

SDP Synthesis of Maximum Coverage Trees for Probabilistic Planning under Control Constraints

Naman Aggarwal and Jonathan P. How

Laboratory for Information and Decision Systems, Massachusetts Institute of Technology

Abstract—The paper presents Maximal Covariance Backward Reachable Trees (MAXCOVAR BRT), which is a multi-query algorithm for planning of dynamic systems under stochastic motion uncertainty and constraints on the control input with explicit coverage guarantees. In contrast to existing roadmap-based probabilistic planning methods that sample belief nodes randomly and draw edges between them [1], under control constraints, the reachability of belief nodes needs to be explicitly established and is determined by checking the feasibility of a non-convex program. Moreover, there is no explicit consideration of coverage of the roadmap while adding nodes and edges during the construction procedure for the existing methods. Our contribution is a novel optimization formulation to add nodes and construct the corresponding edge controllers such that the generated roadmap results in provably maximal coverage under control constraints as compared to any other method of adding nodes and edges. We characterize formally the notion of coverage of a roadmap in this stochastic domain via introduction of the h-BRS (Backward Reachable Set of Distributions) of a tree of distributions under control constraints, and also support our method with extensive simulations on a 6 DoF model.

I. INTRODUCTION

Multi-query motion planning entails the design of plans that can be reused across different initial configurations of the system. This is typically done via the offline construction of a roadmap of feasible trajectories in the state space, such that in real-time, for a pair of initial and goal configurations of the system, planning proceeds by connecting the initial and goal configurations to the roadmap followed by graph search to find a path [2, 3]. The pre-computation of a roadmap is beneficial since it avoids the computational burden of finding plans from scratch for new configurations, and reuses search effort across queries. Such an approach of constructing a roadmap of feasible trajectories facilitates robust planning since a plan to meet the mission requirements could be computed quickly if the system finds itself in unexpected configurations due to unmodeled dynamics or disturbances during online operation. An important design consideration for any roadmap based planning algorithm is the coverage of the roadmap – it is desirable to be able to re-use search effort across as many queries as possible, therefore, reasoning explicitly about the coverage of the roadmap becomes an important algorithmic design consideration.

There is extensive work on motion planning via the construction of reusable roadmaps for deterministic systems [4–7]. The work proposed in [4], for instance, builds a tree of *funnels*, which are regions of finite-time invariance around trajectories with associated feedback controllers, backwards

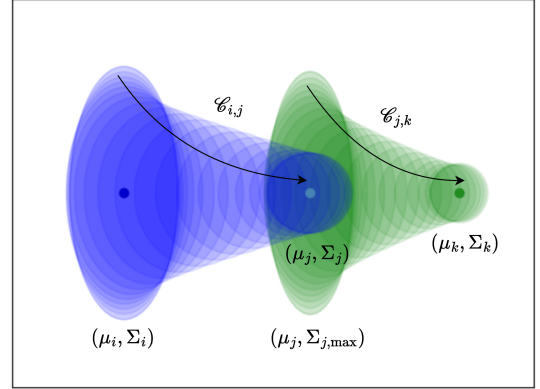


Fig. 1. (Recursive Feasibility through Sequential Composition) An illustration of the satisfaction of constraints (goal distribution reaching constraint, and chance constraints on the state and control input) for the $2N$ -step trajectory initialized at the (μ_i, Σ_i) distribution driven through a sequential composition of two N -step controllers, $\mathcal{C}_{i,j}$ (designed for the $i \rightarrow j$ maneuver) and $\mathcal{C}_{j,k}$ (designed for the $j_{\max} \rightarrow k$ maneuver). The distribution (μ_i, Σ_i) is steered to the goal distribution (μ_k, Σ_k) in $2N$ -steps under the concatenated controller s.t. all chance constraints on the state and control input are satisfied for the $2N$ -step trajectory (refer to Lemma 1 for a formalization).

from the goal in a space-filling manner. Paths to the goal could be found from different regions of the state space by traversing the branches of the constructed tree. Ref. [6] takes a different approach by populating a library with a finite set of feasible trajectories with associated funnels, such that the online planning proceeds through the sequential composition of the available funnels in the library.

In this paper, we are concerned with planning for systems with stochastic dynamics. A stochastic framework helps by making informed plans considering the probability of constraint violation and leads to a lesser conservative approach as compared to a deterministic framework considering worst-case uncertainty in the dynamics or enforcing hard constraints. CC-RRT [8] is a popular framework for single-query motion planning in stochastic systems. The algorithm proceeds by growing a forwards search tree of distributions over the state space, such that the obstacle avoidance constraints are satisfied probabilistically along each edge. The tree expansion phase involves randomly sampling open-loop control signals and simulating the system dynamics forward until the chance constraints on obstacle avoidance are satisfied, and adding the forward simulated state distribution to the tree as a node with the associated open-loop control sequence

as the edge. A major limitation of constructing a search tree via sampling open-loop control signals is that the second order moment of the state distributions (covariance) cannot be controlled via open-loop control even for the simplest case of linear-Gaussian systems. CC-RRT comes with a probabilistic completeness guarantee such that the algorithm finds paths to all goal distributions for which open-loop paths would exist [9]. However, the set of all possible distributions that could be reached with open-loop control is a subset of the set of all distributions that could be reached with a richer control scheme such as feedback control. Particularly, for a mission with tight goal reaching constraints (a specific size of the goal covariance as required by the mission), CC-RRT might fail to find a path even though a closed-loop path might exist.

While CC-RRT is a single-query planner, there exist other roadmap based methods for planning in stochastic domains. FIRM [10] is a planning framework that builds a roadmap in the belief space by sampling stationary belief nodes and drawing belief stabilizing controllers as edges between the nodes. The closest in approach to our work, however, is the recently proposed CS-BRM [11] that provides a faster planning scheme as compared to FIRM by allowing sampling of non-stationary belief nodes and adding finite-time feedback controllers as edges. CS-BRM is built on top of a series of latest work on covariance steering that allows finite-time satisfaction of terminal covariance constraints [12–14]. CS-BRM, however, does not assume control input constraints, and adds edges between randomly sampled belief nodes. In the absence of control constraints, the distribution steering problem decouples into the mean steering and covariance steering problem: both of which are tractable to solve [14]. In the presence of control input constraints however, the reachability of belief nodes needs to be explicitly established. Even for the simplest case of a controllable linear-Gaussian system, it is no longer possible to drive arbitrary distributions to arbitrary distributions, and the existence of a steering maneuver for a pair of belief nodes must be determined by checking the feasibility of a nonconvex program. Therefore, CS-BRM is not efficient for settings with constraints on the control input due to unnecessary sampling of belief nodes that would not be added to the roadmap due to the non-existence of a steering control.

