Submitted to the IEEE In't Conference on Robotics and Automation (ICRA), 2023.

Chance-Constrained Motion Planning with Event-Triggered Estimation

Anne Theurkauf, Qi Heng Ho, Roland Ilyes, Nisar Ahmed, and Morteza Lahijanian

***Abstract*— We consider the problem of autonomous navigation using limited information from a remote sensor network. Because the remote sensors are power and bandwidth limited, we use event-triggered (ET) estimation to manage communication costs. We introduce a fast and efficient sampling-based planner which computes motion plans coupled with ET communication strategies that minimize communication costs, while satisfying constraints on the probability of reaching the goal region and the point-wise probability of collision. We derive a novel method for offline propagation of the expected state distribution, and corresponding bounds on this distribution. These bounds are used to evaluate the chance constraints in the algorithm. Case studies establish the validity of our approach, demonstrating fast computation of optimal plans.**

I. INTRODUCTION

As robots become more capable, they also become more adept at autonomously exploring remote environments, especially those that are hostile to humans. Examples include deep ocean, planetary, and subterranean exploration. Because of inaccessibility and limited resources, these robots must be able to operate *efficiently* and *safely*. However, these are often *competing* objectives, i.e., safer performance requires more resources. This problem is exacerbated when robots are part of a network: frequent information communication increases the probability of safely completing the mission, but can also lead to unacceptable resource consumption. This work focuses on this problem and aims to develop a framework for robot motion and communication planning that guarantees safety while minimizing resource cost.

Consider the case of a robot in a lunar exploration scenario with a network of remote sensors. The robot is tasked to navigate to a scientifically interesting location and avoid craters and boulders. The robot relies on the sensor network for localization, but the remote sensors have limited battery life, and transmitting information is costly (e.g., transmission is an order of magnitude more costly than any other function on a Zigbee networking chip [1]). The network must also limit communication due to bandwidth constraints. The robot must find a motion and communication plan that minimizes the resource cost on the sensors while guaranteeing safety.

A ubiquitous technique to manage communication cost in power or bandwidth limited systems is event-triggered (ET) estimation [2]–[5], where communication is limited to only situations where the information is deemed useful. This typically takes the form of a threshold on some useful quantity, like the Kalman Filter (KF) innovation [3]–[5]. ET allows the user to lower resource consumption by changing this threshold, thus reducing the volume of transmitted information and trading off estimation accuracy with resource use. Ref. [6] introduces a method that guarantees the trade-off is optimal by accounting for both task performance and resources. While providing a very comprehensive Pareto Front solution, the methods do not include any path or motion planning, and require heavy computation power, resulting in lengthy computation times and precluding onboard implementation.

Sampling-based motion planners are well established tools that can rapidly find solutions to complex problems [7]–[11]. The first developed planners deal with deterministic transitions and operate solely in the state space [8], [12]. Later formulations extend these deterministic planners to obtain asymptotically-optimal sampling-based planners such as SST and SST* [11], and incorporate motion uncertainty via chance constraints on the probability of collision (CC-RRT), [9], [10], [13]. These techniques have more recently been extended to accommodate measurement uncertainty with Gaussian belief trees [14], [15], and feedback-based information roadmaps [16]. However, while these fast and efficient methods have been extended to accommodate other uncertainties, no sampling-based motion planning algorithm currently exists that incorporate ET estimation.

In this paper, we develop a sampling-based algorithm for *motion and ET-communication* (METC) planning that, for a given system and environment, generates both a motion and ET estimation threshold plan. This algorithm is fast and efficient, and provides guarantees on the point-wise probability of collision and reaching the goal (i.e., safety constraints), while minimizing communication cost. We first derive an offline method to predict the expected state distribution, and provide the corresponding bounds for this distribution. Using these results, we then derive a sampling-based algorithm that generates the METC plans. We show the efficacy of our methods in several case studies. The results show that we can successfully generate plans for a variety of environments, these plans are valid with respect to the safety constraint, and optimality increases with increased computation time. We additionally compare the results of our algorithm to a Pareto optimal strategy generated in [6], showing that we can quickly generate plans with similar performance.

In summary, our contributions are: (i) a derivation of propagation equations for the distribution across states under ET estimation, (ii) a derivation of the bounds for these distributions, (iii) an algorithm for generating METC plans that minimize communication costs while respecting a safety constraint, and (iv) a series of case studies and benchmarks, demonstrating the algorithm's efficiency and optimality.

Authors are with the department of Aerospace Engineering Sciences at the University of Colorado Boulder, CO, USA {*firstname.lastname*}@colorado.edu

II. PROBLEM FORMULATION

We consider a scenario where a lunar robot must navigate to a scientifically interesting goal region while avoiding obstacles. The robot receives measurements from remote sensors, which are resource and bandwidth constrained. The goal is to generate METC plans that respect a safety constraint on obstacle collision and reaching the goal region, while simultaneously minimizing communication cost. We achieve this through a belief-based motion planner.

A. Robot Motion and Remote Sensor Models

The motion of the robot is uncertain and described by:

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad w_k \sim \mathcal{N}(0, Q), \tag{1}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, $u_k \in \mathcal{U} \subseteq \mathbb{R}^p$ is the control, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$, and $w_k \in \mathbb{R}^n$ is a random variable that represents a zero-mean Gaussian distributed noise with covariance $Q \in \mathbb{R}^{n\times n}$.

