicle may be subjected to large external disturbances from the interaction with the terrain. In order to overcome difficulties stemming from the susceptibility of classical MPC methods to environmental uncertainty and noise, robust MPC (RMPC) and stochastic MPC (SMPC) extensions have been developed (see [4], [8], [9], [10] for an extensive literature review) to deal with various forms of uncertainty.

RMPC approaches assume deterministic uncertainties that lie within a given compact set. For example, min–max MPC [11] computes a control command that can deal with the worst case scenario in terms of system uncertainty. Another RMPC approach is the tube-MPC [12], which separates the controller into a nominal controller and a feedback controller, with the latter being proportional to the deviation from the nominal state value, thus achieving asymptotic stability to a set [13]. RMPC approaches are effective against worst case deterministic disturbances but can be conservative in the case of stochastic disturbances since they ignore any knowledge about the probabilistic nature of the disturbance. In addition, they do not guarantee recursive feasibility in the case of possibly unbounded disturbances. Recursive feasibility is a crucial property that ensures that the MPC optimization problem has a solution (i.e., satisfies all the constraints) at each time step [5], [14].

In order to explicitly deal with the probability characteristics of system uncertainties, several SMPC approaches have been developed. As with RMPC, in SMPC feedback *policies* are optimized instead of open-loop control sequences. However,

Manuscript received 17 September 2022; revised 12 March 2023; accepted 22 April 2023. Date of publication 21 July 2023; date of current version 18 August 2023. Recommended by Associate Editor M. Cannon. This work was supported in part by the Office of Naval Research (ONR) under Award N00014-18-1-2828, in part by NSF under Award CPS-1544814, and in part Autonomous Solutions Inc., through Small Business Innovation Research (SBIR) under Contract W56KGU-20-C-0003. *(Corresponding author: Panagiotis Tsiotras.)*

Jacob Knaup is with the School of Interactive Computing, College of Computing and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: jacobk@gatech.edu).

Kazuhide Okamoto is with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: kazuhide@gatech.edu).

Panagiotis Tsiotras is with the School of Aerospace Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: tsiotras@gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2023.3291570.

Digital Object Identifier 10.1109/TCST.2023.3291570

SMPC abandons the worst case point of view addressing, instead, the desired "expected" or "average" system behavior that takes into consideration the most likely disturbance the system may encounter in practice. As a result, SMPC methods tend to provide better performance.

Although there is no agreed consensus about classifying the numerous SMPC approaches proposed in the literature [4], the most common approaches are the so-called analytic approaches and the randomized (or scenario-based) approaches. The former include stochastic-tube [15], [16], [17] and affine-parameterization [18], [19] approaches, which reformulate the cost and the probabilistic constraints in deterministic terms. They typically assume some form of additive white Gaussian noise acting on the system and are most closely related to the proposed CS-SMPC algorithm in this article. Scenario approaches, such as [20] and [21], on the other hand, compute the expected future system behavior by generating several noise realizations. For this reason, they can handle more generic systems, costs, and state and control constraints. However, their computational requirements are much higher than analytic approaches. In addition, their feasibility and convergence properties are difficult to access. Scenario-based SMPC approaches will not be discussed further in this work.

From the analytic approaches, stochastic-tube MPC [15], [16], [22], [23] decomposes the state to deterministic and random components. The random component is controlled using a state feedback controller with a *precomputed* stabilizing gain, and only the additional control command to steer the deterministic component is computed online. By doing so, stochastic-tube MPC succeeds in avoiding the optimization over arbitrary feedback policies. Although this approach reduces computational complexity, it requires trial and error to compute a priori a state feedback gain that is not too conservative, especially when constraints are expected to be active.

In order to overcome the offline computation of the feedback gain, an affine parameterization SMPC approach has been proposed [18], [19]. In this approach, both the feedback gain and the deterministic component are design variables that have to be simultaneously optimized online. The stochastic-tube MPC approach predicts the future state uncertainty evolution a priori because the feedback gain is precomputed; it tries to control the mean state so that the predicted state satisfies the given constraints at the end of the horizon. On the other hand, the affine parameterization approach simultaneously computes the feedback gains and the open-loop control sequences so that the predicted state solution satisfies the constraints. As a result, the affine parameterization approach leads to less conservative controllers that tend to operate closer to the boundary of the constraints, thus increasing performance.

It is known that a state feedback parameterization approach leads to a nonconvex problem [24], [25], [26]. In practice, one, thus, relaxes the constraints to make the problem convex, which may lead to unnecessarily conservative results. As an alternative, Oldewurtel et al. [19] employ a disturbance feedback parameterization of the control policy instead, which leads to a convex problem formulation. It has been shown that disturbance feedback parameterization is equivalent to state feedback parameterization [27]. The disturbance feedback parameterization approach has been extended to accommodate input hard constraints in [18], [28], [29], and [30], a task that is difficult to satisfy using stochastic tube-MPC or state feedback parameterization approaches.

Owing to the problem stochasticity, SMPC approaches use probabilistic (e.g., chance) constraints and impose a maximum probability of state or input constraint violation [31], [32], instead of an absolute constraint violation requirement. Satisfaction of the constraints with high probability is the price to pay for being able to explicitly handle uncertainty in the problem formulation. When using the stochastic tube-MPC approach (and since the feedback gain is precomputed) the chance constraints can be converted to linear inequality constraints, whereas in the affine parameterization approach (and since the feedback gains are computed online) the chance constraints are converted to second-order cone constraints [4].

A. Challenges of Current Stochastic MPC Approaches

MPC controller design is based on a receding horizon approach, by which an optimization problem is solved repeatedly every time a new measurement of the state becomes available. As a result, a key requirement for the success of an MPC design is the ability to satisfy the constraints at each iteration, and a property known as recursive feasibility. While the issue of recursive feasibility is well understood for the case of deterministic MPC, showing the same for SMPC is much more challenging. This is because, in general, it is not possible to enforce recursive satisfaction of the state and control constraints in the face of *unbounded* additive disturbances. Therefore, most works on SMPC achieve recursive feasibility and convergence by assuming a bounded probability distribution of the disturbance [19], [22], [33], [34], [35].

Not surprisingly, SMPC theory dealing with unbounded (e.g., Gaussian) disturbances is less developed. Some recent work includes [25], [30], [36]. The approach proposed in [25] devises a reinitialization strategy that switches between closed-loop and open-loop control to ensure recursive feasibility. It also uses state-feedback parameterization, which leads to the need to solve a nonconvex programming problem. Paulson et al. [30] consider the MPC problem for stochastic linear systems with arbitrary, possibly unbounded, disturbances subject to both joint state chance constraints and hard input constraints. Contrary to [25], Paulson et al. [30] do not use a reinitialization strategy to ensure recursive feasibility and stability, but instead, they suggest softening the chance constraints. Paulson et al. [36] address the SMPC problem for additive Gaussian noise and time-invariant probabilistic system uncertainty using state feedback parameterization. Since guaranteeing feasibility and stability under both unbounded noise and system uncertainty is very challenging, stability is established for the unconstrained case only.

Farina et al. [25] consider a time-invariant linear system and design a terminal set using a precomputed feedback matrix, which is the steady-state solution of a Lyapunov equation. However, in the case of a time-varying system, it is not possible to compute such a steady-state solution.

Other prior works addressing the problem of stability and recursive feasibility of SMPC for linear time-varying systems are [33] and [37]. However, in both cases, the authors limit their analysis to systems with bounded disturbances and only establish stability for the nominal system.

The main challenge in all current SMPC formulations stems from the difficulty of controlling the dispersion of the trajectories subject to system uncertainty, which makes ensuring recursive feasibility and stability elusive, especially in the case of stochastic disturbances with unbounded support. The approach we propose in this article utilizes the results from the newly developed finite-horizon optimal covariance steering (CS) theory [38], [39] that allow the distribution of trajectories subject to Gaussian noise to be driven from an initial distribution to a specified final distribution while minimizing a state and control expectation-dependent cost.

While infinite-horizon covariance control has been researched extensively since the late 1980s [40], the finite-horizon CS case had not been investigated until very recently [41], [42], [43]. Specifically, and most closely related to the results in this article, in our previous work [38], we introduced state chance constraints into the optimal CS problem and used it to solve challenging path planning problems in the presence of uncertainty [39]. To the best of our knowledge, this article is the first work in the literature to incorporate optimal CS into the SMPC framework in order to deal with the issue of convergence of SMPC in a principled manner. Since the proposed CS-SMPC approach simultaneously computes the open loop and the feedback gains, it can be regarded as an affine parameterization approach.

B. Contributions

The main contributions of this work are summarized as follows: first, we introduce a new SMPC approach for linear time-varying systems with unbounded noise that takes advantage of the recent theory of CS to provide direct control of the state covariance at the end of the horizon, thus enabling the analysis of the stability properties of the system. Second, we introduce the covariance steering stochastic MPC (CS-SMPC) algorithm and demonstrate how the design of the feedback policy, along with the terminal constraints, achieves computational efficiency and the ability to deal with unbounded Gaussian disturbances, while also ensuring stability, provided the problem remains feasible. Owing to space limitations, we address only the stability question, referring to the proof of recursive feasibility for a future publication. As discussed previously, existing approaches require either relaxing the initialization policy in order to ensure feasibility in the presence of unbounded disturbances, or they assume that feasibility is already established. In this article, we follow a similar approach. In fact, in case of disturbances with unbounded support (e.g., Gaussian noise) one can only guarantee recursive feasibility with high probability, as there always exists the possibility that a large enough disturbance will drive the state into the infeasible region regardless of the control policy. In practice, of course, disturbances are bounded. Nonetheless, dealing with unbounded disturbances removes the need to specify a prior bound on the disturbances, which typically increases the conservatism of the ensuing control policy. We demonstrate the safety advantages of the proposed control design for a stochastic system in an autonomous racing task using a 1:5 scale AutoRally autonomous vehicle on a dirt race track. Finally, we present the results of the implementation of the CS-SMPC controller on an actual full-size commercial vehicle during a normal, self-driving task.

Notation: The notation used in this article is quite standard. We denote the set of n-dimensional real vectors and $m \times n$ real matrices by $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$, respectively. We use $P \succ 0$ and $P \succeq 0$ to denote the fact that the matrix P is symmetric positive definite and semidefinite, respectively. $\mathtt{tr}(P)$ denotes the trace of the square matrix P, and $\mathtt{blkdiag}(P_0, \ldots, P_N)$ denotes the block-diagonal matrix with matrices $P_0, \ldots, P_N$. We use $(\star)$ to represent repeated terms. $\|v\|$ is the two-norm of the vector v, and $I_d \in \mathbb{R}^{d\times d}$ is the identity matrix of size d. The notation $x_{t|k}$ indicates the state at time step t predicted at time step $k \geq 0$, where $t \geq k$. The notation $x \sim \mathcal{N}(\mu, \Sigma)$ indicates that the random variable x is distributed according to a Gaussian with mean μ and (co)variance Σ. Finally, $\mathbb{E}[\cdot]$ denotes the expectation operation, and $\Pr(A)$ denotes the probability of the event A.

The rest of this article is structured as follows. In Section II, we define the stochastic optimal control problem we wish to solve, and in Section III, we develop the optimal CS problem and provide its solution. In Section IV, we present the CS-SMPC algorithm along with its stability properties. In Section V, we discuss the implementation of the CS-SMPC algorithm, including the development of a high-fidelity vehicle model, used for control design and simulation. Section VI then presents simulation and experimental results on an actual vehicle platform demonstrating the safety and performance advantages of CS-SMPC over a state-of-the-art sampling-based MPC controller. For conciseness, some of the more lengthy proofs are given in the Appendix.

II. Problem Statement

In this section, we formulate the general SMPC problem. We first provide the class of dynamical systems the approach can handle, and we then introduce the control design problem we wish to solve.

A. Problem Formulation

We consider discrete-time-varying stochastic affine systems with additive uncertainty of the form

$$x_{k+1} = A_k x_k + B_k u_k + D_k w_k + r_k, \quad k = 0, 1, \ldots \tag{1}$$

where k is the time step, $A_k \in \mathbb{R}^{n_x\times n_x}$, $B_k \in \mathbb{R}^{n_x\times n_u}$, $D_k \in \mathbb{R}^{n_x\times n_w}$, and $r_k \in \mathbb{R}^{n_x}$, $x \in \mathbb{R}^{n_x}$ is the state, with initial state x_0 known, $u \in \mathbb{R}^{n_u}$ is the control input, and $w \in \mathbb{R}^{n_w}$ is a zero-mean white Gaussian noise with unit covariance, that is,

$$\mathbb{E}[w_k] = 0, \quad \mathbb{E}\big[w_{k_1} w_{k_2}^\top\big] = \begin{cases} I_{n_w}, & \text{if } k_1 = k_2 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

We also assume that

$$\mathbb{E}\left[x_{k_1} w_{k_2}^\top\right] = 0, \quad 0 \le k_1 \le k_2 \tag{3}$$

which stems from causality considerations.

In (1), the system matrices A_k, B_k, D_k, and r_k are assumed to be known and bounded for all $k = 0, 1, \ldots$. The case when these matrices are not known exactly or they depend on some random parameters is out of the scope of this article, but is an important problem in its own right. Probabilistic MPC approaches that can also handle parametric uncertainties in the system matrices have been proposed in the literature; see [10] and the references therein.

We require that the state x_k and the control input u_k obey probabilistic (chance) constraints, in the sense that we require, for all $k = 0, 1, \ldots$

$$\Pr(x_k \in \mathcal{X}) \ge 1 - p_x \tag{4a}$$

$$\Pr(u_k \in \mathcal{U}) \ge 1 - p_u \tag{4b}$$

where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ convex, and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ compact and convex. Here, p_x and $p_u \ge 0$ denote the maximum probabilities of constraint violation. In practice, typically, $p_x, p_u \ll 1$.