Moreover, even for successful insertion of belief nodes, there is neither an explicit consideration of coverage while selecting nodes during the roadmap construction procedure, nor a post-analysis on the coverage characteristics of the roadmap generated by CS-BRM. We provide a systematic procedure that reasons about the coverage of the to-be-generated roadmap while adding nodes through a novel optimization formulation. The contributions of the paper are as follows:

- We characterize the notion of coverage of a roadmap in the stochastic domain formally via what we call h-BRS (Backward Reachable Set of Distributions) of the tree, which is the set of all distributions that can reach the

goal in h-hops where 1 hop is a finite-time horizon of N-steps under control constraints.

- We propose a novel method of edge construction such that the BRT constructed following our method finds paths from provably the largest set of initial distributions for the overall planning problem, i.e. provides maximal coverage.
- We supplement our approach with extensive simulations on a 6 DoF model.

II. PROBLEM STATEMENT

Consider a discrete-time stochastic linear system represented by the following difference equation

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + D\mathbf{w}_t \quad (1)$$

where $\mathbf{x}_t \in \mathcal{X}$, $\mathbf{u}_t \in \mathcal{U}$ are the state and control inputs at time t , respectively, $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{U} \subseteq \mathbb{R}^m$ is the control space, $n, m \in \mathbb{N}$, $D \in \mathbb{R}^{n \times n}$ and \mathbf{w}_t represents the stochastic disturbance at time t that is assumed to have a Gaussian distribution with zero mean and unitary covariance. We also assume that $\{\mathbf{w}_t\}$ is an *i.i.d.* process, and that A is non-singular. Note that we use the notation $\mathcal{N}(\mu, \Sigma)$ to denote a Gaussian distribution with mean μ and covariance Σ .

We define a finite-horizon optimal control problem, OPTSTEER($\mathcal{I}, \mathcal{G}, N$), where $\mathcal{I}, \mathcal{G}, N$ are the initial distribution, goal distribution, and the time-horizon corresponding to the control problem respectively. OPTSTEER($\mathcal{I}, \mathcal{G}, N$) solves for a control sequence that is optimal with respect to a performance index J , and that steers the system from an initial distribution \mathcal{I} to a goal distribution \mathcal{G} in a time-horizon of length N while avoiding obstacles such that the control sequence respects the prescribed constraints on the control input at each time-step:

$$\min_{\Phi_k} J = \mathbb{E} \left[\sum_{k=0}^{N-1} \mathbf{x}_k^\top Q_k \mathbf{x}_k + \mathbf{u}_k^\top R_k \mathbf{u}_k \right] \quad (\text{OPTSTEER})$$

such that, for all $k = 0, 1, \dots, N-1$,

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + D\mathbf{w}_k, \quad (2a)$$

$$\mathbf{x}_0 \sim \mathcal{N}(\mu_{\mathcal{I}}, \Sigma_{\mathcal{I}}), \quad (2b)$$

$$\mathbf{x}_N \sim \mathcal{N}(\mu_{\mathcal{G}}, \Sigma_{\mathcal{G}}), \quad (2c)$$

$$\mu_N = \mu_{\mathcal{G}}, \quad (2d)$$

$$\Sigma_N \preceq \Sigma_{\mathcal{G}}, \quad (2e)$$

$$\mathbb{P}(\mathbf{x}_k \in \mathcal{X}) \geq 1 - \epsilon_x, \quad (2f)$$

$$\mathbb{P}(\mathbf{u}_k \in \mathcal{U}) \geq 1 - \epsilon_u, \quad (2g)$$

where $\Phi_k(\cdot)$ is the parameterization for the finite-horizon controller such that $\mathbf{u}_k := \Phi_k(\mathbf{x}_k)$.

Under constraints on the control input of the form (2g), not all *instances* of OPTSTEER($\mathcal{I}, \mathcal{G}, N$) have a solution where an *instance* is described by the tuple (\mathcal{I}, N) of the initial distribution and the planning horizon for a fixed goal distribution \mathcal{G} . In particular, OPTSTEER($\mathcal{I}, \mathcal{G}, N$) has a

solution if and only if the constraint set of the optimization problem defined by the set of equations (2a)-(2g) is non-empty in the space of the decision variables $\{\Phi_k\}_{k=0}^{N-1}$. Alternatively, under control constraints, the existence of an N -step controller that steers the system from an initial distribution to a goal distribution needs to be established explicitly by solving a feasibility question even when the underlying system (A, B) is controllable. This existence (or not) of a control sequence for a planning instance defined by the triplet $\mathcal{I}, \mathcal{G}, N$ is a special artifact of the presence of control input constraints in distribution steering (see Section III). This paper presents solutions to the following problem:

Problem II.1. Find paths to the goal distribution \mathcal{G} from all query initial distributions \mathcal{I} for which paths (respecting all the constraints) exist.

Note that we use the terms *path* and *control sequence* interchangeably in this paper. Problem II.1 is an instance of multi-query planning, and is a difficult problem to solve in it's full generality. We employ graph-based methods to build a roadmap in the space of distributions of the state of the system such that the roadmap could be used across query initial distributions.

We propose a novel method of edge construction such that the edge controller can be re-used across a wider set of initial distributions for the edge, and such that the BRT constructed following the novel method of edge construction finds paths from provably the largest set of initial distributions for the overall planning task. We supplement our approach with extensive simulations.

We begin with a discussion on the feasibility of a covariance steering problem instance in Section III, followed by a discussion of our approach in Section IV. We conclude by a discussion of our experiments in Section V.

III. FINITE HORIZON COVARIANCE STEERING UNDER CONTROL CONSTRAINTS

In this section, we discuss the finite horizon covariance steering problem, and the feasibility of any covariance steering problem instance. The development in this section follows closely [13] [14]. We consider a linear state feedback parameterization for the controller to solve (OPTSTEER) as follows,

$$\mathbf{u}_k = K_k(\mathbf{x}_k - \mu_k) + v_k \quad (3)$$

where $K_k \in \mathbb{R}^{n \times m}$ is the feedback gain that controls the covariance dynamics, and $v_k \in \mathbb{R}^n$ is the feedforward gain that controls the mean dynamics. Under a linearly parameterized controller as described in (3), the state distribution remains Gaussian at all times and we can express the objective J (OPTSTEER) completely in terms of the first and second order moments of the state process:

$$J = \sum_{k=0}^{N-1} \text{tr}(Q_k \Sigma_k) + \text{tr}(R_k K_k \Sigma_k K_k^\top) + \mu_k^\top Q_k \mu_k + v_k^\top R_k v_k,$$