The robot receives measurements from a remote sensor network, for example beacons that provide one-way ranging measurements. We assume that the robot communicates with the closest sensor, and therefore only fuses at most one measurement. For ease of presentation, we assume that the remote sensors are identical, but we emphasize that extending to different models is trivial. Hence, the sensor network can be represented as a single sensor with model:

$$y_k = Cx_k + v_k, \quad v_k \sim \mathcal{N}(0, R), \tag{2}$$

where $y_k \in \mathbb{R}^m$ is the measurement, $C \in \mathbb{R}^{m\times n}$, and $v_k \in \mathbb{R}^m$ is a random variable that represents zero-mean Gaussian distributed sensor noise with covariance $R \in \mathbb{R}^{m\times m}$.

The initial state of the robot is described by a Gaussian distribution $x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_0)$, with mean $\hat{x}_0 \in \mathbb{R}^n$ and covariance $\Sigma_0 \in \mathbb{R}^{n\times n}$. We assume the robot is fully controllable and observable, and that the covariances Q and R are positive definite.

B. Event-triggered Estimation

Due to process and measurement noise, the robot's true state is unknown. Hence, an estimator maintains a probability distribution over the states, $x_k \sim b(x_k)$, called the *belief*. Note that this is an online estimate conditioned on observed measurements. To conserve communication costs, the system operates with the KF innovation-based ET estimator presented in [5], where measurements are only communicated when they are "surprising". That work derives a recursive minimum mean square error (MMSE) estimator under the assumption that the belief is Gaussian (see Sec. III-B for details). For a given a threshold $\delta_k \in \Delta = \mathbb{R}_{>0}$, measurement y_k is *surprising* if the norm of the Mahalanobis (whitening) transformation of $z_k = y_k - C\hat{x}_k$ is larger than threshold δ_k.

We define $\gamma_k \in \{0, 1\}$ to be the triggering indicator, i.e., $\gamma_k = 1$ if the measurement is sent, and $\gamma_k = 0$ otherwise. When $\gamma_k = 0$, the robot is *implicitly* informed that y_k is not surprising; the key advantage of ET filter is the exploitation of this information to improve state estimation. In this framework, the threshold δ_k is a design parameter, which trades off estimation accuracy with resource cost. We seek a method for optimally setting this threshold.

C. Controller

The robot is equipped with a trajectory following controller. Given a nominal trajectory as a sequence of nominal control inputs $\check{U}_0^T = (\check{u}_0, \ldots, \check{u}_T)$ and nominal states $\check{X}_0^T = (\check{x}_0, \ldots, \check{x}_T)$ the feedback controller is: $u_k = \check{u}_{k-1} - K(\hat{x}_k - \check{x}_k)$, where K is the controller gain. Under this controller, the closed-loop system dynamics become: $x_k = Ax_{k-1} + B(\check{u}_{k-1} - K(\hat{x}_k - \check{x}_k)) + w_k$. The goal of this work is to compute the nominal trajectory $(\check{U}_0^T, \check{X}_0^T)$ along with the sequence of ET estimation thresholds $\Delta_0^T = (\delta_0, ..., \delta_T)$ that satisfy the mission objectives and constraints described below. We define a *motion and ET-communication (METC) plan* to be $(\check{U}_0^T, \check{X}_0^T, \Delta_0^T)$.

D. Mission Objectives

The mission consists of three objectives: respect constraints on the probability of reaching the goal and avoiding obstacles, and minimize resource consumption. Here, we formalize these objectives.

The environment contains a set of obstacles, $\mathcal{X}_{obs} \subset \mathbb{R}^n$, and a goal region, $\mathcal{X}_{goal} \subset \mathbb{R}^n$. The probability of collision at time step k is defined as:

$$P(x_k \in \mathcal{X}_{obs}) = \int_{\mathcal{X}_{obs}} b(x_k)(s)ds, \tag{3}$$

where $b(x_k)(s)$ is the distribution $b(x_k)$ evaluated at state $s \in \mathcal{X}$. The probability of terminating in the goal region is:

$$P(x_T \in \mathcal{X}_{goal}) = \int_{\mathcal{X}_{goal}} b(x_T)(s)ds, \tag{4}$$

where x_T is the terminal point on a trajectory.

The third objective is to conserve resources for the remote sensors. The cost of transmitting a single measurement at time step k is $c_m\gamma_k$. Since trigger γ_k depends on the triggering threshold δ_k, the *total communication cost* for a trajectory with T time steps and $\Delta_0^T = (\delta_0, \ldots, \delta_T)$ is $\mathcal{J}_T(\Delta_0^T) = \sum_{k=1}^{T} c_m\gamma_k$. Because we are reasoning in belief space, the cost is considered in expectation:

$$J_T(\Delta_0^T) = \mathbb{E}[\mathcal{J}_T(\Delta_0^T)] = \sum_{k=1}^{T} c_m\mathbb{E}[\gamma_k] = \sum_{k=1}^{T} c_m\Gamma(\delta_k) \tag{5}$$

where $\Gamma(\delta_k)$ is the expected triggering rate for δ_k. Because we focus on sensor resources, (5) contains only communication cost, but can be easily extended to include terms for path length and control effort.

E. Problem Statement

Given a robot with dynamics in (1), sensor network with measurement model in (2), set of obstacles $\mathcal{X}_{obs}$, goal region $\mathcal{X}_{goal}$, and safety probability bound $p_{\text{safe}} \in [0, 1]$, compute an optimal METC plan $(\check{U}_0^T, \check{X}_0^T, \Delta_0^T)^*$ that minimizes the expected total communication cost, i.e.,

$$(\check{U}_0^T, \check{X}_0^T, \Delta_0^T)^* = \arg \min_{\check{u}_k, \check{x}_k, \delta_k, T} J_T(\Delta_0^T) \tag{6}$$

subject to the following constraints on the point-wise probability of collision and reaching the goal:

$$P(x_k \in \mathcal{X}_{obs}) < 1 - p_{\text{safe}}, \quad \forall k \in [0, T] \tag{7}$$
$$P(x_T \in \mathcal{X}_{goal}) > p_{\text{safe}}. \tag{8}$$

The key challenge in addressing this problem is to account for the uncertainty introduced by ET estimation, which requires forecasting state distributions over both unknown measurements and unknown triggers. Measurement uncertainty in a KF is accounted for in [15]; however, there are no existing methods to forecast uncertainty over a triggering condition. Straightforward attempts to extending [15] to account for triggering quickly run into problems of interdependency between variables and exponentially exploding belief trees. We address these challenges by developing a method to propagate bounds on the state distributions under a given choice of δ_k and using these to over-approximate estimates to check the safety constraint. We then develop a planning algorithm by integrating these methods with the sampling-based algorithm in [14] to generate METC plans that asymptotically minimize communication costs while respecting safety constraints.