Given x_k and a control sequence $u_k, u_{k+1}, \ldots, u_{k+N-1}$, where N is a time horizon, we associate with (1) the cost

$$J_N(x_k; \mathbf{u}_k^{N-1}) = \mathbb{E}\left[\sum_{t=k}^{k+N-1} \ell_t(x_t, u_t) + J_f(x_{k+N})\right] \tag{5}$$

where $\ell_t(x, u)$ is the running cost, $J_f(x)$ is a terminal cost, and where, for notational simplicity, we write $\mathbf{u}_k^{N-1} = \{u_k, u_{k+1}, \ldots, u_{k+N-1}\}$. Since we are primarily interested in tracking problems, we assume that the running cost has the special form

$$\ell_t(x, u) = (x - x_{g_t})^\top Q_t (x - x_{g_t}) + u^\top R_t u \tag{6}$$

where, for all $t = 0, 1, \ldots$, $Q_t \succeq 0$, $R_t \succ 0$ are known bounded matrices, and $\{x_{g_t}\}_{t=0}^{\infty} \subset \mathbb{R}^{n_x}$ is a given bounded sequence representing the target state to track. To ensure that the problem has a solution, it is further assumed that $x_{g_t} \in \mathcal{X}$ for all $t = 0, 1, \ldots$.

B. SMPC Optimization Problem

In order to obtain a tractable optimization problem, we will further assume that the sets $\mathcal{X}$ and $\mathcal{U}$ can be written as the intersection of a finite number of linear inequality constraints as follows:

$$\mathcal{X} \triangleq \bigcap_{i=0}^{N_s-1} \left\{x : \alpha_{x,i}^\top x \le \beta_{x,i}\right\} \tag{7}$$

$$\mathcal{U} \triangleq \bigcap_{j=0}^{N_c-1} \left\{u : \alpha_{u,j}^\top u \le \beta_{u,j}\right\} \tag{8}$$

where $\alpha_{x,i} \in \mathbb{R}^{n_x}$ and $\alpha_{u,j} \in \mathbb{R}^{n_u}$ are constant vectors, and $\beta_{x,i}$ and $\beta_{u,j}$ are constant scalars.

Using Boole's inequality [44], the constraint (4a) is satisfied assuming the inequality

$$\Pr\left(\alpha_{x,i}^\top x_k \le \beta_{x,i}\right) \ge 1 - p_{x,i} \tag{9a}$$

holds for all $i = 0, \ldots, N_s - 1$, where $p_{x,i} \ge 0$ are such that

$$\sum_{i=0}^{N_s-1} p_{x,i} \le p_x. \tag{9b}$$

Similarly, the chance constraint (4b) is satisfied by imposing the following conditions for all $j = 0, \ldots, N_c - 1$:

$$\Pr\left(\alpha_{u,j}^\top u_k \le \beta_{u,j}\right) \ge 1 - p_{u,j} \tag{10a}$$

where $p_{u,j} \ge 0$ are such that

$$\sum_{j=0}^{N_c-1} p_{u,j} \le p_u. \tag{10b}$$

In the following, we assume that $p_{x,i}$ and $p_{u,j}$ are given such that they satisfy (9b) and (10b), respectively. The problem of risk allocation [4], [30], [45] allows for $p_{x,i}$ and $p_{u,j}$ to be free parameters to be optimized as well. However, adding $p_{x,i}$ and $p_{u,j}$ as optimization variables results in a nonconvex optimization problem, so the risk allocation problem will not be addressed in this article where we seek computationally efficient (albeit conservative) solutions to the SMPC problem.

The SMPC aims to solve, at each time step k, a finite-horizon optimal control problem with cost as in (5) subject to the dynamics in (1) and the state and control constraints (9) and (10). Let the horizon N be given and let $\mathbf{u}_{k|k}^{N-1} = \{u_{k|k}, \ldots, u_{k+N-1|k}\}$ be a control sequence over the horizon $k, \ldots, k + N - 1$. The SMPC optimization problem at time k can then be summarized as follows:

$$\min_{\mathbf{u}_{k|k}^{N-1}} J_N(x_k; \mathbf{u}_{k|k}^{N-1}) = \mathbb{E}\Big[\sum_{t=k}^{k+N-1} \ell_t(x_{t|k}, u_{t|k}) + J_f(x_{k+N|k})\Big] \tag{11a}$$

$$\text{s.t. } x_{t+1|k} = A_t x_{t|k} + B_t u_{t|k} + D_t w_t + r_t \tag{11b}$$

$$\Pr\left(\alpha_{x,i}^\top x_{t|k} \le \beta_{x,i}\right) \ge 1 - p_{x,i}, \quad i = 0, \ldots, N_s - 1 \tag{11c}$$

$$\Pr\left(\alpha_{u,j}^\top u_{t|k} \le \beta_{u,j}\right) \ge 1 - p_{u,j}, \quad j = 0, \ldots, N_c - 1 \tag{11d}$$

where $t = k, \ldots, k + N - 1$ and $x_{k|k} = x_k$. We introduce the following assumption which will aid the subsequent discussion. This assumption is used for the stability analysis and will be made more precise in Assumption 3.

Assumption 1: For all $k = 0, 1, \ldots$, there exists a feasible control sequence $\mathbf{u}_{k|k}^{N-1}$ that solves problem (11).

Let us denote the optimal solution to (11) as $\mathbf{u}_{k|k}^{*N-1} = \{u_{k|k}^*, \ldots, u_{k+N-1|k}^*\}$. The SMPC generates an (implicit) feedback strategy by applying, at each time step k, the control $u_{k|k}^*$ to the system (1), that is, $u_k = u_{k|k}^*$.

Remark 1: In the problem formulation (11), we have probabilistic constraints for both the state and the control input. In practice, one would prefer to impose hard constraints in terms of the input. Although hard input constraints can be incorporated in SMPC problems [4], enforcing input hard constraints for SMPC problems and guaranteeing stability and feasibility is not possible if the disturbance is unbounded and the system is not Schur stable [28] or Lyapunov stable [29]. This is not a major issue for most engineering applications;

however, since almost sure satisfaction of the control constraint (11d) can be ensured by choosing a very small value of p_u in (10b). For other alternatives, see [26], [46]. For more details on the difference between hard and chance input constraints in SMPC problems, we refer the interested reader to the discussion in [30] and [47].

III. Optimal Covariance Steering

In this section, we introduce the basic theory behind optimal CS controller design under state and control chance constraints, which will be the main ingredient in solving a relaxed version of the SMPC problem (11). The proposed CS-SMPC algorithm modifies the optimization problem (11) in terms of two important aspects. First, the terminal cost in (11a) is replaced by *explicit* constraints in the final state mean and covariance. Second, we allow for the initial state x_k to be drawn from a normal[1] distribution with known mean and covariance. Hence, the mean and covariance of the initial state become additional optimization parameters of the problem, a technique that has been used in other similar SMPC approaches [25], [48].

In the discrete-time optimal CS problem setup [38], we wish to steer the state distribution of system (1) from an initial Gaussian distribution $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$, to a prescribed Gaussian distribution at a given time step N, namely, $x_N = x_f \sim \mathcal{N}(\mu_f, \Sigma_f)$, where $\Sigma_0 \succeq 0$ and $\Sigma_f \succ 0$.

Specifically, we wish to solve the following optimization problem:

$$\min_{\mathbf{u}_0^{N-1}} J(\mu_0, \Sigma_0; \mathbf{u}_0^{N-1}) = \mathbb{E}\left[\sum_{k=0}^{N-1} \ell_k(x_k, u_k)\right] \tag{12a}$$

$$\text{s.t. } x_{k+1} = A_k x_k + B_k u_k + D_k w_k + r_k \tag{12b}$$

$$x_0 \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{12c}$$

$$\Pr\left(\alpha_{x,i}^\top x_k \le \beta_{x,i}\right) \ge 1 - p_{x,i}, \quad i = 0, \dots, N_s - 1 \tag{12d}$$

$$\Pr\left(\alpha_{u,j}^\top u_k \le \beta_{u,j}\right) \ge 1 - p_{u,j}, \quad j = 0, \dots, N_c - 1 \tag{12e}$$

$$x_N \sim \mathcal{N}(\mu_f, \Sigma_f) \tag{12f}$$

where $k = 0, \dots, N-1$ and $\mathbf{u}_0^{N-1} = \{u_0, \dots, u_{N-1}\}$.

As discussed in [38], [39], and [42], by relaxing the terminal constraint (12f) to the pair of convex constraints

$$\mathbb{E}[x_N] = \mu_f, \quad \mathbb{E}\left[(x_N - \mathbb{E}[x_N])(x_N - \mathbb{E}[x_N])^\top\right] \preceq \Sigma_f \tag{13}$$

Problem (12) can be reduced to a convex optimization problem that can be efficiently solved to global optimality using a semidefinite programming solver. Specifically, we have the following theorem, whose proof can be found in [39].

Theorem 1 ([39]): Let the optimization problem (12), where (12f), is replaced with (13), and let $p_x, p_u \in [0, 0.5)$. Then, for all $k = 0, \dots, N-1$, the control law $u_k = v_k + K_k y_k$, where $v_k \in \mathbb{R}^{n_u}$, $K_k \in \mathbb{R}^{n_u \times n_x}$, and $y_k \in \mathbb{R}^{n_x}$ from $y_{k+1} = A_k y_k + D_k w_k$, with $y_0 = x_0 - \mu_0$, results in a convex optimization problem in terms of the controller parameters v_k and K_k, for $k = 0, \dots, N-1$.

[1]Actually, CS theory does not require that the distribution be Gaussian, only knowledge of the required mean and covariance of the distribution is needed.

IV. CS-SMPC Controller Design

In Section III, we introduced the optimal CS controller. We are now ready to discuss the main result of this article that uses the CS convex problem formulation within an SMPC framework.

A. CS-SMPC Formulation

In this section, we utilize the solution of the optimal CS problem (12) to obtain a new SMPC problem formulation that includes terminal constraints on the state distribution to facilitate feasibility and stability. As shown in the Appendix, the resulting CS-SMPC problem can be efficiently solved using a convex programming solver.

Specifically, at time step k, we wish to solve the following finite-horizon stochastic optimal control problem:

$$\min_{\mathbf{u}_{k|k}^{N-1}} J_N(\mu_k, \Sigma_k; \mathbf{u}_{k|k}^{N-1}) = \mathbb{E}\left[\sum_{t=k}^{k+N-1} \ell_t(x_{t|k}, u_{t|k})\right] \tag{14a}$$

$$\text{s.t. } x_{t+1|k} = A_t x_{t|k} + B_t u_{t|k} + D_t w_t + r_t \tag{14b}$$

$$x_{k|k} \sim \mathcal{N}(\mu_k, \Sigma_k) \tag{14c}$$

$$\Pr\left(\alpha_{x,i}^\top x_{t|k} \le \beta_{x,i}\right) \ge 1 - p_{x,i}, \quad i = 0, \dots, N_s - 1 \tag{14d}$$

$$\Pr\left(\alpha_{u,j}^\top u_{t|k} \le \beta_{u,j}\right) \ge 1 - p_{u,j}, \quad j = 0, \dots, N_c - 1 \tag{14e}$$

$$\mathbb{E}\left[x_{k+N|k}\right] \in \mathcal{X}_f^\mu \tag{14f}$$

$$\mathbb{E}\left[(x_{k+N|k} - \mathbb{E}[x_{k+N|k}])(x_{k+N|k} - \mathbb{E}[x_{k+N|k}])^\top\right] \preceq \Sigma_f \tag{14g}$$

where $\mathcal{X}_f^\mu \subseteq \mathcal{X} \subset \mathbb{R}^{n_x}$ is compact and convex and $\Sigma_f \in \mathbb{R}^{n_x \times n_x}$.

Pictorially, problem (14) is illustrated in Fig. 1. At each time step k, the *predicted* system state and the *predicted* control have to satisfy the chance constraints for all initial state and noise realizations. In addition, at the end of the optimization horizon, the predicted state mean has to be within the set $\mathcal{X}_f^\mu$, denoted by the yellow polytope, and the system covariance has to be smaller than Σ_f, denoted by the yellow ellipse in Fig. 1.

Adding terminal constraints is a common methodology to guarantee recursive feasibility and stability for MPC problems [4], [9], [14]. However, as we are dealing in this article with a time-varying system subject to disturbances with unbounded support, computing terminal costs and constraints that ensure the recursive feasibility of the system are extremely challenging [33], [37]. Instead, in this article, we assume that recursive feasibility holds and will focus our analysis only on the stability question in Section IV-C. Specifically, we will assume that for problem (14), the conditions $\mathbb{E}[x_{k+N|k}] \in \mathcal{X}_f^\mu$ and $\mathbb{E}[(x_{k+N|k} - \mathbb{E}[x_{k+N|k}])(x_{k+N|k} - \mathbb{E}[x_{k+N|k}])^\top] \preceq \Sigma_f$ imply that $x_{k+N|k}$ also satisfies the chance constraint (14d) and, furthermore, if $\mathbf{u}_{k|k}^{N-1}$ is a feasible control for problem (14), then there exists u satisfying (14e), such that the trajectory generated by the control sequence $\{\mathbf{u}_{k|k}^{N-1}, u\}$

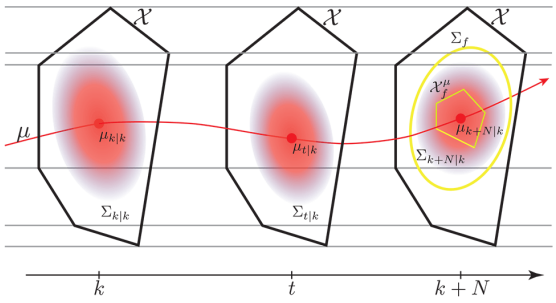


Fig. 1. Schematic describing the proposed CS-SMPC approach.

is such that $\mathbb{E}[x_{k+N+1|k}] \in \mathcal{X}_f^\mu$ and such that $\mathbb{E}[(x_{k+N+1|k} - \mathbb{E}[x_{k+N+1|k}])(x_{k+N+1|k} - \mathbb{E}[x_{k+N+1|k}])^\top] \preceq \Sigma_f$.

The next result provides a control *policy* that solves the optimization problem (14) in an efficient manner.