We consider polytopic state and control constraints of the form $\mathcal{X} := \{x_k \in \mathbb{R}^n \mid \alpha_x^\top x_k \leq \beta_x\}$, $\mathcal{U} := \{u_k \in \mathbb{R}^m \mid \alpha_u^\top u_k \leq \beta_u\}$ such that,

$$\mathbb{P}(\alpha_x^\top \mathbf{x}_k \leq \beta_x) \geq 1 - \epsilon_x, \quad (4)$$

$$\mathbb{P}(\alpha_u^\top \mathbf{u}_k \leq \beta_u) \geq 1 - \epsilon_u, \quad (5)$$

where $\alpha_x \in \mathbb{R}^n$, $\alpha_u \in \mathbb{R}^m$, and $\beta_x, \beta_u \in \mathbb{R}$. $\epsilon_x, \epsilon_u \in [0, 0.5]$ represent the tolerance levels with respect to state and control constraint violation respectively, and $\{\alpha_x \mathbf{x}_k\}_{k=0}^{N-1}$ and $\{\alpha_u \mathbf{u}_k\}_{k=0}^{N-1}$ are univariate random variables with first and second order moments,

$$\mathbb{E}(\alpha_x \mathbf{x}_k) = \alpha_x \mu_k \quad (6)$$

$$\mathbb{E}(\alpha_u \mathbf{u}_k) = \alpha_u v_k \quad (7)$$

$$\mathbb{E}(\alpha_x^\top (\mathbf{x}_k - \mu_k)(\mathbf{x}_k - \mu_k)^\top \alpha_x) = \alpha_x^\top \Sigma_k \alpha_x \quad (8)$$

$$\mathbb{E}(\alpha_u^\top (\mathbf{u}_k - v_k)(\mathbf{u}_k - v_k)^\top \alpha_u) = \alpha_u^\top K_k \Sigma_k K_k^\top \alpha_u. \quad (9)$$

Ref. [12] shows that the chance constraints can be written as,

$$\Phi^{-1}(1 - \epsilon_x) \sqrt{\alpha_x^\top \Sigma_k \alpha_x} + \alpha_x^\top \mu_k - \beta_x \leq 0, \quad (10a)$$

$$\Phi^{-1}(1 - \epsilon_u) \sqrt{\alpha_u^\top K_k \Sigma_k K_k^\top \alpha_u} + \alpha_u^\top v_k - \beta_u \leq 0, \quad (10b)$$

where $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of the normal distribution. Therefore, the optimization problem OPTSTEER can be recast as the nonlinear program,

$$\begin{aligned} \min_{\Sigma_k, K_k, \mu_k, v_k} J = \sum_{k=0}^{N-1} & \text{tr}(Q_k \Sigma_k) + \text{tr}(R_k K_k \Sigma_k K_k^\top) \\ & + \mu_k^\top Q_k \mu_k + v_k^\top R_k v_k, \end{aligned} \quad (11)$$

such that for all $k = 0, 1, \dots, N-1$,

$$\begin{aligned} \Sigma_{k+1} = & A \Sigma_k A^\top + B K_k \Sigma_k A^\top + A \Sigma_k K_k^\top B^\top \\ & + B K_k \Sigma_k K_k^\top B^\top + D D^\top, \end{aligned} \quad (11a)$$

$$\Sigma_0 = \Sigma_{\mathcal{I}}, \quad (11b)$$

$$\Sigma_N \preceq \Sigma_{\mathcal{G}}, \quad (11c)$$

$$\mu_{k+1} = A \mu_k + B v_k, \quad (11d)$$

$$\mu_0 = \mu_{\mathcal{I}}, \quad (11e)$$

$$\mu_N = \mu_{\mathcal{G}}, \quad (11f)$$

$$\Phi^{-1}(1 - \epsilon_x) \sqrt{\alpha_x^\top \Sigma_k \alpha_x} + \alpha_x^\top \mu_k - \beta_x \leq 0, \quad (11g)$$

$$\Phi^{-1}(1 - \epsilon_u) \sqrt{\alpha_u^\top K_k \Sigma_k K_k^\top \alpha_u} + \alpha_u^\top v_k - \beta_u \leq 0. \quad (11h)$$

The existence of an N -step steering control that drives the state distribution from \mathcal{I} to \mathcal{G} satisfying all the constraints is determined by the feasibility of the set of equations (11a)-(11h) which represents a non-convex set in the space of the decision variables $\{K_k\}_{k=0}^{N-1}, \{v_k\}_{k=0}^{N-1}$. Determining if a non-convex set is empty or not from it's algebraic description is an NP-hard problem in general, and the complexity of this feasibility check scales with the time-horizon since the problem size becomes larger and the number of variables increase. Problem II.1 is concerned with finding feasible paths from all possible initial distributions, and our solution

methodology proceeds by building a backward reachable tree of feasible paths from the goal distribution, and sequencing them together to find a feasible path at run-time for the query initial distribution, hence avoiding solving for the feasible path of the query distribution from scratch (see Section IV for details).

IV. MAXCOVAR BRT: A MAXIMUM COVERAGE TREE FOR PROBABILISTIC PLANNING

We solve Problem II.1 by constructing a Backward Reachable Tree (BRT) of distributions that verifiably reach the goal distribution \mathcal{G} under constraints on the control input. As discussed previously, in the presence of control constraints, existence of a control sequence that steers the system from an initial distribution to a goal distribution is established by solving a feasibility problem. The size of this feasibility check scales with the time-horizon and the BRT enables a faster feasibility check on a long time-horizon by checking the feasibility of reaching any existing node on the BRT instead of directly checking feasibility against the goal distribution.

We refer to this idea as *recursive feasibility* since the branches of the tree can be thought of as carrying a certificate of feasibility along its edges from the goal in a backwards fashion s.t. guaranteeing feasibility to any of the children nodes in the tree guarantees feasibility to all the upstream parent nodes and consequently the root node that corresponds to the goal distribution.

We introduce a novel objective function MAXCOVAR, as discussed in the following subsection, for adding nodes and constructing edge controllers such that the resulting tree provides maximum coverage. We also characterize formally the notion of coverage mathematically in this section.