III. PRELIMINARIES

A. Belief Prediction under Kalman Filter

For a linearizable and controllable system operating under a standard Kalman Filter, [15] presents a method to forecast the belief over the state while accounting for the fact that the measurements are unknown random variables *a priori*. This *expected belief*, $\mathbf{b}(x_k)$, is defined with respect to all possible measurements as:

$$\mathbf{b}(x_k) = \mathbb{E}_Y[b(x_k \mid x_0, y_{0:k})] = \int_{y_{0:k}} b(x_k \mid x_0, y_{0:k}) pr(y_{0:k}) dy$$

This forecast enables the evaluation of the chance constraints.

For a given nominal trajectory, $\check{X}_0^T = (\check{x}_0, \ldots, \check{x}_T)$, the expected belief $\mathbf{b}(x_k) = \mathcal{N}(\check{x}_k, \Sigma_k + \Lambda_k)$, can be recursively calculated from an initial belief $b(x_0)$ using the belief propagation method from [15]:

$$\Sigma_k^- = A\Sigma_{k-1}A^T + Q, \quad \Sigma_k = \Sigma_k^- - L_k C \Sigma_k^-, \tag{9}$$
$$\Lambda_k = (A - BK)\Lambda_{k-1}(A - BK)^T + L_k C \Sigma_k^-, \tag{10}$$

where Σ_k is the online uncertainty given by the KF, and Λ_k is covariance of the forecasted state estimates $\hat{x}_k$ (note that $\hat{x}_k$ is a random variable offline). Intuitively, this distribution can be thought of as the sum of the online estimation error and the forecasted uncertainty from not-yet-known measurements that the system receives during execution. In this work, we develop a method of forecasting belief under an ET Filter.

B. MMSE Filter for Event-triggered Estimation

For state estimation, we use the triggering scheme described in [5], which is based on the KF innovation. Recall that the trigger, γ_k, depends on the triggering threshold δ_k, and that $\gamma_k = 1$ indicates that measurement y_k is sent, and $\gamma_k = 0$ otherwise. According to the MMSE estimator in [5], the estimate of the state is Gaussian with *a priori* update:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k, \quad \Sigma_k^- = A\Sigma_{k-1}A^T + Q, \tag{11}$$

and *a posteriori* update given by:

$$\hat{x}_k = \hat{x}_k^- + \gamma_k L_k z_k, \quad z_k = y - C\hat{x}_k^- \tag{12}$$
$$\Sigma_k = \Sigma_k^- - \Big[\gamma_k + (1-\gamma_k)\beta(\delta_k)\Big] L_k C \Sigma_k^-. \tag{13}$$

where $L_k = \Sigma_k^- C^T [C\Sigma_k^- C^T + R]^{-1}$ is the KF gain. The term $\beta(\delta_k)$ is a scalar multiplier that effectively attenuates the KF gain in the covariance update as a function of δ_k, and is given by

$$\beta(\delta) = \frac{2}{\sqrt{2\pi}} \delta e^{-\frac{\delta^2}{2}} \left[1 - 2\mathcal{Q}(\delta)\right]^{-1}, \tag{14}$$

where

$$\mathcal{Q}(\delta) \triangleq \int_{\delta}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx. \tag{15}$$

The triggering condition takes the form:

$$\gamma_k = \begin{cases} 0 & \text{if } \|\epsilon_k\|_\infty \leq \delta_k \\ 1 & \text{otherwise,} \end{cases} \tag{16}$$

where ϵ_k is the Mahalanobis (whitening) transformation of the KF innovation, z_k. Note that, because of the whitening transformation, ϵ_k is always distributed as a standard normal. Hence, the expected value of γ_k is solely dependent on δ_k and is given by $\Gamma(\delta_k) = 1 - [1 - 2\mathcal{Q}(\delta_k)]^m$, where m is the dimension of the measurement vector.

IV. BELIEF PREDICTION UNDER ET FILTER

In this section, we present a novel method to predict the state distribution offline under an ET filter. The resulting distribution can be used to check the validity of the safety constraints. Based on this method, we devise the planning algorithm in Sec. V.

A key requirement for the method of belief prediction in Sec. III-A is that the distribution over state estimates, $\hat{x}_k$, is Gaussian, and defined by $\hat{x}_k \sim \mathcal{N}(\check{x}_k, \Lambda_k)$. When this distribution is known, it can be used to represent the joint distribution over x_k and $\hat{x}_k$, which can in turn be marginalized to obtain the expected distribution over states, $x_k \sim \mathbf{b}(x_k) = \mathcal{N}(\check{x}_k, \Sigma_k + \Lambda_k)$. This marginal is then used to evaluate the chance constraints in (7) and (8) offline.