Theorem 2: Given μ_k, Σ_k, $\mathcal{X}_f^\mu$, and $\Sigma_f \succ 0$, let the control law

$$u_{t|k} = v_{t|k} + K_{t|k} y_{t|k} \tag{15a}$$

where for all $t = k, \ldots, k+N-1$

$$y_{t+1|k} = A_t y_{t|k} + D_t w_t \tag{15b}$$

and $y_{k|k} = x_k - \mu_k$. Then, problem (14) can be cast as a convex programming problem.

Proof: See Appendix A. ∎

Using the controller policy parameterization (15), at each step $k = 0, 1, \ldots$, one optimizes over the controller parameters $v_{t|k}$ and $K_{t|k}$ instead of the control actions $u_{t|k}$, where $t = k, \ldots, k+N-1$. Optimizing over control policies instead of control actions has the benefit of reducing conservatism by encoding the response to future disturbance realizations and that the satisfaction of the constraints is not only ensured for the measured initial state x_k but also for all initial states $x_k \sim \mathcal{N}(\mu_k, \Sigma_k)$. In that respect, the CS-SMPC controller design also guards—to a certain extent—against state measurement uncertainties. Since problem (14) is convex in terms of the controller variables $v_{t|k}$ and $K_{t|k}$, it can be efficiently solved using an SDP solver, such as Mosek [49].

Let the optimal cost of problem (14) at time step k be denoted by $J_N^*(\mu_k, \Sigma_k)$ and let the corresponding optimal control policy be denoted by $\mathbf{v}_{k|k}^{*N-1} = \{v_{k|k}^*, \ldots, v_{k+N-1|k}^*\}$ and $\mathbf{K}_{k|k}^{*N-1} = \{K_{k|k}^*, \ldots, K_{k+N-1|k}^*\}$. Then, the corresponding sequence of future control actions $\mathbf{u}_{k|k}^{*N-1} = \{u_{k|k}^*, \ldots, u_{k+N-1|k}^*\} = \{v_{k|k}^* + K_{k|k}^* y_{k|k}, \ldots, v_{k+N-1|k}^* + K_{k+N-1|k}^* y_{k+N-1|k}\}$ generates the predicted optimal state distribution sequence with mean and covariance pairs as $\{(\mu_{k|k}^*, \Sigma_{k|k}^*), (\mu_{k+1|k}^*, \Sigma_{k+1|k}^*), \ldots, (\mu_{k+N|k}^*, \Sigma_{k+N|k}^*)\}$, where, for all $t = k, \ldots, k+N-1$

$$\mu_{t+1|k}^* = A_t \mu_{t|k}^* + B_t v_{t|k}^* + r_t \tag{16a}$$

$$\begin{aligned}\Sigma_{t+1|k}^* = {} & A_t \Sigma_{t|k}^* A_t^\top + D_t D_t^\top + A_t \Sigma_{x y_{t|k}}^* K_{t|k}^{*\top} B_t^\top \\ & + B_t K_{t|k}^* \Sigma_{x y_{t|k}}^{*\top} A_t^\top + B_t K_{t|k}^* \Sigma_{y_{t|k}} K_{t|k}^{*\top} B_t^\top\end{aligned} \tag{16b}$$

and where

$$\Sigma_{x y_{t+1|k}}^* = A_t \Sigma_{x y_{t|k}}^* A_t^\top + B_t K_{t|k}^* \Sigma_{y_{t|k}} A_t^\top + D_t D_t^\top \tag{16c}$$

$$\Sigma_{y_{t+1|k}} = A_t \Sigma_{y_{t|k}} A_t^\top + D_t D_t^\top \tag{16d}$$

with $\mu_{k|k}^* = \mu_k$ and $\Sigma_{k|k}^* = \Sigma_{x y_{k|k}}^* = \Sigma_{y_{k|k}} = \Sigma_k$.

B. CS-SMPC Initialization

In problem (14), μ_k and Σ_k are additional optimization parameters that can be used to ensure the feasibility of the problem. If we assume we have perfect state information and ignore measurement noise, the value of x_k is available, and thus, we have the option to set $\mu_k = x_k$ and $\Sigma_k = 0$. However, since we have unbounded additive noise, the state may become unbounded as well and problem (14) can become infeasible if we always set $\mu_k = x_k$ and $\Sigma_k = 0$. In order to proceed, we, thus, impose the following assumption.

Assumption 2: At time step $k = 0$, problem (14) is feasible subject to the initial condition $\mathbb{E}[x_0] = \mu_0 = x_0$, $\Sigma_0 = 0$.

In order to keep problem (14) feasible for all future time steps $k \geq 1$, several approaches have been proposed [25], [30], [48]. In this work, we follow a dual-initialization strategy similar to [25], [36], and [48], which considers the initialization of Problem (14) as its own optimization problem with the choice of either measurement or prediction. Specifically, we set μ_k and Σ_k as either

$$\mu_k = x_k, \quad \Sigma_k = 0 \tag{17}$$

or

$$\mu_k = \mu_{k|k-1}^*, \quad \Sigma_k = \Sigma_{k|k-1}^* \tag{18}$$

according to

$$(\mu_k, \Sigma_k) = \begin{cases} (x_k, 0), & \text{if (14) is feasible and} \\ & J_N^*(x_k, 0) \leq J_N^*\left(\mu_{k|k-1}^*, \Sigma_{k|k-1}^*\right) \\ \left(\mu_{k|k-1}^*, \Sigma_{k|k-1}^*\right), & \text{otherwise} \end{cases} \tag{19}$$

where x_k is the current state measurement and $\mu_{k|k-1}^*$ and $\Sigma_{k|k-1}^*$ are the optimal *predicted* state mean and covariance from the previous time step as in (16).

Specifically, initialization (17) is used if the problem is feasible and results in a lower optimal cost than the problem with initialization as in (18). Otherwise, (18) is used. It follows that the optimal cost of problem (14) when the controller is implemented using (19) is always bounded by the optimal cost initialized using the predicted values of the mean and the covariance, that is, the inequality $J_N^*(\mu_k, \Sigma_k) \leq J_N^*(\mu_{k|k-1}^*, \Sigma_{k|k-1}^*)$ holds, where μ_k and Σ_k are chosen according to (19).

Notice that (17) is the result of closing the loop with the current measurement, while the choice (18) does not use the most recent measurement in problem (14), but instead uses it in the determination of the applied control action from (15a).

C. Stability Considerations

We now turn our attention to analyzing the stability properties of the closed-loop system with the proposed CS-SMPC controller. Since the stochastic system (1) is time-varying, analyzing the stability and recursive feasibility of the closed-loop system is extremely challenging, even in a deterministic setting. This is due to the difficulty of establishing a control law a priori which is both stabilizing and

feasible over all potential realizations [33], [37]. Alternatively, one can seek to derive weaker conditions that ensure some form of stability or convergence, by assuming the recursive feasibility of the problem as was proposed, for example [36] and [50]. We follow a similar approach and analyze the stability of the system assuming that the problem is recursively feasible.

In the rest of this article, we prove two important properties of the CS-SMPC algorithm. First, and similar to [33], [37], and [50], we show that, under suitable additional assumptions, the nominal system given by

$$\mu_{k+1} = A_k\mu_k + B_k v_k + r_k \tag{20}$$

where $\mu_k = \mathbb{E}[x_k]$ and $v_k = \mathbb{E}[u_k]$ is locally asymptotically stable, in the sense that $\lim_{t\to\infty}\mu_t = x_{g_t}$ for all μ_0 satisfying Assumption 2. In particular, we assume that $\{x_{g_t}\}_{t=0}^{\infty}$ are such that there exists a sequence $\{\eta_t\}_{t=0}^{\infty}$, where $\eta_t \in \mathcal{U} \subset \mathbb{R}^{n_u}$, such that

$$x_{g_{t+1}} = A_t x_{g_t} + B_t\eta_t + r_t \tag{21}$$

and $x_{g_t} \in \mathcal{X}$ holds for all $t \geq 0$.[2] Second, we show that the average stage cost for the stochastic system (1) is bounded from earlier, provided the problem remains feasible.

We introduce the following assumption.

Assumption 3: For all $k = 1, 2, \ldots$, there exists a vector $v \in \mathbb{R}^{n_x}$ and a matrix $K \in \mathbb{R}^{n_u\times n_x}$ such that $\mathbf{v}_{k|k}^{N-1} = \{v_{k|k-1}^*, \ldots, v_{k+N-2|k-1}^*, v\}$ and $\mathbf{K}_{k|k}^{N-1} = \{K_{k|k-1}^*, \ldots, K_{k+N-2|k-1}^*, K\}$ and such that the corresponding control sequence $\mathbf{u}_{k|k}^{N-1} = \{v_{k|k-1}^* + K_{k|k-1}^* y_{k|k}, \ldots, v + Ky_{k+N-1|k}\}$ is a feasible solution for problem (14) with the initialization strategy (18).

Assumptions 2 and 3 imply that, for all $k = 0, 1, \ldots$, problem (14) is feasible using the initialization strategy given by (19). Furthermore, by Assumption 3, at time $k+1$, for all $k = 0, 1, \ldots$, there exists a control policy

$$\begin{aligned}\mathbf{v}_{k+1|k+1}^{N-1} &= \{v_{k+1|k+1}, \ldots, v_{k+N|k+1}\}\\ &= \{v_{k+1|k}^*, \ldots, v_{k+N-1|k}^*, v_{k+N|k+1}\} \qquad (22a)\\ \mathbf{K}_{k+1|k+1}^{N-1} &= \{K_{k+1|k+1}, \ldots, K_{k+N|k+1}\}\\ &= \{K_{k+1|k}^*, \ldots, K_{k+N-1|k}^*, K_{k+N|k+1}\} \qquad (22b)\end{aligned}$$

with the corresponding sequence of predicted control actions

$$\begin{aligned}\mathbf{u}_{k+1|k+1}^{N-1} &= \{v_{k+1|k+1} + K_{k+1|k+1}y_{k+1|k+1}, \ldots\\ &\qquad v_{k+N|k+1} + K_{k+N|k+1}y_{k+N|k+1}\}\\ &= \{v_{k+1|k}^* + K_{k+1|k}^* y_{k+1|k+1}, \ldots\\ &\qquad v_{k+N-1|k}^* + K_{k+N-1|k}^* y_{k+N-1|k+1}\\ &\qquad v_{k+N|k+1} + K_{k+N|k+1}y_{k+N|k+1}\}\end{aligned} \tag{23}$$

that is a feasible solution of problem (14) with initial conditions (18). The control sequence (23) with initialization (18) generates the corresponding predicted state distribution sequence with mean and covariance pairs $(\mu_{k+1|k+1}, \Sigma_{k+1|k+1}), \ldots, (\mu_{k+N+1|k+1}, \Sigma_{k+N+1|k+1})\} = \{(\mu_{k+1|k}^*, \Sigma_{k+1|k}^*), \ldots, (\mu_{k+N|k}^*, \Sigma_{k+N|k}^*), (\mu_{k+N+1|k+1}, \Sigma_{k+N+1|k+1})\}$ as in (16), with $\mu_{k+1|k+1} = \mu_{k+1} = \mu_{k+1|k}^*$ and $\Sigma_{k+1|k+1} = \Sigma_{x_{y_{k+1|k+1}}} = \Sigma_{y_{k+1|k+1}} = \Sigma_{k+1} = \Sigma_{k+1|k}^*$.

Next, we show that $\mathbb{E}[\ell_t(x_{t|k}, u_{t|k})]$, where $\ell_t(\cdot,\cdot)$ has the special form (6), can be written as a deterministic expression in terms of the tuple $(\mu_{t|k}, \Sigma_{t|k}, v_{t|k}, K_{t|k})$. First, it follows from (6) that $\mathbb{E}[\ell_t(x_{t|k}, u_{t|k})] = \mathbb{E}[(x_{t|k} - x_{g_t})^\top Q_t(x_{t|k} - x_{g_t}) + u_{t|k}^\top R_t u_{t|k}] = \mathtt{tr}(Q_t(\mathbb{E}[(x_{t|k} - \mathbb{E}[x_{t|k}])(\star)^\top] + (\mathbb{E}[x_{t|k}] - x_{g_t})(\star)^\top)) + \mathtt{tr}(R_t(\mathbb{E}[(u_{t|k} - \mathbb{E}[u_{t|k}])(\star)^\top] + \mathbb{E}[u_{t|k}]\mathbb{E}[u_{t|k}]^\top))$. Let $\mathbb{E}[(x_{t|k} - \mathbb{E}[x_{t|k}])(\star)^\top] = \Sigma_{t|k}$, $\mathbb{E}[x_{t|k}] = \mu_{t|k}$, $\mathbb{E}[(u_{t|k} - \mathbb{E}[u_{t|k}])(u_{t|k} - \mathbb{E}[u_{t|k}])^\top] = K_{t|k}\Sigma_{y_{t|k}}K_{t|k}^\top$, and $\mathbb{E}[u_{t|k}] = v_{t|k}$. Therefore, the stage cost can be written as

$$\begin{aligned}\mathbb{E}[\ell_t(x_{t|k}, u_{t|k})] &= \tilde{\ell}_t(\mu_{t|k}, \Sigma_{t|k}, v_{t|k}, K_{t|k})\\ &= \mathtt{tr}(Q_t\Sigma_{t|k}) + \mathtt{tr}(R_t K_{t|k}\Sigma_{y_{t|k}}K_{t|k}^\top)\\ &\quad + (\mu_{t|k} - x_{g_t})^\top Q_t(\mu_{t|k} - x_{g_t}) + v_{t|k}^\top R_t v_{t|k}.\end{aligned} \tag{24}$$

Furthermore, the stage cost may be split into the mean and covariance components as $\tilde{\ell}_t(\mu_{t|k}, \Sigma_{t|k}, v_{t|k}, K_{t|k}) = \tilde{\ell}_t^m(\mu_{t|k}, v_{t|k}) + \tilde{\ell}_t^c(\Sigma_{t|k}, K_{t|k})$, where $\tilde{\ell}_t^m(\mu_{t|k}, v_{t|k}) = (\mu_{t|k} - x_{g_t})^\top Q_t(\mu_{t|k} - x_{g_t}) + v_{t|k}^\top R_t v_{t|k}$, and $\tilde{\ell}_t^c(\Sigma_{t|k}, K_{t|k}) = \mathtt{tr}(Q_t\Sigma_{t|k}) + \mathtt{tr}(R_t K_{t|k}\Sigma_{y_{t|k}}K_{t|k}^\top)$.