A. MAXCOVAR: Novel Objective for Construction of the Edge Controller

We define a procedure to construct an N -step edge controller as follows. The procedure MAXCOVAR takes in as input a candidate initial mean μ_{initial} , a target distribution at the end of the N -step steering maneuver $(\mu_{\text{target}}, \Sigma_{\text{target}})$ and computes an initial covariance Σ_{initial} in a *maximal* sense, henceforth referred to as $\Sigma_{\text{initial}, \text{max}}$, and the associated control sequence $\mathcal{C} := \{\mathcal{C}_k\}_{k=0}^{N-1}$ that achieves the corresponding steering maneuver. The control at time k is a tuple of the feedback and the feedforward term s.t. $\mathcal{C}_k = (K_k, v_k)$.

$$\min_{\Sigma_k, K_k, \mu_k, v_k} J_{\text{MAX_COVAR}} = -\lambda_{\min}(\Sigma_0) \quad (\text{MAXCOVAR})$$

such that for all $k = 0, 1, \dots, N-1$,

$$\begin{aligned} \Sigma_{k+1} &= A\Sigma_k A^\top + BK_k \Sigma_k A^\top + A\Sigma_k K_k^\top B^\top \\ &\quad + BK_k \Sigma_k K_k^\top B^\top + DD^\top, \\ \lambda_{\max}(\Sigma_N) &\leq \lambda_{\min}(\Sigma_{\mathcal{G}}), \end{aligned} \quad (13a)$$

$$\mu_{k+1} = A\mu_k + Bv_k, \quad (13b)$$

$$\mu_0 = \mu_i, \quad \mu_N = \mu_{\mathcal{G}} \quad (13c)$$

$$\Phi^{-1}(1 - \epsilon_x) \sqrt{\alpha_x^\top \Sigma_k \alpha_x} + \alpha_x^\top \mu_k - \beta_x \leq 0, \quad (13d)$$

$$\Phi^{-1}(1 - \epsilon_u) \sqrt{\alpha_u^\top K_k \Sigma_k K_k^\top \alpha_u} + \alpha_u^\top v_k - \beta_u \leq 0, \quad (13e)$$

where $\lambda_{\min}(\cdot)$ is the minimum eigenvalue operator. $\lambda_{\min}(A)$ is a concave function of the positive semidefinite matrix variable A , therefore MAXCOVAR is a convex minimization objective. The feasible region of the above optimization problem is non-convex and we use the similar convexification procedure as described in [14] to convert MAXCOVAR into a convex semidefinite program for the purpose of numerical experiments.

Notation: We define a predicate FEASIBLE $:=$ FEASIBLE(q, p, N) that returns a boolean TRUE or FALSE if there exists a feasible N -step control sequence such that the system of equations (13a)-(13e) defined for $(\mu_{\mathcal{I}}, \Sigma_{\mathcal{I}}) = (\mu_q, \Sigma_q)$, and $(\mu_{\mathcal{G}}, \Sigma_{\mathcal{G}}) = (\mu_p, \Sigma_p)$ is feasible. Also, we use the notation $q \xrightarrow[N]{\mathcal{C}} p$ to denote that the mean and covariance dynamics initialized at (μ_q, Σ_q) and driven by the N -step control sequence \mathcal{C} satisfy the state and control chance constraints (13d)-(13e) at all time-steps and the terminal goal reaching constraint (13a) corresponding to $(\mu_{\mathcal{G}}, \Sigma_{\mathcal{G}}) = (\mu_p, \Sigma_p)$.

Remark 1. (Reuse) Let \mathcal{C} be a N -step control sequence s.t. $\mathcal{I} \xrightarrow[N]{\mathcal{C}} \mathcal{G}$, then it follows that $\mathcal{I}^- \xrightarrow[N]{\mathcal{C}} \mathcal{G}$ for all \mathcal{I}^- s.t. $\mu_{\mathcal{I}^-} = \mu_{\mathcal{I}}$, and $\Sigma_{\mathcal{I}^-} \preceq \Sigma_{\mathcal{I}}$. In other words, a control sequence computed for the steering maneuver from \mathcal{I} to \mathcal{G} remains a feasible maneuver from \mathcal{I}^- to \mathcal{G} and thus could be reused.

The above directly follows from an analysis of equations (13a)-(13e) and we omit a formal proof.

Rationale behind MAXCOVAR: MAXCOVAR is based on the maximization of the minimum eigenvalue of Σ_{initial} for a given initial mean, and a desired target distribution. It is based on the observation that if $\mathcal{I} \xrightarrow[N]{\mathcal{C}} \mathcal{G}$, then $\mathcal{I}' \xrightarrow[N]{\mathcal{C}} \mathcal{G} \quad \forall \quad \mathcal{I}'$ s.t. $\mu_{\mathcal{I}'} = \mu_{\mathcal{I}}$ and $\Sigma_{\mathcal{I}'} \preceq \Sigma_{\mathcal{I}}$. Therefore, we aim to find a $(\Sigma_{\text{initial}}, \mathcal{C})$ such that the computed \mathcal{C} could be reused across largest possible number of initial distributions, i.e., find \mathcal{I} such that $\{\mathcal{I}' \mid \mathcal{I}' \prec \mathcal{I}\}$ is the largest. This leads to the maximization of the minimum eigenvalue of the initial covariance as a natural objective function for our search.

Significance of MAXCOVAR: MAXCOVAR provides a certificate of reachability for any goal distribution in terms of the maximum permissible value of the minimum eigenvalue of the covariance at any query mean for which there exists a feasible control sequence that can achieve the corresponding steering maneuver under control constraints. For instance,

Algorithm 1 Constructing the MAXCOVAR BRT**Input:** \mathcal{G} , N , n_{iter} **Output:** \mathcal{T}

```

1:  $\nu \leftarrow \phi$ ,  $\varepsilon \leftarrow \phi$ 
2:  $\nu_0 \leftarrow \text{CREATENODE}(\mu_{\mathcal{G}}, \Sigma_{\mathcal{G}}, \text{NONE}, \text{NONE}, \{\})$ 
3:  $\nu \leftarrow \nu \cup \{\nu_0\}$ 
4: for  $i \leftarrow 1$  to  $n_{\text{iter}}$  do
5:    $\nu_k \leftarrow \text{RAND}(\nu)$ 
6:    $\mu_q \leftarrow \text{RANDMEANAROUND}(\nu_k, r_{\text{sample}})$ 
7:    $\text{status}, \Sigma_{\text{max}}, \mathcal{C}_q \leftarrow \text{MAXCOVAR}(\mu_q, (\mu_k, \Sigma_k), N)$ 
8:   if  $\text{status} \neq \text{infeasible}$  then
9:      $\text{idx} \leftarrow \text{size}(\nu) + 1$ 
10:     $\text{ch}_{\text{idx}} \leftarrow \{\}$ 
11:     $\nu_{\text{new}} \leftarrow \text{CREATENODE}(\mu_q, \Sigma_{\text{max}}, \text{idx}, k, \text{ch}_{\text{idx}})$ 
12:     $\nu \leftarrow \nu \cup \{\nu_{\text{new}}\}$ 
13:     $\text{ch}_k \leftarrow \text{ch}_k \cup \{\text{idx}\}$ 
14:     $\varepsilon_{\text{idx}, k} \leftarrow \mathcal{C}_q$ 
15:     $\varepsilon \leftarrow \varepsilon \cup \{\varepsilon_{\text{idx}, k}\}$ 
16:  $\mathcal{T} \leftarrow \{\nu, \varepsilon\}$ 
17: Return  $\mathcal{T}$ 

