Pareto Optimal Strategies for Event-triggered Estimation

Anne Theurkauf, Nisar Ahmed, and Morteza Lahijanian

Abstract—This work considers the problem of resource-performance trade-off analysis for a system consisting of an active agent and a sensor network. Specifically, we focus on Event-triggered (ET) estimation, which allows communication of measurements only when deemed useful, e.g., when Kalman filter innovations exceed some threshold. We introduce a framework for ET-threshold strategy synthesis that optimally trades off resource consumption with task performance. Our approach is based on a novel belief space discretization technique that abstracts a continuous-space dynamics model for ET estimation to a finite Markov decision process. We show this abstraction method is more efficient than alternatives and also benefits from interpretability. Then, we leverage off-the-shelf tools to compute the set of all optimal trade-offs between resource consumption and task performance. Given a choice from this set, we synthesize an ET-threshold strategy that achieves the desirable behavior. Simulated results show our approach identifies non-trivial trade-offs between performance and energy savings, with only modest computational effort.

I. INTRODUCTION

As autonomous agents become more prevalent in society, they must be able to reliably perform given tasks with limited resources. This is a difficult challenge given that these two aspects are naturally competing: often the best performance of a task comes at a greater expense of resources. One facet of this problem is the cost of accurate estimation of the system state; more measurements of the state can be costly, but lead to better estimation and hence better decisions. This is a particularly troublesome issue for networked systems that rely on shared information, where communication cost directly competes with the accuracy of state estimation.

For example, a system might combine mobile agents with remote sensors as shown in Figure 1. While the sensors can make and communicate measurements with the agents, they may have limited computing capabilities and battery life and, in remote environments, may be difficult to access to change batteries. According to [1], wireless transmission is at least an order of magnitude more costly in terms of energy consumption than any other function in a standard ZigBee networking chip. Maintaining consistent operation in these kinds of scenarios necessarily means limiting communication in the remote sensing network. Additionally, systems may be bandwidth limited, in which case limiting communication frees up space for other valuable information, e.g., science data in robotic exploration missions. In this paper, we develop a formal framework to explore optimal trade-offs between communication cost and task performance.

This work was supported by NASA STTR award 80NSSC20C0314. Authors are with the Department of Smead Aerospace Engineering Sciences, University of Colorado Boulder, CO, USA. {firstname.lastname}@colorado.edu

A popular method of managing communication costs is event-triggered (ET) estimation [2]–[5]. In the ET framework, information is only communicated when it is deemed useful. This condition can take many forms [2], but a common choice is a threshold on the innovation of a measurement [3]–[5]. Changing this threshold changes the frequency of communication. While this can be a powerful tool for resource-performance trade-off, there are no methods for tuning this threshold to guarantee the trade-off is optimal and accounts for task performance.

Trade-off analysis techniques have been studied mostly in formal methods literature [6]–[9]. Given a set of quantitative objectives and system model, the goal of multi-objective analysis is to calculate the set of all optimal trade-offs between the objectives, called the Pareto front. Recent studies explore applications in robotics [10], path planning [11], [12], multi-agent coordination [13]–[15], and sensor scheduling [16]. The latter specifically focuses on resource and performance objectives and develops a scheduling framework for the usage of a high precision, high cost sensor. Nevertheless, finding Pareto-optimal communication strategies between a sensor network and mobile agents with respect to multiple objectives remains an open problem.

In this paper, we fill this gap by introducing a method of synthesizing strategies for setting ET thresholds to achieve optimal trade-offs between resource consumption and task performance. We specifically consider a scenario where a mobile agent has to reach a goal region while avoiding unsafe states (obstacles) by communicating with a resource-limited remote sensor network. Our approach is based on abstracting this system to a discrete Markov Decision Process (MDP). The abstraction involves a novel belief space discretization method using the spectral decomposition of the estimated covariance. We show this abstraction method is more efficient than alternatives and also benefits from interpretability. Then, we leverage off-the-shelf tools (PRISM [17], PRISM-games [18]) to obtain the Pareto front and synthesize ET strategies for individual points on the front. We show the efficacy of the approach in several case studies.