Finally, and with a slight abuse of notation, we may rewrite the cost function $J_N(\cdot)$ in (14a) as

$$\begin{aligned}J_N\left(\mu_k, \Sigma_k; \mathbf{u}_{k|k}^{N-1}\right) &= J_N\left(\mu_k, \Sigma_k; \mathbf{v}_{k|k}^{N-1}, \mathbf{K}_{k|k}^{N-1}\right)\\ &= J_N^m\left(\mu_k; \mathbf{v}_{k|k}^{N-1}\right) + J_N^c\left(\Sigma_k; \mathbf{K}_{k|k}^{N-1}\right)\end{aligned} \tag{25a}$$

$$J_N^m\left(\mu_k; \mathbf{v}_{k|k}^{N-1}\right) = \sum_{t=k}^{k+N-1}\tilde{\ell}_t^m(\mu_{t|k}, v_{t|k}) \tag{25b}$$

$$J_N^c\left(\Sigma_k; \mathbf{K}_{k|k}^{N-1}\right) = \sum_{t=k}^{k+N-1}\tilde{\ell}_t^c(\Sigma_{t|k}, K_{t|k}). \tag{25c}$$

Associated with the nominal system (20), we have the cost function $J_N(\mu_k, 0; \mathbf{v}_{k|k}^{N-1}, 0) = J_N^m(\mu_k; \mathbf{v}_{k|k}^{N-1})$, which is positive definite since Q_t and R_t are positive definite matrices for all $t = 0, 1, \ldots$. Let now $\breve{\mu}_k = \mu_k - x_{g_k}$. From (21), it follows that $\breve{\mu}_k$ satisfies the dynamics $\breve{\mu}_{k+1} = A_k\breve{\mu}_k + B_k(v_k - \eta_k)$.

The cost function may then be written in terms of $\breve{\mu}_k$ as

$$\breve{J}_N^m(\breve{\mu}_k; \mathbf{v}_{k|k}^{N-1}) = \sum_{t=k}^{k+N-1}\breve{\mu}_{t|k}^\top Q_t\breve{\mu}_{t|k} + v_{t|k}^\top R_t v_{t|k} \tag{26}$$

where $\breve{\mu}_{k|k} = \breve{\mu}_k$ and $\breve{\mu}_{t+1|k} = A_t\breve{\mu}_{t|k} + B_t(v_{t|k} - \eta_t)$ for all $k = 0, 1, \ldots$ and $t = k, k+1, \ldots, k+N-1$. When evaluated at $\breve{\mu}_k$ with the optimal control $\mathbf{v}_{k|k}^{*N-1}$, the cost (26) is given by $\breve{J}_N^m(\breve{\mu}_k; \mathbf{v}_{k|k}^{*N-1}) = \sum_{t=k}^{k+N-1}\breve{\mu}_{t|k}^{*\top}Q_t\breve{\mu}_{t|k}^* + v_{t|k}^{*\top}R_t v_{t|k}^*$, where $\breve{\mu}_{k|k}^* = \breve{\mu}_k$ and $\breve{\mu}_{t+1|k}^* = A_t\breve{\mu}_{t|k}^* + B_t(v_{t|k}^* - \eta_t)$.

Lemma 1: The function $V(k, \breve{\mu}_k) = \breve{J}_N^m(\breve{\mu}_k; \mathbf{v}_{k|k}^{*N-1})$ is positive definite and decrescent.

Proof: A straightforward calculation shows that $V(k, \mu)$ can be written as $V(k, \mu) = \mu^\top\Theta_k\mu + \mu^\top\theta_k + \rho_k$, where Θ_k is a positive definite matrix that depends on A_t and Q_t and θ_k and ρ_k are a vector and a scalar, respectively, that depend on Q_t, A_t, B_t, R_t, η_t, and $v_{t|k}^*$, for $t = k, \ldots, k+N-1$. From the problem definition, the first five of the previous expressions are

[2] In fact, for asymptotic tracking it only suffices to assume that there exist $T \geq 0$ such that (21) holds for all $t \geq T$. To simplify the analysis, we assume here $T = 0$.

bounded by a bound that is independent of k, whereas $v^*_{t|k}$ is bounded because $u^*_{t|k}$ satisfies the chance constraints (14e), and hence, from Lemma 3 in Appendix A, it follows that $\alpha^\top_{u,j} v^*_{t|k} \leq \beta_{u,j}$ for all $j = 0, \ldots, N_c - 1$ and all $t = k, \ldots, k+N-1$. Hence, $v^*_{t|k} \in \mathcal{U}$ which is compact, independent of k. The result follows immediately. See also [50, Lemma 1]. ■

Lemma 2: Let Assumption 3 hold and let $\Delta V(k, \check{\mu}_k) = V(k, \check{\mu}_k) - V(k+1, \check{\mu}_{k+1}) = \check{J}^m_N(\check{\mu}_k; \mathbf{v}^{*N-1}_{k|k}) - \check{J}^m_N(\check{\mu}_{k+1}; \mathbf{v}^{*N-1}_{k+1|k+1})$. Then, if

$$\check{\mu}^{*\top}_{k+N|k} Q_{k+N} \check{\mu}^*_{k+N|k} + v^\top_{k+N|k+1} R_{k+N} v_{k+N|k+1} < \check{\mu}^{*\top}_{k|k} Q_k \check{\mu}^*_{k|k} + v^{*\top}_{k|k} R_k v^*_{k|k} \quad (27)$$

where $v_{k+N|k+1}$ is an admissible control as in Assumption 3, $\Delta V(k, \check{\mu}_k)$ is positive definite for all $\mu_k = \check{\mu}_k + x_{g_k}$ admitting a feasible solution for problem (14).

Proof: First, note that $\check{\mu}_{k+1} = \check{\mu}^*_{k+1|k}$. The cost difference $\check{J}^m_N(\check{\mu}_k; \mathbf{v}^{*N-1}_{k|k}) - \check{J}^m_N(\check{\mu}^*_{k+1|k}; \mathbf{v}^{N-1}_{k+1|k+1})$, where $\mathbf{v}^{N-1}_{k+1|k+1}$ is the control sequence given in (22a), can be equivalently written as $\check{J}^m_N(\check{\mu}_k; \mathbf{v}^{*N-1}_{k|k}) - \big(\check{J}^m_N(\check{\mu}_k; \mathbf{v}^{*N-1}_{k|k}) - \check{\mu}^{*\top}_{k|k} Q_k \check{\mu}^*_{k|k} - v^{*\top}_{k|k} R_k v^*_{k|k} + \check{\mu}^{*\top}_{k+N|k} Q_{k+N} \check{\mu}^*_{k+N|k} + v^\top_{k+N|k+1} R_{k+N} v_{k+N|k+1}\big)$. Therefore, from (27), it follows that $\check{J}^m_N(\check{\mu}_k, \mathbf{v}^{*N-1}_{k|k}) - \check{J}^m_N(\check{\mu}^*_{k+1|k}; \mathbf{v}^{N-1}_{k+1|k+1}) > 0$. Assumption 3 along with (19) imply that $\check{J}^m_N(\check{\mu}_{k+1}, \mathbf{v}^{*N-1}_{k+1|k+1}) \leq \check{J}_N(\check{\mu}^*_{k+1|k}; \mathbf{v}^{N-1}_{k+1|k+1})$. Hence, $\check{J}^m_N(\check{\mu}_k, \mathbf{v}^{*N-1}_{k|k}) - \check{J}^m_N(\check{\mu}_{k+1}, \mathbf{v}^{*N-1}_{k+1|k+1}) > 0$. ■

We are now ready to state the stability result for the nominal system.

Theorem 3: Consider system (1) where the mean dynamics $\mathbb{E}[x_k] = \mu_k$ are given by (20), $\mu_0 = x_0$, $\Sigma_0 = 0$, and $\mathbb{E}[u_k] = v_k = v^*_{k|k}$ is given by the optimal solution to problem (14). Given Assumption 3, if inequality (27) is satisfied for all $k = 0, 1, \ldots$, the closed-loop nominal system converges asymptotically to the reference trajectory x_{g_k}, that is, $\check{\mu}_k = \check{\mu}^*_{k|k} \to 0$ as $k \to \infty$.

Proof: From Lemma 1, $V(k, \check{\mu}_k) = \check{J}^m_N(\check{\mu}_k; \mathbf{v}^{*N-1}_{k|k})$ is positive definite and decrescent. Moreover, inequality (27) implies that $\Delta V(k, \check{\mu}_k) = V(k, \check{\mu}_k) - V(k+1, \check{\mu}_{k+1})$ is positive definite along the trajectories of the closed-loop nominal system, for all $\check{\mu}_k$ such that μ_k is feasible. Therefore, using [50, Th. 1], it follows that the nominal closed-loop system is locally asymptotically stable and its trajectories converge asymptotically to zero, that is, $\lim_{k\to\infty} \check{\mu}_k = 0$. ■

Remark 2: Inequality (27) may be incorporated into the optimal control problem as a constraint to ensure that the asymptotic stability condition is satisfied [50]. However, this may negatively impact the feasibility of problem (14).

Next, we investigate the stability properties of the system (1). Since we are dealing with additive unbounded uncertainty, it is difficult to design a control law that ensures the mean square stability of the state [15]. Instead, and similar to [22], we show that the expected average stage cost is bounded from above. Specifically, we have the following result.

Theorem 4: Suppose Assumption 3 holds. Then, there exists $\ell_{\max} > 0$ such that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} \mathbb{E}\big[\ell_{k+t}(x^*_{k+t|k}, u^*_{k+t|k})\big] \leq \ell_{\max}. \quad (28)$$

Proof: See Appendix B. ■

Fig. 2. AutoRally chassis and compute box [52].

V. CS-SMPC Autonomous Driving Implementation

In this section, we utilize the previous CS-SMPC algorithm from Theorem 2 for the autonomous racing of a scaled autonomous ground vehicle, as shown in Fig. 2.

A. AutoRally Experimental Platform

The AutoRally platform, as shown in Fig. 2, is a 1/5-scale autonomous vehicle platform developed at Georgia Institute of Technology, Atlanta, GA, USA, for performing aggressive maneuvers on dirt tracks at the limits of the vehicle's handling capabilities [51]. The vehicle is rear-wheel drive and powered by a brushless motor and is capable of achieving a top speed of 25 m/s. The motor electronic speed controller (ESC) also features an electronic braking feature, which is currently used as the only means of braking. The vehicle uses Ackermann steering driven by an RC servo motor. The actuators have a maximum update rate of 50 Hz. The AutoRally platform includes its own onboard computation utilizing an Intel i7 CPU and NVIDIA GTX GPU. The vehicle performs state estimation onboard at 200 Hz utilizing real-time kinematic global positioning system (RTK GPS), inertial measurement unit (IMU), and wheel encoder measurements. Additionally, the vehicle features stereo cameras which can be used to augment the state estimate by providing a prediction of the vehicle's position relative to a known track template.

The AutoRally is tested on a dirt track constructed using a 0.15-m-diameter corrugated drainage pipe to form the track boundaries. The track formed by these boundaries is 180-m long and 4-m wide. The track includes nine corners ranging in radii from 2 to 7 m and its longest straight is 32-m long. A racing line is generated for the track using the iterative quadratic programming method described in [53]. This method is initialized with a nominal path calculated as the track centerline between the inner and outer boundaries. The algorithm then seeks to minimize the curvature of the path while obeying boundary constraints, that is, the path must not pass within a given threshold of the track boundaries.

The task of the CS-SMPC controller is to use the vehicle's current state estimate, along with the initialization strategy given by (17) and (18), to determine the optimal motor duty cycle and steering angle $u_{k|k}$ at each sampling time to drive the vehicle so that it follows the racing line around the track while providing a specified level of safety with respect to the uncertainty of the planned trajectory by remaining within the track boundaries.

B. Vehicle Dynamics Model

A nonlinear nominal vehicle model was developed and its parameters were calculated using parameter identification. The system was then linearized about the current and future predicted operating trajectories to obtain a system of the form (1).

1) Nonlinear Dynamic Vehicle Model: The AutoRally autonomous vehicle is modeled using a rear-wheel drive, single-track bicycle model. The vehicle kinematics are represented in a map-based curvilinear coordinate system (as opposed to a Cartesian coordinate system), which provides an intuitive understanding of the vehicle's position and heading relative to the track. The state and control variables are given as

$$x = \left[v_x, v_y, \dot{\psi}, \omega_F, \omega_R, e_\psi, e_y, s\right]^\top, \quad u = \left[\delta, T\right]^\top \tag{29}$$

where v_x, v_y, and $\dot{\psi}$ are the vehicle's longitudinal velocity, lateral velocity, and yaw rate, respectively, and the front and rear wheel rotation rates are ω_F and ω_R, respectively. The direction error and the lateral distance error from the racing line are denoted with e_ψ and e_y. The variable s is the curvilinear position along the racing line in meters. The control command consists of the steering angle δ and the rear-wheel drive/brake command T.

The tire friction model uses a combination of the friction ellipse model described in [54] and a nonparametric model inspired by [55] to approximate the residual error. The longitudinal and lateral friction forces acting on the tires are given by

$$f_{ix}(\delta, v_x, v_y, \dot{\psi}, \omega_F, \omega_R) = f_{iz}\mu_{ix} \tag{30a}$$
$$f_{iy}(\delta, v_x, v_y, \dot{\psi}, \omega_F, \omega_R) = f_{iz}\mu_{iy} + \Phi_i(\delta, \alpha_i) \tag{30b}$$

where f_{iz} represents the normal force acting on the respective tire, μ_{ix} and μ_{iy} are the dynamic coefficients of friction, which depend on the wheel slips and tire parameters as given in [54], and $i = F, R$ represents the front or back wheel, respectively. The function $\Phi_i(\delta, \alpha_i)$, where $\alpha_i = \arctan(v_{iy}/v_{ix})$ is the tire slip angle, augments the friction ellipse model to capture error seen in the experimental data and is implemented as a neural network trained using data collected from the vehicle.