Unfortunately, under ET estimation the covariance is updated under two randomly switching modes (according to $\gamma_k = 0$ or $\gamma_k = 1$), which means that the methods described in Sec. III.A cannot be directly applied to ET estimation. If the trigger γ_k is treated as an unknown random variable, then the expected belief must be taken with respect to all possible γ_k, and is not Gaussian. However, if γ_k is assumed to be given, then the distribution over $\hat{x}_k$ is Gaussian. This can be seen by examining the ET estimation update in (12): when γ_k is given, the conditional dependency of $\hat{x}_k$ on γ_k vanishes and the Gaussian property is preserved during the update. The mean and covariance can be obtained by evaluating $\mathbb{E}[\hat{x}_k] = \check{x}_k$ and $\mathbb{E}[(\hat{x}_k - \check{x}_k)(\hat{x}_k - \check{x}_k)^T] = \Lambda_k$ respectively.

The resulting distribution $x_k \sim \mathcal{N}(\check{x}_k, \Sigma_k + \Lambda_k)$ can then be calculated, with Σ_k given by the ET update in (12) and Λ_k given by:

$$\Lambda_k = (A - BK)\Lambda_{k-1}(A - BK)^T + \gamma_k L_k C \Sigma_k^- . \quad (17)$$

The assumption that γ_k is given is significant. Instead of taking the expected belief with respect to all possible γ_k, we assume a realization of some specific sequence of γ_k. However, if we are only concerned with bounding the probability of being within some region, this assumption can be easily accommodated. We simply need to determine the triggering condition that results in the expected belief with the highest, or lowest, probability of being in that region.

It is intuitive to consider the following discussion from the perspective of risk. In robotics, we generally consider higher uncertainty as producing higher risk, or higher probability of violating safety constraints. While this is not technically true for certain situations, e.g., a tight distribution in close proximity to a small obstacle, it is reasonable for many robotics problems. So, for our problem we seek to determine the largest possible covariance, which results in the highest probability of violating the safety constraint. If the safety constraint is satisfied for this worst case belief, then it must also be satisfied for all possible beliefs.

Consider the case for an arbitrary Gaussian belief $b(x)$ and some region $\mathcal{A} \subseteq \mathcal{X}$, where we require $P(x \in \mathcal{A}) < p_{\mathcal{A}}$. We make the assumption that if the covariance of $b(x)$ satisfies the constraint, then any Gaussian belief with smaller covariance also satisfies the constraint, formally:

Assumption 1. *If $x_a \sim b_a = \mathcal{N}(\mu, C_a)$, $x_b \sim b_b = \mathcal{N}(\mu, C_b)$, and $C_a > C_b$, then $P(x_a \in \mathcal{A}) < p_{\mathcal{A}} \implies P(x_b \in \mathcal{A}) < p_{\mathcal{A}}; \forall x \in \mathcal{X}$.*

Note that for matrices $X, Y \in \mathbb{R}^{n \times n}$, $X > Y$ implies $X - Y$ is positive definite and $X \geq Y$ means $X - Y$ is positive semi-definite. With this, it is only necessary to check the triggering sequence that results in the largest possible covariance, since we know that all other sequences (which necessarily result in smaller covariances) must also satisfy the constraint according to Assumption 1.

For ET estimation, this means determining the largest covariance for a given δ_k. Under Assumption 1, this ensures satisfaction of constraints (7) and (8). This is accomplished by calculating the bounding covariance that subsumes all covariances produced by any triggering condition. We use a bounding method similar to the one described in [17].

Let the scalar constants $\underline{a}, \bar{a}, \underline{k}, \bar{k}, \underline{c}, \bar{c}, \underline{q}, \bar{q}, \underline{r}, \bar{r} \in \mathbb{R}_{>0}$ define bounds such that

$$\begin{aligned} &\underline{a}^2 I \leq AA^T \leq \bar{a}^2 I, \qquad \underline{c}^2 I \leq CC^T \leq \bar{c}^2 I, \\ &\underline{k}^2 I \leq (A - BK)(A - BK)^T \leq \bar{k}^2 I, \\ &\underline{q} I \leq Q \leq \bar{q} I, \qquad\qquad \underline{r} I \leq R \leq \bar{r} I, \end{aligned} \quad (18)$$

where I is the identity matrix. The existence of these positive bounds requires that the eigenvalues of A, $(A - BK)$, and CC^T be real.

Theorem 1. *Consider the predicted belief for ET estimation $\boldsymbol{b}(x_k) = \mathcal{N}(\check{x}_k, \Sigma_k + \Lambda_k)$, where the covariance $\Sigma_k + \Lambda_k$ is recursively updated according to* (17) *with initial covariance $\Sigma_0 + \Lambda_0$, and scalars $\underline{a}, \bar{a}, \underline{k}, \bar{k}, \underline{c}, \bar{c}, \underline{q}, \bar{q}, \underline{r}, \bar{r} \in \mathbb{R}_{>0}$ that bound the terms in* (18)*. Then, the covariance belief of the predicted ET estimation $\boldsymbol{b}(x_k)$ can be recursively bounded by*

$$\Lambda_k + \Sigma_k \leq (\bar{\lambda}_k + \bar{p}_k) I, \quad (19)$$

where $\bar{\lambda}_k, \bar{p}_k \in \mathbb{R}_{\geq 0}$ are given by

$$\begin{aligned} \bar{\lambda}_k &= \bar{k}^2 \bar{\lambda}_{k-1} + \frac{\bar{c}^2 \left(\bar{p}_{k-1}\bar{a}^2 + \bar{q}\right)^2}{\underline{c}^2 \left(\underline{p}_{k-1}\underline{a}^2 + \underline{q}\right) + \underline{r}} , \\ \bar{p}_k &= \left(\frac{1}{\bar{p}_{k-1}\bar{a}^2 + \bar{q}} + \frac{\beta \underline{c}^2}{\bar{r} + (1-\beta)\, \bar{c}^2 \left(\bar{a}^2 \bar{p}_{k-1} + \bar{q}\right)} \right)^{-1} , \\ \underline{p}_k &= \left(\frac{1}{\underline{q}} + \frac{\bar{c}^2}{\underline{r}} \right)^{-1} \end{aligned} \quad (20)$$

with the initial values

$$\begin{aligned} \bar{\lambda}_0 &= \max \left(Eigenval \left(\Lambda_0 \right) \right), \\ \bar{p}_0 &= \max \left(Eigenval \left(\Sigma_0 \right) \right), \\ \underline{p}_0 &= \min \left(Eigenval \left(\Sigma_0 \right) \right). \end{aligned}$$