Fig. 1: Diagram of triggering and estimation process.
II. PROBLEM FORMULATION

We consider an active agent communicating with a resource-limited remote sensor network. The nearest sensor to the agent is able to make and communicate measurements of the agent’s state, but communication comes at a resource cost. This setup can apply to many types of measurements, e.g., remote one-way ranging measurements for a mobile robot from beacons whose battery energy and data volume must be conserved in austere environments. The goal is to find a communication strategy to optimally trade-off sensor resource costs and agent task performance.

A. System Model

The active agent evolves according to linear dynamics,

\[ x_{k+1} = F x_k + G u_k + w_k, \quad w_k \sim \mathcal{N}(0, Q), \]

where \( x_k \in X \subseteq \mathbb{R}^n \) and \( u_k \in U \subseteq \mathbb{R}^p \) are the state and input, respectively, with \( F \in \mathbb{R}^{n \times n} \) and \( G \in \mathbb{R}^{n \times p} \). Random process noise \( w_k \in \mathbb{R}^n \) is modeled as white Gaussian noise with covariance \( Q \in \mathbb{R}^{n \times n} \). The initial agent state \( x_0 \sim \mathcal{N}(\hat{x}_0, P_0) \) is assumed Gaussian distributed, with mean \( \hat{x}_0 \in \mathbb{R}^n \) and covariance \( P_0 \in \mathbb{R}^{n \times n} \). The nearest remote sensor measures the agent state as

\[ y_k = H x_k + v_k, \quad v_k \sim \mathcal{N}(0, R), \]

where \( y_k \in Y \subseteq \mathbb{R}^m \) is the measurement with \( H \in \mathbb{R}^{m \times n} \). Random measurement noise \( v_k \in \mathbb{R}^m \) is also modeled as white Gaussian noise with covariance \( R \in \mathbb{R}^{m \times m} \). We assume the system is both controllable and observable, and that the noise covariance matrices \( Q \) and \( R \) are positive definite. For ease of presentation, we assume the agent only has access to measurement data from the nearest remote sensor. The extension to include local measurements is trivial.

B. Event-Triggered State Estimation

Since the sensors are resource limited, the agent uses an event triggered (ET) state estimator for efficient information sharing. When a measurement \( y_k \) is taken, the sensor determines whether to report \( y_k \) to the agent. This decision is based on a shared estimate of \( x_k \) provided by some central estimator, e.g., the agent. We specifically focus on innovation-based ET described in [5].

Let \( \gamma_k \in \{0, 1\} \) be the indicator for measurement communication at time \( k \), where \( \gamma_k = 1 \) if the measurement is sent, and \( \gamma_k = 0 \) otherwise. Define \( \mathcal{I}_k = (I_1, \ldots, I_k) \), \( k \in \mathbb{N} \), as the sequence of available information up to time \( k \), where

\[ I_j = \begin{cases} \{y_j\} & \text{if } \gamma_j = 1, \\ \emptyset & \text{otherwise}. \end{cases} \quad \text{for } 1 \leq j \leq k. \]

Information \( \mathcal{I}_k \) is used to reason over the probability distribution of \( x_k \). This distribution is denoted by \( b_k \) and is referred to as the belief, i.e.,

\[ x_k \sim b_k = \mathcal{P}(x_k \mid x_0, \mathcal{I}_k), \]

with the set of all beliefs denoted by \( \mathbb{B} \).