The rear-wheel drive/brake control is a nonlinear mapping to a rear-wheel speed setting. We model the transient response as a first-order, nonlinear system

$$\dot{\omega}_R = \Theta(\omega_R, T) \tag{31}$$

where $\Theta(\cdot, \cdot)$ is computed by a neural network trained on experimental data. Thus, the system dynamics can be summarized as

$$\dot{v}_x = \frac{f_{Fx}\cos\delta - f_{Fy}\sin\delta + f_{Rx}}{m} + v_y\dot{\psi} \tag{32a}$$
$$\dot{v}_y = \frac{f_{Fx}\sin\delta + f_{Fy}\cos\delta + f_{Ry}}{m} - v_x\dot{\psi} \tag{32b}$$
$$\ddot{\psi} = \frac{\left(f_{Fy}\cos\delta + f_{Fx}\sin\delta\right)\ell_F - f_{Ry}\ell_R}{I_z} \tag{32c}$$
$$\dot{\omega}_F = -\frac{r_F}{I_{\omega F}} f_{Fx} \tag{32d}$$
$$\dot{\omega}_R = \Theta(\omega_R, T) \tag{32e}$$
$$\dot{e}_\psi = \dot{\psi} - \frac{v_x\cos e_\psi - v_y\sin e_\psi}{1 - \rho(s)e_y}\rho(s) \tag{32f}$$
$$\dot{e}_y = v_x\sin e_\psi + v_y\cos e_\psi \tag{32g}$$
$$\dot{s} = \frac{v_x\cos e_\psi - v_y\sin e_\psi}{1 - \rho(s)e_y} \tag{32h}$$

TABLE I

DYNAMIC BICYCLE MODEL AND FRICTION ELLIPSE PARAMETERS FOR THE AUTORALLY PLATFORM AND HYPERPARAMETERS USED FOR AUTONOMOUS DRIVING IMPLEMENTATIONS

Parameter	Value
m (kg)	22
I_z (kg·m^2)	1.1
ℓ_F (m)	0.34
ℓ_R (m)	0.23
$I_{\omega F}$ (kg·m^2)	0.10
r_F (m)	0.095
h (m)	0.12
tire_B	4.1
tire_C	0.95
tire_D	1.1

Parameter	Value
N	20
Δt (s)	0.05
Update Rate (Hz)	40
Q_{v_x}	4
Q_{v_y}	1
Q_{e_ψ}	20
Q_{e_y}	20
R_δ	20
R_T	2

where m is the vehicle mass, I_z is the vehicle moment of inertia about the vertical axis, r_F is the radius of the front wheel, $I_{\omega F}$ is the moment of inertia of the front wheel about the axle, and $\rho(s)$ is the curvature of the racing line at position s along the racing line.

The nonlinear system given by (32) can be succinctly written as $\dot{x}(t) = f(x(t), u(t))$, where $x(t) \in \mathbb{R}^8$ is the state vector and $u(t) \in \mathbb{R}^2$ is the control action. The model is discretized using Euler integration such that

$$x_{k+1} = F(x_k, u_k) = x_k + f(x_k, u_k)\Delta t \tag{33}$$

where $x_k = x(t_k)$, $u_k = u(t_k)$, and $\Delta t = t_{k+1} - t_k$. The discretized system (33) is used for model identification and control design. The model includes five unknown vehicle parameters, three unknown tire parameters, and the unknown functions $\Theta(\omega_R, T)$ and $\Phi_i(\delta, \alpha_i)$, where $i = F, R$. An unscented Kalman filter (UKF) was employed to estimate the vehicle and tire parameters following the method outlined in [56], while neural networks were trained to estimate the unknown nonlinear functions. The identified nominal vehicle and tire parameters are given in Table I, where tire_B, tire_C, and tire_D are parameters in Pacejka's magic formula used in the friction ellipse model given in [54].

2) Affine Time-Varying System: Starting with the current state x_k, let the predicted nominal trajectory be $(\hat{x}_{k|k}, \ldots, \hat{x}_{k+N|k})$, where $\hat{x}_{k|k} = x_k$ and the corresponding control sequence be $(\hat{u}_{k|k}, \ldots, \hat{u}_{k+N-1|k})$, such that $\hat{x}_{t+1|k} = F(\hat{x}_{t|k}, \hat{u}_{t|k})$, for $t = k, \ldots, k+N-1$. Then, we can rewrite system (33) as

$$x_{k+1} = A_k x_k + B_k u_k + r_k \tag{34}$$

where A_k and B_k are given by

$$A_k = \left.\frac{\partial F}{\partial x}\right|_{\hat{x}_k, \hat{u}_k}, \quad B_k = \left.\frac{\partial F}{\partial u}\right|_{\hat{x}_k, \hat{u}_k} \tag{35}$$

and

$$r_k = F(\hat{x}_{k|k}, \hat{u}_{k|k}) - A_k\hat{x}_{k|k} - B_k\hat{u}_{k|k}. \tag{36}$$

We, thus, obtain an approximation of (33) about the current and expected future operating points using an affine time-varying system of the form (34).

3) Disturbance Model: To complete the model we introduce a stochastic term to (34) that accounts for model inaccuracies, linearization error, and external disturbances. To this end, we introduce an additive Gaussian disturbance w_k with zero mean and unit covariance. We can then write the modified system (34) as in (1).

Remark 3: It should be noted that contrary to (1), where the matrices are assumed to be known and given for all $k = 0, 1, \ldots$, and since we are using a receding horizon implementation, in our case, the system matrices are only known over the given horizon as future matrices are determined by (35) and (36), which depend on the future nominal trajectory, which has not been computed yet. Nonetheless, this is standard practice in the MPC controller implementation that often works well in most applications. Our experimental results in Section VI-B show that this is indeed the case for our autonomous racing problem.

Since (34) provides a nominal model for the system and D_k determines the covariance of the additive disturbance [since $w_k \sim \mathcal{N}(0, I)$], we estimate D_k online using the physical system's observed deviation from the nominal model as follows. From the value of x_k at step $k \geq 0$, we can estimate the previous disturbances using (34) and (1) from $z_{t+1|t} = D_t w_t = x_{t+1} - (A_t x_t + B_t u_t + r_t)$, for $t = 0, \ldots, k-1$, where k is the current iteration index. We collect all the measurements of $z_{t+1|t}$ as $S_{y|k} = \{z_{k-N_y+1|k-N_y}, z_{k-N_y+2|k-N_y+1}, \ldots, z_{k|k-1}\}$, where $N_y \leq k$ is the number of previous iterations over which the sample is stored. Choosing $N_y = k$ amounts to keeping all previous disturbance realizations from each sampling time, whereas a choice of $N_y < k$, leads to a sliding window in which the current disturbance D_k is estimated only from the N_y most recent measurements. Using the fact that $\mathbb{E}[w_t w_t^\top] = I$ and $\mathbb{E}[w_t] = 0$, we can write the covariance of $z_{t+1|t}$ as $\mathbb{E}[z_{t+1|t} z_{t+1|t}^\top] - \mathbb{E}[z_{t+1|t}]\mathbb{E}[z_{t+1|t}]^\top = \mathbb{E}[D_t w_t w_t^\top D_t^\top] = D_t D_t^\top$.

At each sampling time k, we update the estimate of the current D_k using the sample covariance of the latest data, and we predict forward using a time-invariant approximation over the prediction horizon given by $D_t \approx \mathrm{Cov}(S_{y|k})^{1/2}$, for $t = k, \ldots, k+N-1$, where $\mathrm{Cov}(S_{y|k})$ is the sample covariance of $S_{y|k}$ and the square root is computed using the Cholesky decomposition. This means that the disturbance matrix D_k is time-varying between sampling times, but constant over any given prediction horizon. This is because future disturbances are estimated purely from previously observed disturbances and are considered to be independent of the state and the control.

C. Cost Function and Constraints

Since the state (29) expresses the vehicle's position in terms of the racing line, the controller objective is to drive e_ψ and e_y to zero while trying to maintain v_x to a given target velocity. An additional cost can be placed on v_y to minimize side-slip; however, this would restrict aggressive cornering, as the latter tends to induce large side-slip angles at high speeds.

The cost function is, therefore, chosen such that $x_{g_k} = [v_{g_k}, 0, 0, 0, 0, 0, 0, 0]^\top$, $Q_k = \mathtt{diag}(Q_{v_x}, Q_{v_y}, 0, 0, 0, Q_{e_\psi}, Q_{e_y}, 0)$, $R_k = \mathtt{diag}(R_\delta, R_T)$, where Q_{v_x} is the velocity cost, Q_{v_y} is the side-slip cost, Q_{e_ψ} is the heading error cost, Q_{e_y} is the off-path error cost, R_δ is the steering cost, and R_T is the throttle cost. The chosen values for these parameters are shown in Table I.

The target velocity is varied according to the curvature of the racing line using the function

$$v_{g_k} = v_{g,\max} - |\rho(s_k)| v_{g,\max} \tag{37}$$

where $\rho(s_k)$ is the curvature of the nearest point on the racing line to the current state. For our track, the curvature is bounded by 1, and (37) implies that the target velocity is varied in the range $(0, v_{g,\max})$, slowing down according to the severity of a curve.

The parameters for the chance constraints are chosen to provide the desired probability of the vehicle trajectory remaining within the track boundaries. The track boundaries are represented using two chance constraints (one for the inner boundary and one for the outer boundary), given by $\Pr(\alpha_{e_y}^\top x_k < \beta_{e_y}) \geq 1 - p_{x,1}$ and $\Pr(\alpha_{e_y}^\top x_k > -\beta_{e_y}) \geq 1 - p_{x,2}$, with $\alpha_{x,1} = \alpha_{x,2} = \alpha_{e_y}$, $\beta_{x,1} = \beta_{x,2} = \beta_{e_y}$, $e_{y_k} = \alpha_{e_y} x_k$, $\beta_{e_y} = 0.5 \times$ (track width), and $p_{x,1} = p_{x,2} = 0.1$ design parameters chosen to provide a suitable level of safety.

The control constraints are designed similarly to keep the actuator commands within the limits imposed by the hardware.

D. Feedback Matrix Offline Computation

To run the original CS-SMPC algorithm on the AutoRally platform, one needs to solve the convex problem (A.5) online at each time step. The current AutoRally hardware does not allow solving this convex problem at a sampling frequency that would be sufficient for driving fast around the track. Due to the periodic nature of the autonomous racing task, however, it is possible to precompute the feedback matrices in a simulation environment described in Section V-E and store them in a lookup table, namely, as $S_K = \{\bar{K}_0, \bar{x}_0; \bar{K}_1, \bar{x}_1; \ldots; \bar{K}_{N_K}, \bar{x}_{N_K}\}$, where $\bar{K}_i = K^*$ is the optimal K from the solution of problem (A.5) for the initial state $\bar{x}_i = x_{i|i}$ and $i = 0, 1, \ldots, N_K$. During runtime, the optimal feedback matrix is chosen as $K_k = \bar{K}_j$ such that $j = \mathtt{argmin}_i(\|x_k - \bar{x}_i\|)$, where x_k is the current realized state and K_k is the feedback matrix applied during the current optimization step. The on-board calculation is then restricted to the calculation only of the feedforward control action V^*. A C++ implementation of this scheme is able to run on the AutoRally at 40 Hz.

E. Simulation Environment

To support the development and validation of the proposed CS-SMPC controller, a custom high-fidelity simulation environment was created, as shown in Fig. 3, which acts as a seamless replacement for the physical hardware. We used the commercial vehicle simulation software CarSim [57] as the simulation back end, integrated our own hybrid friction model described in Section V-B1, and added an ROS front end for communication with the controller. Thanks to the ROS front end, the simulator communication with the controller is identical to that of the physical AutoRally platform, allowing the same controller implementation to be tested

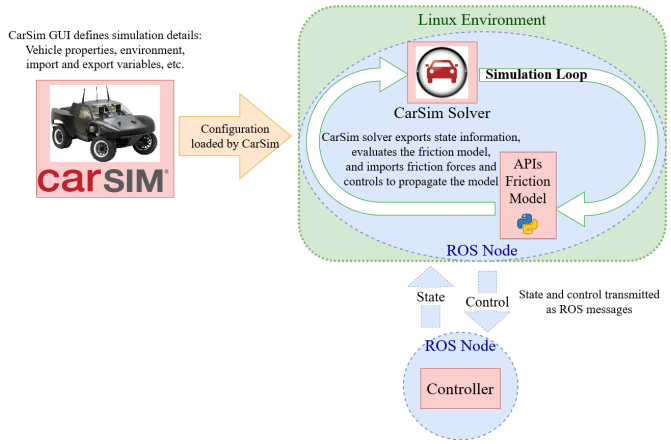
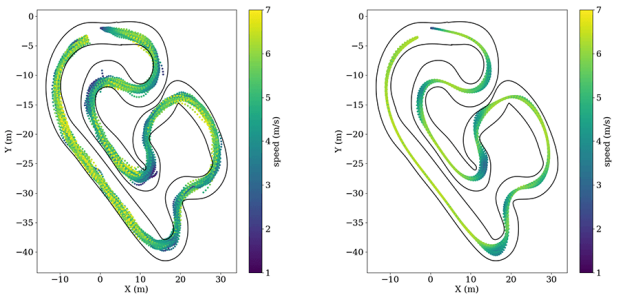


Fig. 3. AutoRally simulation environment.



(a) (b)

Fig. 4. Visualization of all simulated trajectories. (a) MPPI. (b) CS-SMPC.