Proof. We begin by deriving the individual bound for Λ_k such that $\Lambda_k \leq \bar{\lambda}_k I$. Consider the case for $\gamma_k = 0$, the update equation can be written as:

$$\Lambda_k = (A - BK)\Lambda_{k-1}(A - BK)^T. \quad (21)$$

This can be simply bounded as $(A - BK)\Lambda_{k-1}(A - BK)^T \leq \bar{k}^2 \bar{\lambda}_{k-1}$. Now consider the case $\gamma_k = 1$. The update equation can be written as:

$$\begin{aligned} \Lambda_k = (A - BK)&\Lambda_{k-1}(A - BK)^T \\ &+ \Sigma_k^- C^T (C \Sigma_k^- C^T + R)^{-1} C \Sigma_k^- . \end{aligned} \quad (22)$$

The term $(A - BK)\Lambda_{k-1}(A - BK)^T$ is the same as (21), and can be bounded in the same way. In order to bound the second term, we must first derive bounds on Σ_k^-, the covariance from the a priori filter update. This can be bounded in terms of the previously defined bounds on the a posteriori update, $\underline{p}_k$ and $\bar{p}_k$. From (17) we can generate the bounds:

$$\left(\underline{a}^2 \underline{p}_{k-1} + \underline{q} \right) I \leq \Sigma_k^- \leq \left(\bar{a}^2 \bar{p}_{k-1} + \bar{q} \right) I \quad (23)$$

Next, examine the term $(C \Sigma_k^- C^T + R)^{-1}$. We require the following lemma to calculate the bound:

Lemma 2. *Let $X \in \mathbb{R}^{n \times n}$ be bounded such that $X \geq \underline{\mathrm{x}} I$. Then, $X^{-1} \leq 1/\underline{\mathrm{x}} I$*

Applying Lemma 2 we obtain:

$$(C \Sigma_k^- C^T + R)^{-1} \leq (\underline{c}^2 (\underline{p}_{k-1} \underline{a}^2 + \underline{q}) + \underline{r})^{-1} I \quad (24)$$

Using (24) and (23), we can write the full bound in (20). It is simple to see that this bound is larger than the bound for the case $\gamma_k = 0$, and therefore is the true upper bound for all triggering conditions.

We use similar reasoning to gererate bounds on bounds on Σ_k. For the case $\gamma_k = 1$, (12) becomes the standard KF equations. Therefore, the upper and lower bounds for the case $\gamma_k = 1$ are the same KF bounds derived in [18]:

$$\left(\frac{1}{\underline{q}} + \frac{\bar{c}^2}{\underline{r}}\right)^{-1} I \leq \Sigma_{k,\gamma_k=1} \leq \left(\bar{p}_{k-1}\bar{a}^2 + \bar{q}\right) I \tag{25}$$

We derive the bounds for the case $\gamma_k = 0$ based on the inverse form of the covariance update equation (12) for $\gamma_k = 0$:

$$\Sigma_{k,\gamma_k=0} = \\ \left(\left(\Sigma_k^-\right)^{-1} + \beta C_k^T \left(R_k + (1-\beta)\, C_k \Sigma_k^- C_k^T\right)^{-1} C_k\right)^{-1}$$

Under straightforward manipulation and application of Lemma 2, this yields the bounds:

$$\left(\frac{1}{\underline{q}} + \frac{\beta \bar{c}^2}{\underline{r} + (1-\beta)\underline{c}^2 \underline{q}}\right)^{-1} I \leq \Sigma_{k,\gamma_k=0} \tag{26}$$

$$\leq \left(\frac{1}{\bar{p}_{k-1}\bar{a}^2 + \bar{q}} + \frac{\beta \underline{c}^2}{\bar{r} + (1-\beta)\,\bar{c}^2\left(\bar{a}^2\bar{p}_{k-1} + \bar{q}\right)}\right)^{-1} I \tag{27}$$

Note that the lowest lower bound on Σ_k corresponds to the case $\gamma_k = 1$. Similarly, the largest upper corresponds to $\gamma_k = 0$. These widest bounds are presented in (19), and are guaranteed to bound the ET filter covariance for any triggering condition.

Finally, we can bound the sum of the covariances $\Sigma_k + \Lambda_k$ by the sum of their respective upper bounds: $\Sigma_k + \Lambda_k \leq (\bar{p}_k + \bar{\lambda}_k)I$. Because these represent the largest upper bound and lowest lower bound for any triggering condition at any time step, the recursively calculated sequence of bounds is guaranteed to bound the expected belief for any possible sequence of triggers. □

V. ET-GBT PLANNING ALGORITHM

This section introduces the Event-Triggered Gaussian Belief Trees (ET-GBT) algorithm, an adaptation of the Gaussian Belief Trees (GBT) algorithm in [14] for ET estimation in order to minimize communication cost.