Under the assumption that \( \mathcal{P}(x_k \mid x_0, \mathcal{I}_{k-1}) \) is Gaussian, work [5] introduces an ET Minimum Mean Square Error (MMSE) estimator, where the belief is Gaussian. The expected value of the state based on the belief is the MMSE state estimate. Before a measurement occurs, the \textit{a priori} (predicted) state estimate and error \( e_k^- \) are given by

\[ \hat{x}_k^- = \mathbb{E}[x_k \mid \mathcal{I}_{k-1}], \quad e_k^- = x_k - \hat{x}_k^-, \]

\[ P_k^- = \mathbb{E}[e_k^- e_k^{T} \mid \mathcal{I}_{k-1}]. \]

After a measurement occurs, the \textit{a posteriori} (updated) estimate and error \( e_k \) are given by

\[ \hat{x}_k = \mathbb{E}[x_k \mid \mathcal{I}_k], \quad e_k = x_k - \hat{x}_k, \]

\[ P_k = \mathbb{E}[e_k e_k^{T} \mid \mathcal{I}_k]. \]

In innovation-based ET, \( \gamma_k \) is determined based on the measurement innovation, \( z_k = y_k - \mathbb{E}[y_k \mid \mathcal{I}_{k-1}] \), which is a Gaussian random vector with covariance \( P_k = R + H P_k^- H^{T} \). Let \( F_k = V_k \Lambda_k^{-1/2} \), where the columns of \( V_k \in \mathbb{R}^{n \times n} \) are the orthonormal eigenvectors of \( Z_k \), and \( \Lambda_k \in \mathbb{R}^{n \times n} \) is the diagonal eigenvalue matrix of \( Z_k \), so that the covariance of the whittened innovation \( e_k = F_k z_k \) is the identity matrix. The triggering condition is then defined in terms of a given threshold \( \delta \in \mathbb{R}_0^+ \) as,

\[ \gamma_k = \begin{cases} 0 & \text{if } \|e_k\|_{\infty} \leq \delta, \\ 1 & \text{otherwise.} \end{cases} \]

This corresponds to assessing how ‘surprising’ \( y_k \) is. For small \( \delta \), many measurements are surprising, and many triggers (communications) occur; and vice versa for large \( \delta \). As such, (7) and (8) can be obtained through a recursive Kalman filter approximation, which attenuates the Kalman gain as a function of \( \delta \) when \( \gamma_k = 0 \), and produces a standard Kalman update if \( \gamma_k = 1 \) or \( \delta = 0 \) (see [5] for complete details). The key advantage of the ET filter is the ‘implicit’ information that \( \|e_k\|_{\infty} \leq \delta \) when \( \gamma_k = 0 \) (i.e., \( I_k = \emptyset \)). This can be used to compute a more accurate \textit{a posteriori} estimate than the one obtained simply by keeping an unmodified \textit{a priori} estimate from the Kalman prediction step.

Under ET estimation, the agent’s task performance and sensor resource consumption now depend on the choice of \( \delta \). However, there are no established methods to algorithmically select or adjust \( \delta \) to provide collective formal guarantees on task performance and resource consumption effects. This work fills that gap.

C. Task and Control Laws

The agent’s task is to follow a nominal trajectory to a target region while avoiding unsafe states (e.g., obstacles). Assume the trajectory is provided via a sequence of waypoints \( W = (w_0, \ldots, w_N) \) such that at \( w_i \in \mathbb{R}^n \), for each \( i \in \{0, \ldots, N\} \), the nearest sensor to the agent changes. Such waypoints can be easily obtained, e.g., using a motion planner followed by a Voronoi segmentation with respect to the sensor network. Hence, the waypoints are the decision
points where the triggering threshold for the next sensor (and thus the corresponding trajectory segment) is determined.

To follow the trajectory, the agent is equipped with a series of (feedback) control laws connecting the waypoints in \( W \), coupled with a termination condition that determines when to switch to the next controller. Let \( \mathcal{U}_i : X \times W \to U \) be the control law that drives the agent from waypoint \( w_{i-1} \) to \( w_i \) and \( \zeta_i : X \times W \times \mathbb{N} \to \{0,1\} \) be the corresponding termination condition. Then, the agent applies control law \( \mathcal{U}_i \) until \( \zeta_i \) is triggered. We design \( \zeta_i \) to trigger as

\[
\zeta_i(x_k, w_i, k_i) = \begin{cases} 1 & \text{if } \|x_k - w_i\|_2 \leq \varepsilon_x \text{ or } k_i \geq k_{\text{max}} \\ 0 & \text{otherwise}, \end{cases}
\]

where \( \varepsilon_x \in \mathbb{R}_{>0} \) is the convergence tolerance, \( k_i \) is time duration since switching to \( \mathcal{U}_i \), and \( k_{\text{max}} \) is the max time duration threshold. The only requirement on \( \mathcal{U}_i \) is to drive the agent to proximity of \( w_i \). It can, for instance, consist of a reachability and a stabilizing controller using the LQG control architecture described in [16], [19].