TABLE II
COMPARISON OF CONTROLLER SAFETY OVER 100 MONTE CARLO SIMULATIONS

Metric	MPPI	CS-SMPC
Crashes (/100)	11	3
Ave Speed (m/s)	5.10	5.54
Ave Lap Time (s)	28.64	27.82

interchangeably with the actual vehicle or the simulation environment.

The controller communicates with the simulation environment (equivalently, the physical AutoRally vehicle) using a custom chassisCommand ROS message, which includes values for the steering command (δ) and throttle command (T). The simulation executes these commands by calculating the predicted friction forces using the friction and throttle models described in (30) and (31) and the wheel friction forces are sent to CarSim, which then propagates a high fidelity vehicle model using AutoRally's vehicle parameters. These vehicle parameters include those of the dynamic bicycle model identified in Section V-B1 along with additional parameters necessary for a double-track full-vehicle model which can be measured directly or are given in the manufacturer specifications of AutoRally's components. The updated state is published as a standard ROS Odometry message which is consumed by the controller to update the current state estimate for the next planning iteration.

VI. RESULTS

To demonstrate the performance of the proposed CS-SMPC algorithm, we compared it with the model predictive path integral (MPPI) algorithm [2], a state-of-the-art sampling-based MPC controller. The main advantage of CS-SMPC over MPPI is the explicit encoding of the chance constraints, which provide a quantifiable measure of the safety of the planned trajectory. MPPI only implicitly accounts for constraints by incorporating them in the cost function. On the other hand, MPPI is more versatile, as it can handle general nonlinear dynamics and costs. However, since it is based on trajectory rollouts, MPPI's performance is heavily dependent on the number of sampled trajectories.

In prior work, a simple nonparametric vehicle model of the AutoRally was used with MPPI to allow for fast trajectory predictions utilizing the power of GPU hardware for efficient parallelization [2]. In this work, we compare both methods using the high-fidelity dynamic vehicle model presented in Section V-E. Working with this model is computationally more expensive but provides better prediction accuracy. We tuned MPPI to sample as many trajectories as possible while having the same update rate as CS-SMPC. Therefore, both controllers use the same model and parameters, so their differences lie entirely in the representation and solution methods.

A. Monte Carlo Simulations

Because the CS-SMPC controller explicitly considers disturbances when designing the optimal control, it is able to provide robust constraint satisfaction. On the other hand, a deterministic formulation only satisfies constraints for the nominal, undisturbed trajectory. Therefore, a key advantage of the CS-SMPC controller is a systematic way to tradeoff performance for safety.

The robustness of the CS-SMPC controller was tested via 100 simulated runs of length one lap each in the simulation environment shown in Fig. 3, with each controller at a target speed of 6 m/s. In each run, the simulated friction parameters (tire_B, tire_C, and tire_D) in Table I, as well as the steering and throttle inputs δ, T were randomly varied as $\Delta_{\theta_i} \sim N(0, 0.1)$, where $\theta_i = \hat{\theta}_i(1 + \Delta_{\theta_i})$, where θ_i is a given vehicle model parameter or input, and $\hat{\theta}_i$ is the nominal value used by the controller. In other words, both controllers used the nominal parameters to design the control policy, but the simulation was carried out with random variations of these parameters.

As shown in Table II, CS-SMPC has the lowest crash instance count, while also driving faster than MPPI. Moreover, Fig. 4 shows that CS-SMPC has the smallest spread in the realized trajectories and consistently maintains the same speed through the same sections of the track between different runs with different vehicle parameters, demonstrating greater robustness to the disturbances.

B. Experimental Evaluation

1) Autonomous Racing With the AutoRally: We next compared the performance of CS-SMPC with MPPI on the physical AutoRally platform at the Georgia Tech Autonomous Racing Facility (GTARF), Atlanta, dirt race track, as shown in Fig. 5. Although the vehicle model was identified using data collected from the dirt track, it is difficult to model the vehicle

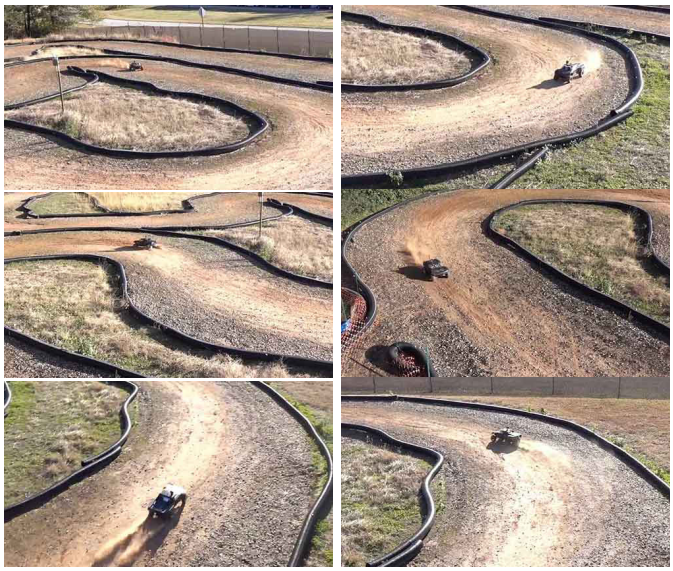
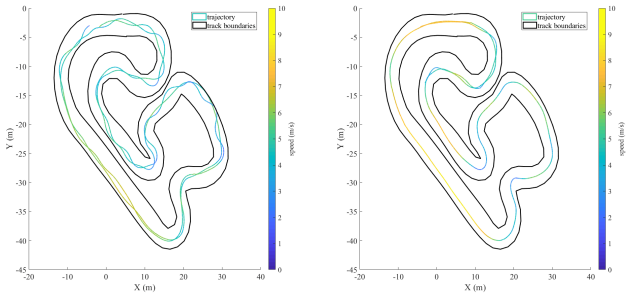
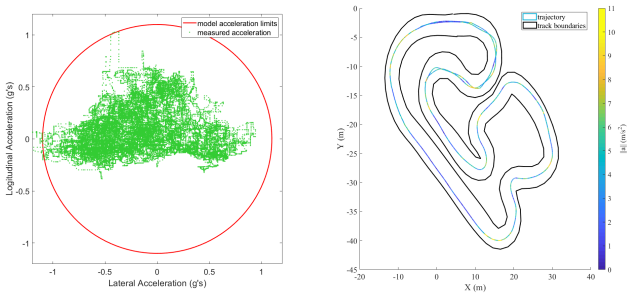


Fig. 5. CS-SMPC AutoRally experimental evaluation.



(a) (b)

Fig. 6. Experimental trajectories on the AutoRally dirt track. (a) MPPI. (b) CS-SMPC.



(a) (b)

Fig. 7. Measured acceleration magnitudes from experimental results with CS-SMPC on the AutoRally dirt track. (a) G–G circle. (b) Combined acceleration.

sliding, slipping, and so on, at high speeds while performing aggressive maneuvers at the limits of the vehicle's handling capabilities. Additionally, the track contains rocks and ruts that cannot be modeled and affect the vehicle's performance at high speeds. Therefore, this task necessitates a controller that can account for uncertainty in the vehicle dynamics and responds to disturbances, making this platform an ideal test bed for SMPC approaches.

With the parameters in Table I fixed, and for both controllers, we initially tested with a target speed of 3 m/s and increased the maximum target speed until the car could no longer reliably complete a lap around the track. The results from the highest speeds at which each controller could reliably complete the entire track (7 m/s for MPPI and 9 m/s for CS-SMPC) are presented here. It should be noted that as the target speed is increased, both controllers increasingly deviate from the racing line, as it becomes more challenging to manage the vehicle slip while driving at high speeds and under large lateral and longitudinal accelerations. Even at a 2-m/s-higher target speed than MPPI, CS-SMPC manages to do a better job at minimizing the off-path and heading errors (e_y and e_ψ, respectively). At target speeds higher than 7 m/s, the MPPI controller becomes unstable and is unable to reliably complete a lap without colliding with the track boundaries.

As shown in Table III, CS-SMPC is able to safely achieve a higher maximum speed than MPPI and also complete the laps with a faster average speed and lower lap time. Photographs of the test with CS-SMPC are shown in Fig. 5 and a video of the test can be viewed at https://youtu.be/1ms5fctfAYs.

Additionally, as seen in Fig. 6, CS-SMPC maintains a smoother and more consistent trajectory than MPPI while simultaneously driving at higher speeds. This demonstrates that CS-SMPC is more robust to disturbances and safely captures the dispersion of trajectories to remain close to the racing line. Finally, Fig. 7 shows the acceleration profiles of the AutoRally running CS-SMPC, as measured by the onboard IMU. The theoretical maximum acceleration for the given tire

TABLE III

EXPERIMENTAL COMPARISON OF CONTROLLER PERFORMANCE

Metric	MPPI	CS-SMPC
Max Speed (m/s)	7.0	8.7
Ave Speed (m/s)	4.6	5.2
Min Lap Time (s)	40.1	35.4

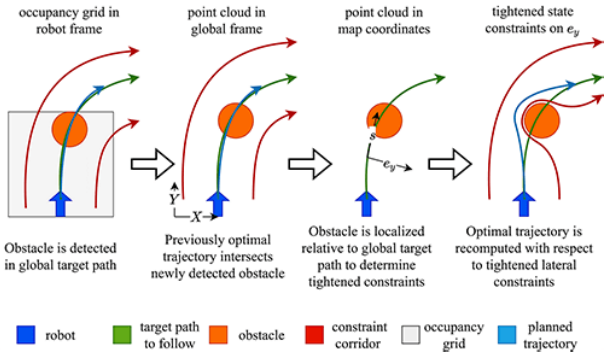


Fig. 8. Obstacle avoidance pipeline.

model is 10.8 m/s^2, indicating that the vehicle indeed reaches the limits of friction in many sections of the track.

The superior performance of CS-SMPC over MPPI can be attributed to the disturbance representation and the different solution methods used in these two approaches. CS-SMPC explicitly models the uncertainty in the vehicle dynamics and solves a convex program. MPPI, on the other hand, does not model the system uncertainty, but rather it injects a disturbance into the control input to search for the optimal control using a sampling-based method.

2) Obstacle Avoidance With a Full-Scale Vehicle: Having demonstrated the performance of CS-SMPC while performing

Fig. 9. Surrogate 1 is an autonomous Ford Escape.

aggressive maneuvers in the dynamic friction regime in an autonomous racing scenario, in this section, we demonstrate the use of the CS-SMPC algorithm in a more conservative, but nonetheless still challenging, autonomous driving task.

While the small size of the AutoRally platform allows it to perform aggressive maneuvers that would be difficult to reproduce on a real vehicle due to safety concerns, commercial applications of autonomous vehicles pose their own unique challenges. For example, RTK GPS provides accurate position measurements that overcome the drift encountered from other measurements. However, a reliable GPS signal is not available in environments, where the horizon is heavily occluded. In such GPS-denied cases, other localization methods are necessary and the vehicle's controller must be able to respond to inaccuracies in the state estimate. Additionally, while on a race track the boundaries are known ahead of time and no obstacles may be expected when merely performing time trials, in commercial applications dynamic obstacles may appear suddenly in the vehicle's planned path, requiring the controller to react quickly to swerve and avoid these obstacles and then reconnect with the reference path, as shown in Fig. 8. Our full-scale vehicle implementation of CS-SMPC addresses these two challenges, namely, unreliable state estimation and dynamic obstacle avoidance.

We implemented CS-SMPC on a full-scale autonomous Ford Escape at the Autonomous Solutions Inc., (ASI) facilities in Petersboro, UT, USA, shown in Fig. 9. The vehicle was tested in a GPS-denied environment, relying solely on radar and inertial measurements for localization. We used CS-SMPC to navigate the vehicle around the test track by following a static target path and avoiding dynamic obstacles which were detected online by LIDAR and may appear in the vehicle's static goal path. That is, given a global plan that navigates through a known environment—but which does not account for dynamic obstacles which may appear in the vehicle's path—the controller must reroute the plan online around obstacles that appear within the vehicle's field of view while minimizing deviation from the global plan.

To perform obstacle avoidance no additional planner was employed beyond the global planner which outputs a static path around the test track, irrespective of any obstacles on the track. The task of responding to an obstacle is left to the CS-SMPC algorithm, which encodes the global path through the cost function as described in Section V-C and accounts for the obstacles using chance constraints. The advantages of our approach over a more traditional replanning and tracking paradigm are twofold. First, we avoid reliance on an online planner that may not be computationally feasible at high update rates which are needed for avoiding dynamic obstacles

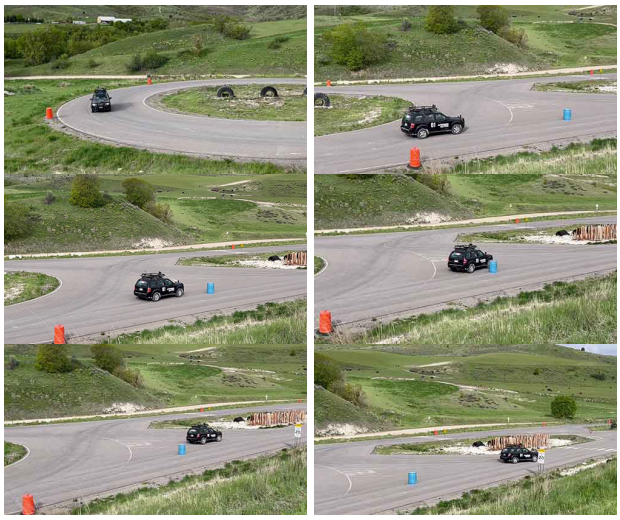


Fig. 10. GPS-denied navigation and obstacle avoidance using CS-SMPC.

appearing suddenly in the vehicle's path. Second, the decision of acceptable proximity to an obstacle is determined through the chance constraints as opposed to through a heuristic (e.g., a fixed distance) and without the need for a stochastic online path planner (which may be computationally demanding). Since the chance constraints must define a convex safe region, an obstacle-free corridor was constructed online using the LIDAR measurements, the edges of which define the chance constraint boundaries with respect to the lateral distance from the global path, as shown in Fig. 8. Although the region is nonconvex in Cartesian space, it is convex in terms of the lateral distance from the global path (i.e., e_y).