A. Gaussian Belief Trees

We first present a brief overview of the GBT motion planner from [14]. There, a framework is developed for extending any kinodynamic tree-based motion planner to the belief space, where the edges are still nominal controllers and trajectories, but the nodes are Gaussian beliefs. The algorithm proceeds as follows: a belief is randomly sampled, its closest node is computed using the 2-Wasserstein distance, and extended by a random control input. The uncertainty covariance is propagated using the technique discussed in Sec. III-A. A new node is only added to the tree if it satisfies the chance constraints of probability of collision with obstacles, which is conservatively approximated using [10], [13], [19]. The process repeats until a solution is found.

Algorithm 1: ET-GBT

Input : X, U, $\mathcal{X}_{obs}$, $\mathcal{X}_{goal}$, N
Output: Tree $G = (\mathbb{V}, \mathbb{E})$

```
G ← (𝕍 ← {b_init}, 𝔼 ← ∅)
for N iterations do
    b_rand ← SampleBelief()
    δ_rand ← SampleDelta()
    u_rand ← SampleControl()
    n_select ← SelectNode()
    n_new ← Extend(n_select, δ_rand)
    if ValidPathCheck(n_select, n_new, δ_rand) then
        𝕍 ← 𝕍 ∪ {n_new}
        𝔼 ← 𝔼 ∪ {edge(n_select, n_new, δ)}
    Prune(𝕍, 𝔼)
return G = (𝕍, 𝔼)
```

B. ET-GBT

ET-GBT adapts GBT in two fundamental ways: tree expansion and chance constraint validity checking. ET-GBT can be used to identify a valid tree that optimizes for communication cost in (5), and satisfies the safety constraints (7) and (8). We optimize for this cost function using SST [11], an asymptotically near-optimal planner. Alg. 1 presents the pseudocode for our proposed algorithm.

1) Tree Expansion: Instead of maintaining and propagating Gaussian beliefs, we propagate the bounds (19) on the beliefs under ET per the equations derived in Sec. IV. The rest of the tree expansion algorithm follows intuitively from this main representation change. The SampleBelief() function is unchanged from GBT and operates analogously to the state sampler in an RRT search. The SelectNode() function is modified to select the 'closest' belief node for extension using the 2-Wasserstein distance metric to the upper bounding belief. SampleDelta() has been added so that each edge of the tree corresponds to a triggering threshold as well as a nominal control input.

2) Chance Constraint Validity Checking: We use an over-approximation to check that the probability of collision is below the safety constraint (7). This allows for very fast constraint checking and preserves the efficiency of using a sampling-based algorithm. First, we define the p_{safe} probability contour as the level set $\mathcal{L} = \{s \mid \mathbf{b}(x_k)(s) = c\}$, where $c \in \mathbb{R}_{\geq 0}$. This contour is calculated such that the interior of $\mathcal{L}$ defines an area, $\mathcal{A}$, containing p_{safe} probability mass:

$$\int_{\mathcal{A}} \mathbf{b}(x_k)(s)ds = p_{\text{safe}}. \tag{28}$$

We define the complement of $\mathcal{A}$ as the area not contained in the set: $\tilde{\mathcal{A}} = \{s \mid s \notin \mathcal{A}\}$. Therefore the probability mass in $\tilde{\mathcal{A}}$ is $1 - p_{\text{safe}}$. If $\mathcal{A}$ and $\mathcal{X}_{obs}$ are non-intersecting, then $\mathcal{X}_{obs} \subset \tilde{\mathcal{A}}$ and $P(x_k \in \mathcal{X}_{obs})$ must be less than $1 - p_{\text{safe}}$. The key insight is that the covariance bounds presented in Theorem 1 can be used to calculate the bounding contour that contains all possible contours for any $\mathbf{b}(x_k)$.

The covariance bound produces a scalar matrix, and thus the probability contour becomes an n-sphere. The radius of the n-sphere contour is given by: $r_k = t_\alpha(\bar{\lambda}_k + \bar{p}_k)$, where t_α is computed from the quantile function of the n-dimensional Gaussian distribution: $t_\alpha = -\phi^{-1}(0.5\, p_{\mathrm{safe}})$. Evaluation of (7) is efficiently computed by checking for intersections of the obstacles with the n-sphere, or whether the n-sphere completely contains an obstacle.

Unfortunately, this method can be very conservative. We consider any intersection, no matter how small, as a violation. Therefore situations where the contour barely overlaps are still considered violating. Also, because the covariance bound is circular (i.e., the covariance is identical in all directions), it may have intersection with obstacles where an elliptical covariance does not. This is compounded by the assumptions that $\gamma_k = 0$ or $\gamma_k = 1$ for all k when computing the bounds. This does not occur in reality, so a bound calculated on these extremes is consequentially conservative.

Evaluation of (8) can be done similarly by determining whether the p_{safe} contour is completely enclosed by the goal region.

C. Correctness, Completeness, and Optimality

In this section, we show that ET-GBT is sound and probabilistically complete with respect to the conservative upper bound, and satisfies the condition for asymptotic (near-)optimality of sampling-based tree search planners that sample the control space, such as SST or SST* [11].

Lemma 3. *Let $G = (\mathbb{V}, \mathbb{E})$ be the tree obtained from ET-GBT for some iterations $N \in \mathbb{N}$. Consider any tree node $v \in \mathbb{V}$. This node is guaranteed to satisfy the chance constraints* (7)*. Additionally, if the algorithm returns a solution, the final tree node is guaranteed to satisfy* (8)*.*

Proof Sketch. The proof is straightforward from Assumption 1, Theorem 1 and our method of maintaining worst case covariance bounds for chance constraint checking. □

Theorem 4. *ET-GBT is probabilistically complete with respect to the conservative upper bound, i.e., if there exists a solution using the upper bound, ET-GBT will find it almost surely as $N \to \infty$.*

Proof Sketch. The proof follows directly from Lemma 3 and the probabilistic properties of GBT. □

Asymptotic Optimality: Asymptotic optimality relates to the optimization problem of (6). We first show that the cost function in (5) satisfies the conditions for an admissible cost function according to [20] for asymptotic near-optimality. It is straightforward to see that the cost function satisfies additivity, monotinicity, and non-degeneracy. In the following, we prove that the cost function is Lipschitz continuous in δ.