```

consider a goal distribution $(\mu_{\mathcal{G}}, \Sigma_{\mathcal{G}})$ and a query mean μ_q , and let $\Sigma_{q, \text{max}}, \mathcal{C}_{q, \text{max}}$ be such that,

$$\Sigma_{q, \text{max}}, \mathcal{C}_{q, \text{max}} \leftarrow \text{MAXCOVAR}(\mu_q, (\mu_{\mathcal{G}}, \Sigma_{\mathcal{G}}), N).$$

It follows from the above that $\forall \Sigma \succ 0$ s.t. $\lambda_{\min}(\Sigma) > \lambda_{\min}(\Sigma_{q, \text{max}})$, $\nexists \mathcal{C}$ s.t. $(\mu_q, \Sigma) \xrightarrow[N]{\mathcal{C}} \mathcal{G}$ by definition of MAXCOVAR otherwise $\lambda_{\min}(\Sigma_{q, \text{max}})$ is not the maximum possible minimum eigenvalue of the initial covariance for the existence of a feasible path and we arrive at a contradiction. On the other hand, all matrices $\Sigma \succ 0$ such that their minimum eigenvalue is less than the minimum eigenvalue of the covariance matrix computed in the maximal sense i.e. $\lambda_{\min}(\Sigma) < \lambda_{\min}(\Sigma_{q, \text{max}})$ have lesser *coverage* than the maximal covariance matrix, a fact that is formalized later in Lemma 2.

B. Construction of the MAXCOVAR BRT

The algorithm proceeds by building a tree $\mathcal{T}(\mathcal{G})$ represented through a set of nodes $\{\nu_i\}$, and a set of edge controllers $\{\varepsilon_{i,j}\}$. Each node i , ν_i , is a tuple $(\mu_i, \Sigma_i, p_i, \mathcal{C}_i, \text{ch}_i)$ where μ_i, Σ_i are the mean and the covariance of the distribution stored in the node, p_i is a pointer to the parent node, $\mathcal{C}_i := \{K_t^{i, p_i}, v_t^{i, p_i}\}_{t=0}^{N-1}$ is the N -step control sequence stored at the node that steers the state distribution from (μ_i, Σ_i) to $(\mu_{p_i}, \Sigma_{p_i})$, and ch_i is the list of pointers of all the children node of node i in the tree $\mathcal{T}(\mathcal{G})$.

ε is another data structure that stores all the edge information for the tree $\mathcal{T}(\mathcal{G})$, such that, $\varepsilon_{i,j} := \{K_t^{i,j}, v_t^{i,j}\}_{t=0}^{N-1}$ stores the N -step control sequence that steers the state distribution from node i to node j if such an edge exists, and is empty otherwise.

Now, we discuss the essential sub-routines of the above algorithm.

1) *Node selection*: The tree is grown in the spirit of finding paths from all query initial distributions for which paths would exist to the goal distribution. In our implementation, the nodes are selected randomly according to the Voronoi bias (of the first order moment of the nodes) to bias population of the 1-BRS of the node distributions whose corresponding Voronoi regions are relatively unexplored in the sense of the first order moment of the distributions.

2) *Node expansion*: Once a node to expand has been selected, a query mean is sampled from a neighbourhood of some radius around it and a connection is attempted through the MAXCOVAR method for edge construction. Let's say the k th node on the tree containing the distribution $(\mu^{(k)}, \Sigma^{(k)})$ has been selected to expand, and let μ_q be the query mean sampled from a neighbourhood around $\mu^{(k)}$ through the $\text{RANDMEANAROUND}(\cdot, r)$ module where r is some sampling radius. We solve the following optimization to construct the edge,

$$\Sigma_{\text{max}}, \mathcal{C}_{\text{max}} \leftarrow \text{MAXCOVAR}(\mu_q, (\mu^{(k)}, \Sigma^{(k)}), N) \quad (14)$$

$(\mu_q, \Sigma_{\text{max}})$ is added as a node to the tree with the edge controller \mathcal{C}_{max} and $(\mu^{(k)}, \Sigma^{(k)})$ as the parent if the status of the above optimization problem (as returned by the solver) is not infeasible.

Definition of h-BRS: We now define the following mathematical objects that will aid the analysis and further discussion of our proposed approach.

The h hop backward reachable set of distributions for a distribution p , $h\text{-BRS}(p)$ is defined as follows,

$$h\text{-BRS}(p) = \{(\mu_q, \Sigma_q) \mid \text{FEASIBLE}(q, p, hN) = \text{TRUE}\}. \quad (15)$$

We also define the h -BRS of a tree of distributions \mathcal{T} as,

$$h\text{-BRS}(\mathcal{T}) = \bigcup_{i \in \nu(\mathcal{T})} (h - d_i)\text{-BRS}(\nu_i), \quad (16)$$

where $\nu(\mathcal{T})$ is the set of all vertices in the tree, and d_i is the distance of the i -th node from the root node in terms of the number of hops.

Concatenation of control sequences: A concatenation $\mathcal{C}_{A,B}$ of two control sequences $\mathcal{C}_A, \mathcal{C}_B$ of lengths N_A, N_B respectively is a control sequence of length $N_A + N_B$ represented through the \bigcup operator i.e. $\mathcal{C}_{A,B} := \mathcal{C}_A \bigcup \mathcal{C}_B$, s.t. $\mathcal{C}_{A,B}(t) := \mathcal{C}_A(t) \forall t = 0, 1, \dots, N_A - 1$, and $\mathcal{C}_{A,B}(t) := \mathcal{C}_B(t - N_A) \forall t = N_A, N_A + 1, \dots, N_A + N_B - 1$. Note that the concatenation operator \bigcup is *non-commutative* in the two argument control sequences, i.e. $\mathcal{C}_A \bigcup \mathcal{C}_B \neq \mathcal{C}_B \bigcup \mathcal{C}_A$.

To establish / guarantee feasibility of the query distribution to the goal distribution, it is sufficient to guarantee feasibility to any distribution that is already verified to reach the goal. This idea can be seen in the following lemma on sequential composition of control sequences ensuring satisfaction of state and control chance constraints along the overall concatenated trajectory (see Fig. 1),

Lemma 1. (Recursive Feasibility through Sequential Composition) Let $(\mu_i, \Sigma_i), (\mu_j, \Sigma_j), (\mu_k, \Sigma_k), \mathcal{C}_{i,j}, \mathcal{C}_{j,k}, h_{i,j}, h_{j,k}$ be such that $(\mu_i, \Sigma_i) \xrightarrow[h_{i,j}N]{\mathcal{C}_{i,j}} (\mu_j, \Sigma_j)$, and $(\mu_j, \Sigma_j) \xrightarrow[h_{j,k}N]{\mathcal{C}_{j,k}} (\mu_k, \Sigma_k)$. It follows that $(\mu_i, \Sigma_i) \xrightarrow[(h_{i,j}+h_{j,k})N]{\mathcal{C}_{i,k}} (\mu_k, \Sigma_k)$ where $\mathcal{C}_{i,k} := \mathcal{C}_{i,j} \cup \mathcal{C}_{j,k}$.