Snapshots of the field experiment at the ASI facilities are shown in Fig. 10. A video of the test may be viewed at https://youtu.be/248S_eNz_Tw. The results in Fig. 10 and the corresponding video show that CS-SMPC is able to successfully respond to an obstacle (barrel) placed in the vehicle's path and find an obstacle-free path around it while providing a buffer corresponding to a prescribed degree of safety as determined by the provided chance constraints. The obstacle avoidance is performed in an optimal fashion that minimizes deviation from the global target path and is able to successfully reconnect with the reference path to continue around the test track once the obstacle has been cleared.

VII. Conclusion

This article proposes an SMPC approach for the control of uncertain linear systems subject to Gaussian noise and chance constraints using CS ideas. The proposed SC-SMPC algorithm demonstrates several safety advantages over existing SMPC methods by explicitly shaping the terminal state distribution of future trajectory realizations. The practical usefulness of the approach is demonstrated with experiments on the 1/5-scale AutoRally platform and on a full-size commercial vehicle in autonomous driving tasks.

For the success of the proposed method, an accurate nominal model of the vehicle and a faithful and realistic representation of the disturbances and the chance constraints must be available. This invites several possible future research directions to extend the proposed methodology. For

instance, uncertainty can be reduced by improving the model by incorporating information learned about the changing driving conditions, leading to less conservative trajectories. Exploring modern learning methods to improve the vehicle and uncertainty models online using data collected while driving is, therefore, a natural next step. Along the same lines, future work may also explore the case when uncertainty directly affects the system model matrices and not only by entering the system as additive Gaussian noise.

APPENDIX A
PROOF OF THEOREM 2

First, we introduce the following well-known lemma [38], [58], which is easy to show, and hence, we state it without proof.

Lemma 3: Let x be a scalar random variable such that $x \sim \mathcal{N}(\mu, \Sigma)$. Then, the chance constraint

$$\Pr(x \leq \beta) \geq 1 - p \tag{A.1}$$

holds if and only if the following inequality holds:

$$\mu + \sqrt{\Sigma}\,\Phi^{-1}(1-p) - \beta \leq 0 \tag{A.2}$$

where $\Phi(\cdot)$ is the cumulative distribution of the standard normal distribution. Furthermore, if $p \in [0, 0.5)$, then $\Phi^{-1}(1-p) \geq 0$ and (A.2) implies that $\mu \leq \beta$.

To prove Theorem 2, we first rewrite the system dynamics in a more convenient form. Following [43], it is straightforward to obtain the following equivalent form of the system dynamics (14b):

$$X = \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{D}W + \mathcal{R} \tag{A.3}$$

where the augmented state is $X = [x_{k|k}^\top, x_{k+1|k}^\top, \ldots, x_{k+N|k}^\top]^\top \in \mathbb{R}^{(N+1)n_x}$, the augmented control vector is $U = [u_{k|k}^\top, u_{k+1|k}^\top, \ldots, u_{k+N-1|k}^\top]^\top \in \mathbb{R}^{Nn_u}$, and the augmented disturbance vector is $W = [w_k^\top, w_{k+1}^\top, \ldots, w_{k+N-1}^\top]^\top \in \mathbb{R}^{Nn_w}$, and where $\mathcal{A} \in \mathbb{R}^{(N+1)n_x \times n_x}$, $\mathcal{B} \in \mathbb{R}^{(N+1)n_x \times Nn_u}$, $\mathcal{D} \in \mathbb{R}^{(N+1)n_x \times Nn_w}$, and $\mathcal{R} \in \mathbb{R}^{(N+1)n_x}$ are augmented system matrices, the explicit expressions of which can be found, for example [43]. From (2) and (3), it is easy to see that the initial condition and noise vector W satisfy $\mathbb{E}[x_{k|k}W^\top] = 0$, $\mathbb{E}[WW^\top] = I_{Nn_w}$, $\mathbb{E}[x_{k|k}x_{k|k}^\top] = \Sigma_k + \mu_k\mu_k^\top$. Introducing the auxiliary matrices $E_{t-k} = [0_{n_x,(t-k)n_x}, I_{n_x}, 0_{n_x,(N-(t-k))n_x}] \in \mathbb{R}^{n_x \times (N+1)n_x}$ and $F_{t-k} = [0_{n_u,(t-k)n_u}, I_{n_u}, 0_{n_u,((N-1)-(t-k))n_u}] \in \mathbb{R}^{n_u \times Nn_u}$, we have that the state and control at time step t can be expressed in terms of X and U via $x_{t|k} = E_{t-k}X$ and $u_{t|k} = F_{t-k}U$ where $t = k, \ldots, k+N$.

Using the previous definitions, it is straightforward to show that problem (14) can be written as

$$\min_U \ \mathbb{E}\big[(X - X_g)^\top \bar{Q}(X - X_g) + U^\top \bar{R}U\big] \tag{A.4a}$$
$$\text{s.t. } X = \mathcal{A}x_{k|k} + \mathcal{B}U + \mathcal{D}W + \mathcal{R} \tag{A.4b}$$
$$x_{k|k} \sim \mathcal{N}(\mu_k, \Sigma_k) \tag{A.4c}$$
$$\Pr\big(\alpha_{x,i}^\top E_{t-k}X \leq \beta_{x,i}\big) \geq 1 - p_{x,i} \tag{A.4d}$$
$$\Pr\big(\alpha_{u,j}^\top F_{t-k}U \leq \beta_{u,j}\big) \geq 1 - p_{u,j} \tag{A.4e}$$
$$E_N\mathbb{E}[X] \in \mathcal{X}_f^\mu \tag{A.4f}$$
$$\Sigma_f \succ E_N\big(\mathbb{E}[XX^\top] - \mathbb{E}[X]\mathbb{E}[X]^\top\big)E_N^\top \tag{A.4g}$$

where the augmented target vector is given by $X_g = [x_{g_k}^\top, x_{g_{k+1}}^\top, \ldots, x_{g_{k+N}}^\top]^\top$, where $\bar{Q} = \mathtt{blkdiag}(Q_k, \ldots, Q_{k+N-1}, 0)$ and $\bar{R} = \mathtt{blkdiag}(R_k, \ldots, R_{k+N-1})$. Note that because $Q_t \succeq 0$ and $R_t \succ 0$, it follows that $\bar{Q} \succeq 0$ and $\bar{R} \succ 0$.

Problem (A.4) can be converted to the convex program

$$\begin{aligned}&\min_{V,K} \ J_N(\mu_k, \Sigma_k; V, K)\\ &= \mathtt{tr}\big[\big((I + \mathcal{B}K)^\top \bar{Q}(I + \mathcal{B}K) + K^\top \bar{R}K\big)\Sigma_y\big]\\ &\quad + (\mathcal{A}\mu_k + \mathcal{B}V + \mathcal{R} - X_g)^\top \bar{Q}(\star) + V^\top \bar{R}V \qquad \text{(A.5a)}\\ &\text{s.t. } \alpha_{x,i}^\top E_{t-k}(\mathcal{A}\mu_k + \mathcal{B}V + \mathcal{R})\\ &\quad + \|\Sigma_y^{1/2}(I + \mathcal{B}K)^\top E_{t-k}^\top \alpha_{x,i}\|\Phi^{-1}(1 - p_{x,i})\\ &\quad - \beta_{x,i} \leq 0, \quad i = 0, \ldots, N_s - 1 \qquad \text{(A.5b)}\\ &\quad \alpha_{u,j}^\top F_{t-k}V + \|\Sigma_y^{1/2}K^\top F_{t-k}^\top \alpha_{u,j}\|\Phi^{-1}\big(1 - p_{u,j}\big)\\ &\quad - \beta_{u,j} \leq 0, \quad j = 0, \ldots, N_c - 1 \qquad \text{(A.5c)}\\ &\quad E_N(\mathcal{A}\mu_k + \mathcal{B}V + \mathcal{R}) \in \mathcal{X}_f^\mu \qquad \text{(A.5d)}\\ &\quad \Sigma_f \succeq E_N(I + \mathcal{B}K)\Sigma_y(I + \mathcal{B}K)^\top E_N^\top \qquad \text{(A.5e)}\end{aligned}$$

where $\Sigma_y = \mathcal{A}\Sigma_k\mathcal{A}^\top + \mathcal{D}\mathcal{D}^\top$, and $V = [v_{k|k}^\top, \ldots, v_{k+N-1|k}^\top]^\top$, $X_g = [x_{g_k}^\top, \ldots, x_{g_{k+N}}^\top]^\top$, $K = \big[\mathtt{blkdiag}(K_{k|k}, \ldots, K_{k+N-1|k}), 0\big]$, $\bar{Q} = \mathtt{blkdiag}(Q_k, \ldots, Q_{k+N-1}, 0_{n_x \times n_x})$, and $\bar{R} = \mathtt{blkdiag}(R_k, \ldots, R_{k+N-1})$.

All steps to convert problem (A.4) to problem (A.5) have already been discussed in our previous work [38], [39], except for the conversion of (A.4e) to (A.5c), which we describe in the following, because neither [38] nor [39] include input constraints.

To this end, notice that using the control law (15), the control sequence U in (A.4e) is represented as $U = V + KY$, where $Y = \big[y_{k|k}^\top \cdots y_{k+N|k}^\top\big]^\top$. It follows from (15b) that $Y = \mathcal{A}y_{k|k} + \mathcal{D}W$, and thus, using the facts that $\mathbb{E}[y_{k|k}] = 0$, $\mathbb{E}[y_{k|k}y_{k|k}^\top] = \Sigma_k$, and $\mathbb{E}[y_{k|k}W^\top] = 0$, one obtains $\mathbb{E}[Y] = 0$ and $\mathbb{E}[YY^\top] = \Sigma_y$. Therefore, $\mathbb{E}[U] = V$ and $\mathbb{E}[(U - \mathbb{E}[U])(\star)^\top] = K\Sigma_yK^\top$. The inequality (A.4e) can be rewritten as

$$\Pr\big(\alpha_{u,j}^\top F_{t-k}(V + KY) \leq \beta_{u,j}\big) \geq 1 - p_{u,j}. \tag{A.6}$$

Notice that $\alpha_{u,j}^\top F_{t-k}(V + KY)$ is a Gaussian distributed scalar random variable with mean $\alpha_{u,j}^\top F_{t-k}V$ and variance $\alpha_{u,j}^\top F_{t-k}K\Sigma_yK^\top F_{t-k}^\top \alpha_{u,j}$. Using Lemma 3, we obtain that (A.6) is equivalent to the inequality $\alpha_{u,j}^\top F_{t-k}V - \beta_{u,j} + \sqrt{\alpha_{u,j}^\top F_{t-k}K\Sigma_yK^\top F_{t-k}^\top \alpha_{u,j}}\,\Phi^{-1}(1 - p_{u,j}) \leq 0$, which can be readily converted to (A.5c). Since $p_{x,i} \in [0, 0.5)$ and $p_{u,j} \in [0, 0.5)$, the constraints (A.5b) and (A.5c) are convex [38], [58].

Finally, the terminal covariance constraint (A.5e) can be converted to a linear matrix inequality (LMI)

$$\begin{bmatrix} \Sigma_f & E_N(I + \mathcal{B}K)\Sigma_y^{1/2} \\ \Sigma_y^{1/2}(I + \mathcal{B}K)^\top E_N^\top & I \end{bmatrix} \succeq 0 \tag{A.7}$$

which is a convex constraint in terms of the matrix K.

APPENDIX B
PROOF OF THEOREM 4

To simplify the proof, we assume that the cost matrices and the goal state are constant. That is, for all $t = 0, 1, \ldots$, we assume that $Q_t = Q$, $R_t = R$, and $x_{g_t} = x_g$, and thus,

$\ell_t(\cdot,\cdot) = \ell(\cdot,\cdot)$. Note, however, that the proof still holds for the time-varying case, provided the cost matrices and the target vector are bounded as per the problem definition.

Using (24) and (25), the cost $J_N(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}; \mathbf{u}^{N-1}_{k+1|k+1})$, where $\mathbf{u}^{N-1}_{k+1|k+1}$ is defined in (23), can be written as $J_N(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}; \mathbf{u}^{N-1}_{k+1|k+1}) = J^*_N(\mu^*_{k|k}, \Sigma^*_{k|k}) - \tilde{\ell}(\mu^*_{k|k}, \Sigma^*_{k|k}, v^*_{k|k}, K^*_{k|k}) + \ell(\mu^*_{k+N|k}, \Sigma^*_{k+N|k}, v_{k+N|k+1}, K_{k+N|k+1})$, where $(\mu^*_{k|k}, \Sigma^*_{k|k}) = (\mu_k, \Sigma_k)$ and the pair (μ_k, Σ_k) is given by the initialization (19). Hence,

$$\begin{aligned} &J_N\left(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}; \mathbf{u}^{N-1}_{k+1|k+1}\right) - J^*_N\left(\mu^*_{k|k}, \Sigma^*_{k|k}\right) \\ &\quad + \tilde{\ell}\left(\mu^*_{k|k}, \Sigma^*_{k|k}, v^*_{k|k}, K^*_{k|k}\right) \\ &\quad = \mathtt{tr}\left(Q\Sigma^*_{k+N|k}\right) + v^\top_{k+N|k+1} R\, v_{k+N|k+1} \\ &\quad\quad + \mathtt{tr}\left(RK_{k+N|k+1}\Sigma_{y_{k+N|k+1}}K^\top_{k+N|k+1}\right) \\ &\quad\quad + \left(\mu^*_{k+N|k} - x_g\right)^\top Q\left(\mu^*_{k+N|k} - x_g\right). \end{aligned} \tag{B.8}$$

Let now $J^{\mathcal{X}^\mu_f}_{\max} = \max_{\mu\in\mathcal{X}^\mu_f}[(\mu - x_g)^\top Q(\mu - x_g)]$. Since $\mu^*_{k+N|k} \in \mathcal{X}^\mu_f$, it follows that: $(\mu^*_{k+N|k} - x_g)^\top Q\,(\mu^*_{k+N|k} - x_g) \le J^{\mathcal{X}^\mu_f}_{\max}$ and the last term in (B.8) is bounded by $J^{\mathcal{X}^\mu_f}_{\max}$. Similarly, since $\Sigma^*_{k+N|k} \preceq \Sigma_f$, it follows that the first term in (B.8) is bounded by $\mathtt{tr}(Q\Sigma_f)$.