Lemma 5. *The cost function in* (5) *is Lipschitz Continuous.*

Proof. It is enough to show the Lipschitz continuity of one time step, since the cost function is a sum of stage costs at each time step (5). For $\delta \in \mathbb{R}_{>0}$, we see that $\Gamma(\delta_t) = 1 - [1 - 2\mathcal{Q}(\delta_k)]^m$, is everywhere differentiable, and also that J_t has a bounded first derivative, since $\delta \in \mathbb{R}_{>0}$. Therefore, it is Lipschitz continuous in δ. □

Proving asymptotic optimality of ET-GBT is an open problem, since the bounds in (19) may not be Lipschitz continuous in the triggering threshold δ. However, empirically we see that smooth shifts in δ cause smooth shifts in the upper bound. In our evaluation, using ET-GBT with the asymptotically near-optimal planner SST [11] shows cost function optimizing behavior as planning time increases.

VI. EVALUATIONS

We evaluate our algorithm on two systems and two environments. For each scenario, we generate an METC plan and study it by running MC simulations to collect statistics on the objectives: resource cost, collision probability and probability of reaching the goal. Benchmarking shows improved optimality, i.e. lower cost, with longer runtimes. We implemented our algorithm with the Open Motion Planning Library (OMPL) [21]. All benchmarks were computed single-threaded on a 3.6 GHz CPU. We also provide a comparison with the MDP method in [6].

Simple 2D System: We study the robotic system from [15], with dynamics $x_{k+1} = x_k + u_k + w_k$ and measurements $y_k = x_k + w_k$. The noise is distributed as $w_k \sim \mathcal{N}(0, 0.1^2 I)$ and $v_k \sim \mathcal{N}(0, 0.1^2 I)$.

We first consider an environment with a narrow corridor, shown in Figure 1a, and constraint $p_{\mathrm{safe}} = 0.99$. We ran ET-GBT for 1 minute, generating the METC plan shown in Figure 1a. We see the δ threshold relax in regions far from obstacles; indicating a lower trigger rate, causing the belief bounds to expand and accruing less cost. As the robot traverses the narrow corridor, the threshold tightens, causing the belief bounds to contract in order to satisfy the safety constraint. The threshold decreases at the end to ensure the robot terminates within the goal region.

The generated nominal plan was verified by running 3000 Monte Carlo simulations, none of which collided with obstacles. A sampling of 50 of these trajectories is shown in Figure 1b. This affirms that the chance constraint has been met, but also indicates the bounds are very conservative.

We verified the derived upper bound, $\Lambda_k + \Sigma_k \leq (\bar{\lambda}_k + \bar{p}_k)I$ by checking that the matrix difference, $(\bar{\lambda}_k + \bar{p}_k)I - (\Lambda_k + \Sigma_k)$, is positive semi-definite. This condition is checked by ensuring the eigenvalues of the matrix difference, denoted by $\epsilon = Eigenval((\bar{\lambda}_k + \bar{p}_k)I - (\Lambda_k + \Sigma_k))$, are all positive, as shown in Figure 1c. Note that while the bound is valid, it is very loose, contributing to conservativeness in collision checking. Future work could investigate a tighter bound and reduce the conservatism of our algorithm.

Benchmarks: Next, we performed benchmark analysis to demonstrate that the optimality improves with increased computation time. In addition to the 2D system, we evaluated our algorithm on a second order unicycle system with dynamics $\dot{x} = v\cos(\phi), \dot{y} = v\sin(\phi), \dot{\phi} = \omega, \dot{v} = a$, and feedback linearized according to [22].

We consider two environments: the narrow corridor environment and randomize environments. Each instance of

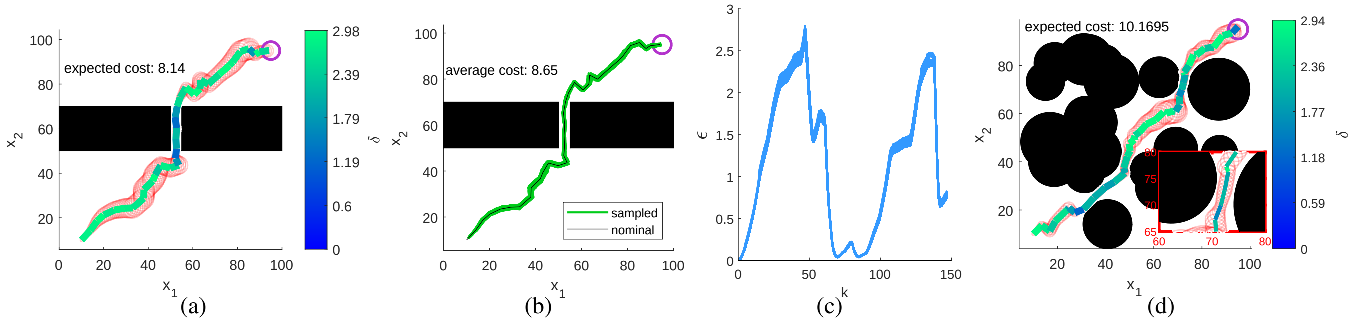

Fig. 1: (a) Nominal trajectory for 2D system, with $\check{X}_0^T$ plotted as lines colored by Δ_0^T, and 99% contours plotted as red circles. (b) 50 sampled MC runs. (c) 50 sampled MC runs of ϵ. (d) Nominal trajectory for 2D system with random obstacles

the randomized environment contains 15 circular obstacles, with obstacle centers, c_o distributed as $c_o \sim \mathcal{U}(0, 100)$, and obstacle radius, r_o, distributed as $r_o \sim \mathcal{N}(10, 2)$. A sample environment and the generated plan for the 2D system are shown in Figure 1d. For both environments, we conducted 100 trials with computation times from 10 seconds to 200 seconds. The results of this analysis are presented in Table I. In each case the average cost of the plans decreases with computation time, showing asymptotic optimal behavior.