Proof: We want to prove that the system initialized at the distribution (μ_i, Σ_i) and driven by the control sequence $\mathcal{C}_{i,k}$ which is obtained from the concatenation of the two control sequences $\mathcal{C}_{i,j}$, and $\mathcal{C}_{j,k}$ i.e. $\mathcal{C}_{i,k} = \mathcal{C}_{i,j} \cup \mathcal{C}_{j,k}$ satisfies all the state and control chance constraints of the form (13d)-(13e) for a trajectory of length $(h_{i,j} + h_{j,k})N$, and the terminal goal reaching constraints of the form (13c) and (13a) for the goal distribution (μ_k, Σ_k) .

The mean and covariance dynamics at any time t as a function of the initial distribution and the control sequence are represented as $\mu_t(\mu_i, \mathcal{C}_{i,k})$ and $\Sigma_t(\Sigma_i, \mathcal{C}_{i,k})$ respectively.

For $t = 0, 1, \dots, h_{i,j}N - 1$, $\mu_t(\mu_i, \mathcal{C}_{i,k}) = \mu_t(\mu_i, \mathcal{C}_{i,j})$, and $\Sigma_t(\Sigma_i, \mathcal{C}_{i,k}) = \Sigma_t(\Sigma_i, \mathcal{C}_{i,j})$. Since $\mathcal{C}_{i,j}$ is a feasible control sequence for the (i, j) maneuver s.t. $(\mu_i, \Sigma_i) \xrightarrow[N_{i,j}]{\mathcal{C}_{i,j}} (\mu_j, \Sigma_j)$, all the state and control chance constraints are satisfied by $\mu_t(\mu_i, \mathcal{C}_{i,j})$ and $\Sigma_t(\Sigma_i, \mathcal{C}_{i,j})$ for $t = 0, 1, \dots, N_{i,j} - 1$, and $\Sigma_{N_{i,j}}(\Sigma_i, \mathcal{C}_{i,k})$ is s.t. $\lambda_{\max}(\Sigma_{N_{i,j}}(\Sigma_i, \mathcal{C}_{i,k})) \leq \lambda_{\min}(\Sigma_j)$ which implies $\Sigma_{N_{i,j}}(\Sigma_i, \mathcal{C}_{i,k}) \preceq \Sigma_j$.

The rest of the maneuver for $t = N_{i,j}, N_{i,j} + 1, \dots, N_{i,j} + N_{j,k} - 1$ could be thought of as an $N_{j,k}$ step maneuver initialized at $\mu_{N_{i,j}}$ and $\Sigma_{N_{i,j}}$. $\mathcal{C}_{j,k}$ is s.t. $(\mu_j, \Sigma_j) \xrightarrow[N_{j,k}]{\mathcal{C}_{j,k}} (\mu_k, \Sigma_k)$, therefore from Remark 1, $(\mu_j, \Sigma_{N_{i,j}}(\Sigma_i, \mathcal{C}_{i,k})) \xrightarrow[N_{j,k}]{\mathcal{C}_{j,k}} (\mu_k, \Sigma_k)$ since $\Sigma_{N_{i,j}}(\Sigma_i, \mathcal{C}_{i,k}) \preceq \Sigma_j$. ■

C. Planning through the BRT

In this section, we discuss our approach to find feasible paths to the goal through a BRT.

Finding a feasible path: To find a feasible path to the goal for a query distribution $q := (\mu_q, \Sigma_q)$, single hop connections are attempted one-by-one to M nearest nodes on the BRT for some hyperparameter M . For a candidate node ν_k on the BRT, the following problem is solved,

$$\mathcal{C}_q \leftarrow \text{OPTSTEER}((\mu_q, \Sigma_q), (\mu^{(k)}, \Sigma^{(k)}), N). \quad (17)$$

The search for a feasible path terminates once a connection has successfully been established to one of the existing nodes on the BRT, and is given by a concatenation of the above computed control sequence \mathcal{C}_q with the pre-computed controllers stored in the sequence of edges of the tree from the k th node to the root node. Let ν_k be a distance of d_k hops away from the goal s.t. $\text{idx}_0, \text{idx}_1, \dots, \text{idx}_{d_k}$ be the sequence of nodes encountered from the k th node to the root node where $\text{idx}_0 = k$ and $\text{idx}_{d_k} = 0$. Therefore, the feasible path from q to the goal \mathcal{G} is obtained as,

$$\mathcal{C}_{q,\mathcal{G}} = \mathcal{C}_q \cup \mathcal{C}_{\text{idx}_0, \text{idx}_1} \cup \mathcal{C}_{\text{idx}_1, \text{idx}_2} \dots \cup \mathcal{C}_{\text{idx}_{d_k-1}, \text{idx}_{d_k}} \text{ s.t. } q \xrightarrow[(d_k+1)N]{\mathcal{C}_{q,\mathcal{G}}} \mathcal{G}, \text{ which follows from Lemma 1.}$$

Implication of recursive feasibility on the speed-up in computing a feasible path: Say that the query distribution q is such that $q \in \mathfrak{t} - \text{BRS}(\mathcal{G})$ and $q \notin \mathfrak{t}' - \text{BRS}(\mathcal{G}) \forall t' < t$, i.e. a path from q to the goal shorter than t -hops does not exist. Therefore, to compute a feasible path that steers the system from q to \mathcal{G} without reusing any of the pre-computed controllers from the BRT, a feasibility instance $\text{FEASIBLE}(q, \mathcal{G}, tN)$ of size tN needs to be solved. Alternatively, if the search for a feasible path is carried through attempting connections to the BRT, the expenditure on compute is that of solving a feasibility instance of size N a maximum M number of times resulting in an order of magnitude savings in computation. More details about the empirical experiments and results can be found in Section V.

Maximum Coverage of the MAXCOVAR BRT: As discussed earlier, re-use of controllers stored along the edges of a BRT can lead to significant speedup in the computation of a feasible path to the goal. Therefore, it is a desirable property for a BRT to be such that paths from as large a number of query initial distributions as possible can be found to the tree. This property is characterized in terms of the *coverage* of the BRT, and for any given BRT \mathcal{T} , its coverage is quantified through the set of all distributions that can reach the tree in h hops i.e.

$$\text{Cover}(\mathcal{T}) := h - \text{BRS}(\mathcal{T}). \quad (18)$$

We now show that the BRT constructed through the novel objective function for edge construction defined in Section IV-A henceforth referred to as MAXCOVAR BRT discovers feasible paths from provably the largest possible set of query initial distributions compared to any other procedure of edge construction. This is formalized as Theorem 1 below. We proceed by first proving Lemma 2 that talks about the coverage of a distribution (single node) and is used as a building block for Theorem 1 that concerns the coverage of a tree (multiple nodes).