By Assumption 3, $v_{k+N|k+1}$ and $K_{k+N|k+1}$ are feasible controls, meaning that $u_{k+N|k+1}$ satisfies the chance constraint (14e). Furthermore, $u_{k+N|k+1} \sim \mathcal{N}(v_{k+N|k+1}, \Sigma_{u_{k+N|k+1}})$ and, from Lemma 3, it follows that

$$\begin{aligned} \alpha^\top_{u,j} v_{k+N|k+1} + \sqrt{\alpha^\top_{u,j}\Sigma_{u_{k+N|k+1}}\alpha_{u,j}}\,\Phi^{-1}(1-p_{u,j}) \\ - \beta_{u,j} \le 0, \quad j = 0,\ldots,N_c - 1 \end{aligned} \tag{B.9}$$

where $\Sigma_{u_{k+N|k+1}} = K_{k+N|k+1}\Sigma_{y_{k+N|k+1}}K^\top_{k+N|k+1}$. Since $p_{u,j} \in [0, 0.5)$ the previous inequality implies, in particular, that, for all $j = 0,\ldots,N_c - 1$

$$\alpha^\top_{u,j} v_{k+N|k+1} - \beta_{u,j} \le 0 \tag{B.10}$$

or equivalently, $v_{k+N|k+1} \in \mathcal{U}$. It follows that the second term in (B.8) is bounded by $J^v_{\max} = \max_{v\in\mathcal{U}} v^\top R\, v$. Similarly, (B.9) implies that

$$\alpha^\top_{u,j}\Sigma_{u_{k+N|k+1}}\alpha_{u,j} \le \left(\frac{\beta_{u,j}}{\Phi^{-1}\left(1-p_{u,j}\right)}\right)^2 \tag{B.11}$$

and, therefore, the third term in (B.8) is bounded by $J^K_{\max} = \max_{\Sigma_u\in\mathcal{S}} \mathtt{tr}(R\Sigma_u)$ where $\mathcal{S} = \{\Sigma_u : \alpha^\top_{u,j}\Sigma_u\alpha_{u,j} \le \beta^2_{u,j}/\Phi^{-1}(1-p_{u,j})^2,\ j = 0,\ldots,N_c-1\}$.

Let $\delta = J^{\mathcal{X}^\mu_f}_{\max} + \mathtt{tr}(Q\Sigma_f) + J^v_{\max} + J^K_{\max}$. Since $J^*_N(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}) \le J_N(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}; \mathbf{u}^{N-1}_{k+1|k+1})$, it follows from (24) and (B.8) that:

$$\begin{aligned} \mathbb{E}\left[\ell\left(x_{k|k}, u^*_{k|k}\right)\right] &\le J^*_N\left(\mu^*_{k|k}, \Sigma^*_{k|k}\right) \\ &\quad - J^*_N\left(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}\right) + \delta \\ \mathbb{E}\left[\ell\left(x^*_{k+1|k}, u^*_{k+1|k}\right)\right] &\le J^*_N\left(\mu^*_{k+1|k}, \Sigma^*_{k+1|k}\right) \\ &\quad - J^*_N\left(\mu^*_{k+2|k}, \Sigma^*_{k+2|k}\right) + \delta \\ &\vdots \\ \mathbb{E}\left[\ell\left(x^*_{k+n-1|k}, u^*_{k+n-1|k}\right)\right] &\le J^*_N\left(\mu^*_{k+n-1|k}, \Sigma^*_{k+n-1|k}\right) \\ &\quad - J^*_N\left(\mu^*_{k+n|k}, \Sigma^*_{k+n|k}\right) + \delta \end{aligned}$$

and, thus,

$$\begin{aligned} &\frac{1}{n}\sum_{t=0}^{n-1}\mathbb{E}\left[\ell\left(x^*_{k+t|k}, u^*_{k+t|k}\right)\right] \\ &\quad \le \frac{1}{n}\left(J^*_N(\mu_k, \Sigma_k) - J^*_N\left(\mu^*_{k+n|k}, \Sigma^*_{k+n|k}\right)\right) + \delta \\ &\quad \le \frac{1}{n}J^*_N(\mu_k, \Sigma_k) + \delta \end{aligned} \tag{B.12}$$

where for the last inequality, we used the fact that $J^*_N(\mu^*_{k+n|k}, \Sigma^*_{k+n|k}) \ge 0$. The result now follows by taking $\ell_{\max} = \delta$ and letting $n \to \infty$.

ACKNOWLEDGMENT

The authors would like to thank Ji Yin and Zhiyuan Zhang for assisting with the AutoRally platform experiments, and Austin Costley and Nate Bunderson of Autonomous Solutions Inc. (ASI), Petersboro, UT, USA, for allowing the use of their vehicle and test facilities.

REFERENCES

[1] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2736–2743, Nov. 2020.

[2] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 1433–1440.

[3] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)," *Control Eng. Pract.*, vol. 95, Feb. 2020, Art. no. 104270.

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

[5] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014.

[6] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, Dec. 2014.

[7] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 502–517, Jun. 2010.

[8] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*. London, U.K.: Springer, 1999, pp. 207–226.

[9] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic linear model predictive control with chance constraints—A review," *J. Process Control*, vol. 44, pp. 53–67, Aug. 2016.

[10] B. Kouvaritakis and M. Cannon, *Model Predictive Control*. Cham, Switzerland: Springer, 2016.

[11] D. M. Raimondo, D. Limon, M. Lazar, L. Magni, and E. F. Camacho, "Min-max model predictive control of nonlinear systems: A unifying overview on stability," *Eur. J. Control*, vol. 15, no. 1, pp. 5–21, Jan. 2009.

[12] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, Jan. 2004.

[13] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, Feb. 2005.

[14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

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

[16] M. Cannon, B. Kouvaritakis, and D. Ng, "Probabilistic tubes in linear stochastic model predictive control," *Syst. Control Lett.*, vol. 58, nos. 10–11, pp. 747–753, Oct. 2009.

[17] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *Proc. Int. Symp. Adv. Vehicle Control*, Tokyo, Japan, Sep. 2014, pp. 712–719.

[18] P. Hokayem, E. Cinquemani, D. Chatterjee, F. Ramponi, and J. Lygeros, "Stochastic receding horizon control with output feedback and bounded controls," *Automatica*, vol. 48, no. 1, pp. 77–88, Jan. 2012.

[19] F. Oldewurtel, C. N. Jones, and M. Morari, "A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback," in *Proc. 47th IEEE Conf. Decis. Control*, Cancún, Mexico, Dec. 2008, pp. 4731–4736.

[20] D. Bernardini and A. Bemporad, "Scenario-based model predictive control of stochastic constrained linear systems," in *Proc. 48th IEEE Conf. Decis. Control (CDC)*, Shanghai, China, Dec. 2009, pp. 6333–6338.

[21] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 219–224, Jan. 2013.

[22] M. Cannon, B. Kouvaritakis, S. V. Raković, and Q. Cheng, "Stochastic tubes in model predictive control with probabilistic constraints," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 194–200, Jan. 2011.

[23] M. Lorenzen, M. A. Müller, and F. Allgöwer, "Stochastic model predictive control without terminal constraints," *Int. J. Robust Nonlinear Control*, vol. 29, no. 15, pp. 4987–5001, Oct. 2019.

[24] J. A. Primbs and C. H. Sung, "Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 221–230, Feb. 2009.

[25] M. Farina, L. Giulioni, L. Magni, and R. Scattolini, "A probabilistic approach to model predictive control," in *Proc. 52nd IEEE Conf. Decis. Control*, Firenze, Italy, Dec. 2013, pp. 7734–7739.

[26] M. Farina, L. Giulioni, L. Magni, and R. Scattolini, "An approach to output-feedback MPC of stochastic linear discrete-time systems," *Automatica*, vol. 55, pp. 140–149, May 2015.

[27] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, Apr. 2006.

[28] P. Hokayem, D. Chatterjee, and J. Lygeros, "On stochastic receding horizon control with bounded control inputs," in *Proc. 48th IEEE Conf. Decis. Control (CDC)*, Shanghai, China, Dec. 2009, pp. 6359–6364.

[29] P. Hokayem, D. Chatterjee, F. Ramponi, G. Chaloulos, and J. Lygeros, "Stable stochastic receding horizon control of linear systems with bounded control inputs," in *Proc. Int. Symp. Math. Theory Netw. Syst.*, Budapest, Hungary, Jul. 2010, pp. 31–36.

[30] J. A. Paulson, E. A. Buehler, R. D. Braatz, and A. Mesbah, "Stochastic model predictive control with joint chance constraints," *Int. J. Control*, vol. 93, no. 1, pp. 126–139, Jan. 2020.

[31] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1743–1752, Aug. 1999.

[32] M. Ono, "Joint chance-constrained model predictive control with probabilistic resolvability," in *Proc. Amer. Control Conf. (ACC)*, Montreal, QC, Canada, Jun. 2012, pp. 435–441.

[33] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzmán, and F. Rodríguez, "Online robust tube-based MPC for time-varying systems: A practical approach," *Int. J. Control*, vol. 84, no. 6, pp. 1157–1170, Jun. 2011.

[34] B. Kouvaritakis and M. Cannon, "Developments in robust and stochastic predictive control in the presence of uncertainty," *ASCE-ASME J. Risk Uncertainty Eng. Syst. B, Mech. Eng.*, vol. 1, no. 2, Jun. 2015, Art. no. 021003.

[35] M. Cannon, Q. Cheng, B. Kouvaritakis, and S. V. Raković, "Stochastic tube MPC with state estimation," *Automatica*, vol. 48, no. 3, pp. 536–541, Mar. 2012.

[36] J. A. Paulson, S. Streif, and A. Mesbah, "Stability for receding-horizon stochastic model predictive control," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, Jul. 2015, pp. 937–943.

[37] R. González, M. Fiacchini, J. L. Guzmán, T. Álamo, and F. Rodríguez, "Robust tube-based predictive control for mobile robots in off-road conditions," *Robot. Auto. Syst.*, vol. 59, no. 10, pp. 711–726, Oct. 2011.

[38] K. Okamoto, M. Goldshtein, and P. Tsiotras, "Optimal covariance control for stochastic systems under chance constraints," *IEEE Control Syst. Lett.*, vol. 2, no. 2, pp. 266–271, Apr. 2018.

[39] K. Okamoto and P. Tsiotras, "Optimal stochastic vehicle path planning using covariance steering," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2276–2281, Jul. 2019.

[40] A. Hotz and R. E. Skelton, "Covariance control theory," *Int. J. Control*, vol. 46, no. 1, pp. 13–32, Jul. 1987.

[41] Y. Chen, T. T. Georgiou, and M. Pavon, "Optimal steering of a linear stochastic system to a final probability distribution. Part I," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1158–1169, May 2016.

[42] E. Bakolas, "Optimal covariance control for discrete-time stochastic linear systems subject to constraints," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Las Vegas, NV, USA, Dec. 2016, pp. 1153–1158.

[43] M. Goldshtein and P. Tsiotras, "Finite-horizon covariance control of linear time-varying systems," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Melbourne, VIC, Australia, Dec. 2017, pp. 3606–3611.

[44] A. Prékopa, "Boole–Bonferroni inequalities and linear programming," *Oper. Res.*, vol. 36, no. 1, pp. 145–162, Feb. 1988.

[45] J. Pilipovsky and P. Tsiotras, "Covariance steering with optimal risk allocation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 3719–3733, Dec. 2021.

[46] M. Korda, R. Gondhalekar, J. Cigler, and F. Oldewurtel, "Strongly feasible stochastic model predictive control," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 1245–1251.

[47] D. Chatterjee, F. Ramponi, P. Hokayem, and J. Lygeros, "On mean square boundedness of stochastic linear systems with bounded controls," *Syst. Control Lett.*, vol. 61, no. 2, pp. 375–380, Feb. 2012.

[48] L. Hewing and M. N. Zeilinger, "Stochastic model predictive control for linear systems using probabilistic reachable sets," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, Dec. 2018, pp. 5182–5188.

[49] MOSEK ApS. (2017). *The MOSEK Optimization Toolbox for MATLAB Manual. Version 8.1*. [Online]. Available: http://docs.mosek.com

[50] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, 2008.

[51] B. Goldfain et al., "AutoRally: An open platform for aggressive autonomous driving," *IEEE Control Syst. Mag.*, vol. 39, no. 1, pp. 26–55, Feb. 2019.

[52] GeorgiaTech. (2022). *AutoRally*. [Online]. Available: https://autorally.github.io/

[53] Y. Zhao and P. Tsiotras, "A quadratic programming approach to path smoothing," in *Proc. Amer. Control Conf.*, San Francisco, CA, USA, Jun. 2011, pp. 5324–5329.

[54] E. Velenis, E. Frazzoli, and P. Tsiotras, "Steady-state cornering equilibria and stabilisation for a vehicle during extreme operating conditions," *Int. J. Vehicle Auton. Syst.*, vol. 8, nos. 2–4, pp. 217–241, 2010.

[55] M. Acosta and S. Kanarachos, "Tire lateral force estimation and grip potential identification using neural networks, extended Kalman filter, and recursive least squares," *Neural Comput. Appl.*, vol. 30, no. 11, pp. 3445–3465, Dec. 2018.

[56] C. You and P. Tsiotras, "Vehicle modeling and parameter estimation using adaptive limited memory joint-state UKF," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 322–327.

[57] *Mechanical Simulation*. Accessed: Jun. 16, 2022. [Online]. Available: https://www.carsim.com/

[58] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1080–1094, Dec. 2011.