TABLE I: Benchmarking results.

Time (s)	2D System Cost Narrow	2D System Cost Random	Unicycle System Cost Narrow	Unicycle System Cost Random
10	38.47 ± 1.0	10.7 ± 1.2	60.2 ± 2.2	105.3 ± 12.4
25	19.76 ± 0.8	7.53 ± 0.9	34.8 ± 0.9	54.4 ± 8.7
50	11.40 ± 0.3	6.61 ± 0.91	25.4 ± 0.6	31.9 ± 2.9
100	8.47 ± 0.2	5.94 ± 0.89	20.5 ± 0.4	25.0 ± 2.0
200	8.13 ± 0.1	5.54 ± 0.87	18.3 ± 0.3	21.5 ± 1.6

Comparison to MDP Method: To showcase the efficiency of our approach, we compare to a Pareto point from the open trajectory scenario in [6]: with probability of goal and collision 0.95 and 0.0 respectively, and cost 30.95. Using the same trajectory and discrete δ values, we ran 100 trials, obtaining a very similar average expected cost of 30.56 within 0.5 seconds. This is significantly faster than the method in [6], which relies on MC sampling to build the MDP. Since this trajectory is far from any obstacles, the conservativeness of our approach does not have a profound effect on the result. Nevertheless, this comparison validates the speed and effectiveness of our optimization approach.

VII. CONCLUSION

This paper considers the problem of generating METC plans that satisfy safety constraints while minimizing communication costs. We develop a novel method of propagating the expected belief under ET estimation, as well the corresponding covariance bounds. We use these techniques to develop a fast and efficient sampling-based METC planning algorithm. Case studies and benchmarking demonstrate the efficacy, speed, and asymptotic optimality of the algorithm.

These methods are limited by the conservativeness of the approximations, which could be addressed in future work by tightening the covariance bounds, or by developing a less conservative collision checking method.

REFERENCES

[1] D. Shi, T. Chen, and L. Shi, "Event-triggered maximum likelihood state estimation," *Automatica*, vol. 50, no. 1, pp. 247–254, 2014.

[2] S. Trimpe and M. C. Campi, "On the choice of the event trigger in event-based estimation," in *2015 Int'l Conf. on Event-based Control, Communication, and Signal Processing (EBCCSP)*, 2015, pp. 1–8.

[3] M. Ouimet, D. Iglesias, N. Ahmed, and S. Martínez, "Cooperative robot localization using event-triggered estimation," *Journal of Aerospace Information Systems*, vol. 15, no. 7, pp. 427–449, 2018.

[4] D. Shi, T. Chen, and L. Shi, "An event-triggered approach to state estimation with multiple point- and set-valued measurements," *Automatica*, vol. 50, no. 6, pp. 1641–1648, 2014.

[5] J. Wu, Q. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.

[6] A. Theurkauf, N. Ahmed, and M. Lahijanian, "Pareto optimal strategies for event triggered estimation," 2022. [Online]. Available: https://arxiv.org/abs/2207.08971

[7] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[8] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, 1999, pp. 473–479 vol.1.

[9] B. Luders, M. Kothari, and J. How, "Chance constrained rrt for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, 2010, p. 8160.

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

[11] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016. [Online]. Available: https://doi.org/10.1177/0278364915614386

[12] L. Kavraki, P. Svestka, J.-C. Latombe, and M. 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.

[13] E. Pairet, J. D. Hernández, M. Carreras, Y. Petillot, and M. Lahijanian, "Online mapping and motion planning under uncertainty for safe navigation in unknown environments," *IEEE Transactions on Automation Science and Engineering*, pp. 1–23, 2021.

[14] Q. H. Ho, Z. N. Sunberg, and M. Lahijanian, "Gaussian belief trees for chance constrained asymptotically optimal motion planning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 11 029–11 035.

[15] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE Int'l Conf. on Robotics and Automation*, May 2011, pp. 723–730.

[16] A. akbar 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. [Online]. Available: https://doi.org/10.1177/0278364913501564

[17] W. Li, G. Wei, D. Ding, Y. Liu, and F. E. Alsaadi, "A new look at boundedness of error covariance of kalman filtering," *IEEE Transac-*

tions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 2, pp. 309–314, 2018.

[18] Y. Li, Z. Littlefield, and K. E. Bekris, “Asymptotically optimal sampling-based kinodynamic planning,” *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016. [Online]. Available: https://doi.org/10.1177/0278364915614386

[19] J. S. Park, C. Park, and D. Manocha, “Efficient probabilistic collision detection for non-convex shapes,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1944–1951.

[20] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, “Analysis of asymptotically optimal sampling-based motion planning algorithms for lipschitz continuous dynamical systems,” 2014. [Online]. Available: https://arxiv.org/abs/1405.2872

[21] I. A. Şucan, M. Moll, and L. E. Kavraki, “The Open Motion Planning Library,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, https://ompl.kavrakilab.org.

[22] A. De Luca, G. Oriolo, and M. Vendittelli, “Stabilization of the unicycle via dynamic feedback linearization,” *IFAC Proceedings Volumes*, vol. 33, no. 27, pp. 687–692, 2000.