For the statement of the lemma and the theorem, we recall that a planning scene $\{(\alpha_x, \beta_x, \epsilon_x), (\alpha_u, \beta_u, \epsilon_u)\}$ refers to the collection of all parameters that define chance constraints on the state and control input.

Lemma 2. For $\Sigma_{\max}, \Sigma_{\max}^- \succ 0$, s.t. $\lambda_{\min}(\Sigma_{\max}) > \lambda_{\min}(\Sigma_{\max}^-)$, $h - \text{BRS}(\mu, \Sigma_{\max}) \geq h - \text{BRS}(\mu, \Sigma_{\max}^-) \forall h \geq 1$ for all planning scenes. Also, there exist planning scenes $\{(\alpha_x, \beta_x, \epsilon_x), (\alpha_u, \beta_u, \epsilon_u)\}$ s.t. $h - \text{BRS}(\mu, \Sigma_{\max}) > h - \text{BRS}(\mu, \Sigma_{\max}^-) \forall h \geq 1$.

Proof: Proved in [15]. ■

Before stating Theorem 1, we introduce the following notation. Let $\mathcal{T}_{\text{MAXCOVAR}}^{(r)}$ and $\mathcal{T}_{\text{ANY}}^{(r)}$ be the state of the BRTs at the r th iteration of the node addition procedure following the MAXCOVAR and ANY methods of node and edge construction respectively, initialized at the common tree state of a singleton node of the goal distribution. The ANY

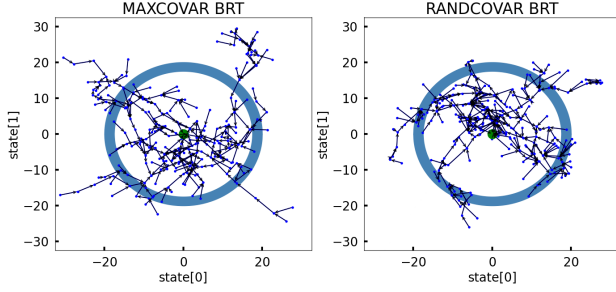


Fig. 2. Coverage experiment setup: MAXCOVAR and RANDCOVAR BRTs

procedure is any method of node and edge addition that does not explicitly maximize the minimum eigenvalue of the covariance corresponding to any query mean μ_q and terminal distribution (μ, Σ) . In other words, ANY refers to any algorithm that returns a node covariance and edge controller tuple $(\Sigma_q, \mathcal{C}_q)$ s.t. $\lambda_{\min}(\Sigma_q) < \lambda_{\min}(\Sigma_{q,\max})$ where $\Sigma_{q,\max}, \mathcal{C}_{q,\max} \leftarrow \text{MAXCOVAR}(\mu_q, (\mu, \Sigma), N)$. ANY could for instance sample the covariance randomly, or add edges that optimize for a regularized objective of the minimum eigenvalue with a performance term.

Theorem 1 (Maximum Coverage). *$h\text{-BRS}(\mathcal{T}_{\text{MAXCOVAR}}^{(r)}) \supseteq h\text{-BRS}(\mathcal{T}_{\text{ANY}}^{(r)})$ for all planning scenes, and there always exist planning scenes such that $h\text{-BRS}(\mathcal{T}_{\text{MAXCOVAR}}^{(r)}) \supset h\text{-BRS}(\mathcal{T}_{\text{ANY}}^{(r)}) \forall r \geq 1, \forall h \geq 1$.*

Proof: Proved in [15]. ■

V. EXPERIMENTS

To illustrate our method, we conduct experiments for the motion planning of a quadrotor in a 2D plane. The lateral and longitudinal dynamics of the quadrotor are modeled as a triple integrator leading to a 6 DoF model with state matrices,

$$A = \begin{bmatrix} I_2 & \Delta T I_2 & 0_2 \\ 0_2 & I_2 & \Delta T I_2 \\ 0_2 & 0_2 & I_2 \end{bmatrix}, B = \begin{bmatrix} 0_2 \\ 0_2 \\ \Delta T I_2 \end{bmatrix}, D = 0.1 I_6,$$

a time step of 0.1 seconds, a horizon of $N = 20$, and a goal distribution \mathcal{G} for the planning task as follows:

$$\mu_{\mathcal{G}} = \mathbf{0}_{6 \times 1}, \quad \Sigma_{\mathcal{G}} = 0.1 * \mathbf{I}_{6 \times 6}.$$

The control input space is characterized by a bounding box represented as $\alpha_u = \{[\pm 1, 0]^\top, [0, \pm 1]^\top\}$, $\beta_u = \{\pm 25, \pm 25\}$. The chance constraint linearization is performed around $\Sigma_r = 1.2 \mathbf{I}_{6 \times 6}$, and $Y_r = 15 \mathbf{I}_{2 \times 2}$. All the optimization programs are solved in Python using cvxpy [16].

Construction of the BRTs: For the tree construction procedure, a sampling radius of $r_{\text{sample}} = [\pm 5, \pm 5, \pm 2.5, \pm 2.5, \pm 1.25, \pm 1.25]$ was used. Fig. 2 shows the constructed MAXCOVAR and the RANDCOVAR trees corresponding to the goal distribution \mathcal{G} that were used for the experiments on coverage. The figure displays the first two states of the 6 DoF model: the

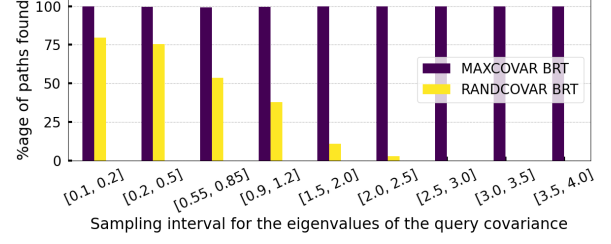


Fig. 3. Coverage experiment metrics

x (state[0]) and y (state[1]) locations of the quadrotor on the x-y plane. Each node on the plot denotes the first two dimensions of the state distribution mean, and the directed edges between the nodes are the corresponding N -step control sequences that are stored offline. The blue annulus around the two trees in Figure 2 represents the region over which the query means were uniformly sampled. Both the trees in Fig. 2 consist of 265 nodes and were generated with the same random seed.

The construction procedure for the MAXCOVAR tree was described in Algorithm 1. For any existing node ν_k on the tree that's selected to expand, and for any query mean μ_q sampled from a box around it, the new node covariance $\Sigma_{q,\max}$ and the corresponding edge controller $\mathcal{C}_{q,\max}$ is the result of solving the MAXCOVAR optimization problem.

For the construction of the RANDCOVAR tree, the node covariance $\Sigma_{q,\text{rand}}$ was randomly sampled from the space of positive definite matrices, similar to [1], and the edge controller was given by the optimal steering control from $(\mu_q, \Sigma_{q,\text{rand}})$ to (μ_k, Σ_k) . To construct samples from the positive definite matrix space, the eigenvalues and orthonormal eigenvectors that constitute a positive definite matrix are sampled separately. It was observed empirically that randomly sampling node covariances resulted in rejecting a lot of candidate nodes due to the non-existence of a corresponding steering maneuver. Therefore, eigenvalues of the candidate node covariance Σ_{rand} were sampled to ensure that $\lambda_{\min}(\Sigma_{q,\text{rand}}) \leq \lambda_{\min}(\Sigma_{q,\max})$.

Coverage Experiment: The coverage experiment proceeds by sampling query distributions and attempting connects to the two trees to find paths. Query means were sampled from the blue annulus as shown in Fig. 2, and the query covariances were considered to be diagonal matrices where the diagonal entries were sampled uniformly over an interval. The experiment was repeated for different intervals of sampling the diagonal entries of the query covariance as shown in the x-axis of Fig. 3. For each interval, the experiment was repeated 250 times and the percentage of times a path was found to the two trees was reported.

Interpretation of the Coverage Experiment: From Fig. 3, it can be seen that for intervals corresponding to higher candidate eigenvalues of the query covariance, the RANDCOVAR tree with randomly sampled node covariances is not able to find paths in contrast to MAXCOVAR tree where the

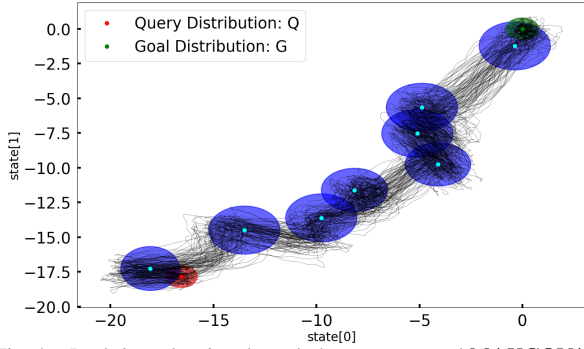


Fig. 4. Real-time planning through the pre-computed MAXCOVAR BRT

node covariances and the edges were constructed explicitly to provide maximal coverage. This is a direct consequence of Lemma 2 which says that for two positive definite matrices Σ and Σ^- s.t. $\lambda_{\min}(\Sigma^-) < \lambda_{\min}(\Sigma)$, there exist planning scenes such that distributions with a larger spectral radius of the covariance reach Σ as compared to Σ^- (Lemma 2 follows a proof by construction, see the Appendix of [15]).

Real-time planning through the MAXCOVAR BRT: The generated tree and the associated control sequences were stored offline. For real-time planning, a query mean μ_q was randomly sampled in a box around the origin with a sampling radius of $[\pm 25, \pm 25, \pm 25, \pm 25, 0, 0]$ with a query covariance of $\Sigma_q = 0.1 \mathbf{I}_{6 \times 6}$. Fig. 4 shows one such run that resulted in a 9-hop chained maneuver that drives the sampled query mean to the goal (a total of $9 \times N = 180$ timesteps). It took 1 second to find a feasible path through the offline generated BRT, and around 2 minutes to find a path through $\text{OPTSTEER}(q, \mathcal{G}, N)$ for $N = 180$. In Fig. 4, the red and green ellipses correspond to the 3-sigma confidence interval of the sampled query node and the goal node respectively. The blue ellipses correspond to the distributions stored in the nodes of the discovered feasible path. Monte carlo simulations were performed and the system trajectories are represented by black lines. The only real-time computation that was done was to find a control sequence that drives the system from the red ellipse to the first blue ellipse in the sequence, the rest of the trajectories were propagated using controllers stored along each edge of the discovered feasible path of the BRT.

ACKNOWLEDGMENTS

This work was supported by Lockheed Martin Corporation.

CONCLUSION

In this paper we introduced Maximal Covariance Backward Reachable Trees (MAXCOVAR BRT), a multi-query algorithm for probabilistic planning under constraints on the control input with explicit coverage guarantees. Our contribution was a novel optimization formulation to add nodes and construct the corresponding edge controllers such that the generated roadmap results in provably maximal coverage. The notion of coverage of a roadmap was characterized formally via h-BRS (Backward Reachable Set of Distributions) of a tree of distributions, and our proposed method

was supported by theoretical analysis as well as extensive simulation on a 6 DoF model.

REFERENCES

- [1] D. Zheng, J. Ridderhof, Z. Zhang, P. Tsotras, and A.-A. Aghamohammadi, "Cs-brm: A probabilistic roadmap for consistent belief space planning with reachability guarantees," *IEEE Transactions on Robotics*, 2024.
- [2] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [3] V. N. Hartmann, M. P. Strub, M. Toussaint, and J. D. Gammell, "Effort informed roadmaps (eirm*): Efficient asymptotically optimal multiquery planning by actively reusing validation effort," in *The International Symposium of Robotics Research*, pp. 555–571, Springer, 2022.
- [4] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.
- [5] A. Majumdar, M. Tobenkin, and R. Tedrake, "Multi-query feedback motion planning with lqr-roadmaps," 2011.
- [6] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.
- [7] M. K. M. Jaffar and M. Otte, "Pip-x: Online feedback motion planning/replanning in dynamic environments using invariant funnels," *The International Journal of Robotics Research*, p. 02783649231209340, 2023.
- [8] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, p. 8160, 2010.
- [9] B. D. Luders, *Robust sampling-based motion planning for autonomous vehicles in uncertain environments*. PhD thesis, Massachusetts Institute of Technology, 2014.
- [10] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.
- [11] D. Zheng, J. Ridderhof, Z. Zhang, P. Tsotras, and A.-A. Aghamohammadi, "Cs-brm: A probabilistic roadmap for consistent belief space planning with reachability guarantees," *IEEE Transactions on Robotics*, 2024.
- [12] K. Okamoto, M. Goldshtein, and P. Tsotras, "Optimal covariance control for stochastic systems under chance constraints," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 266–271, 2018.
- [13] F. Liu, G. Rapakoulias, and P. Tsotras, "Optimal covariance steering for discrete-time linear stochastic systems," *arXiv preprint arXiv:2211.00618*, 2022.
- [14] G. Rapakoulias and P. Tsotras, "Discrete-time optimal covariance steering via semidefinite programming," *arXiv e-prints*, pp. arXiv–2302, 2023.
- [15] N. Aggarwal and J. P. How, "SDP Synthesis of Distributionally Robust Backward Reachable Trees for Probabilistic Planning," *submitted to IEEE Transactions on Automatic Control*.
- [16] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.