D. Resource and Performance Objectives

We seek to choose ET thresholds at waypoints in \( W \) to achieve an optimal resource-performance trade-off. Let \( \Delta = \{\delta_1, \delta_2, \ldots, \delta_{|\Delta|}\} \) be a set of ET threshold candidates, and \( D(\Delta) \) be the set of all probability distributions over \( \Delta \). We define an ET strategy \( \omega : \mathbb{B} \to D(\Delta) \) to be a function that maps a belief \( b_k \in \mathbb{B} \) at a waypoint to a probability distribution over \( \Delta \). We want to compute \( \omega \) according to the following three competing objectives: (O1) minimize the probability of visiting an unsafe state (obstacle), (O2) maximize the probability of reaching the target region, and (O3) minimize resource consumption. We refer to (O1) and (O2) as the performance objective and (O3) as the resource objective. We now formalize (O1)-(O3) as a function of \( \omega \).

Given \( \omega \), \( b_k \) itself is a random variable before agent deployment due to the stochasticity of \( y_k \). To provide guarantees prior to deployment, we express the objectives based on the expected belief defined as:

\[
b_k(\omega) = \mathbb{E}_Y(b_k \mid x_0, y_{0:k}, \omega) = \int_{y_{0:k}} \mathcal{P}(x_k \mid x_0, y_{0:k}, \omega) \mathcal{P}(y_{0:k}) dy = \mathcal{P}(x_k \mid x_0, \omega),
\]

Consider a set of \( M \) obstacles (unsafe states) \( \mathcal{O} = \{X_0, X_1, \ldots, X_M\} \), where \( X_i \subseteq X \) for every \( i \in \{1, \ldots, M\} \). Assuming that collisions with obstacles are terminal, the probability of collision with obstacle \( X_i \) under \( \omega \) is

\[
P_{\text{coll},i}(\omega) = \mathcal{P}(x_{0:k_T} \in X_{i} \mid \omega) = \int_{X_{i}} b_{k_T}(\omega) dx_{k_T},
\]

where \( k_T \) is the termination time of traversing trajectory \( W \). Then, for (O1), the total probability of collision is the sum of the probabilities over all the individual obstacles,

\[
P_{\text{coll}}(\omega) = \mathcal{P}(x_{0:k_T} \in \mathcal{O} \mid \omega) = \sum_{i=1}^{M} P_{\text{coll},i}(\omega).
\]

Similarly, for (O2), the probability of ending in the target region \( X_{\text{tar}} \subset \mathcal{X} \) is given by

\[
P_{\text{tar}}(\omega) = \mathcal{P}(x_{k_T} \in X_{\text{tar}} \mid \omega) = \int_{X_{\text{tar}}} b_{k_T}(\omega) dx_{k_T}.
\]

For (O3), the resource cost can generally be defined as a function of \( u, w \), and \( \delta \). For ease of presentation, we define this cost solely based on the energy consumed by communicating measurements, but extending to include actuation, or anything else, is straightforward. The instantaneous communication cost at time \( k \) is \( c_m \gamma_k \), where \( c_m \in \mathbb{R}_{\geq 0} \). The expected \( \gamma_k \) is a function of \( \delta \), i.e. \( \gamma_k(\delta) = \mathbb{E}[\gamma_k] \). Then, the expected total cost over a trajectory is

\[
C_{E}(\omega) = \sum_{k=0}^{k_T} \sum_{\delta \in \Delta} \omega(\delta) c_m \gamma_k(\delta).
\]

E. Problem Statement

We consider the following problem: given a system model as described in Section II-A, a set of obstacles \( \mathcal{O} \), a target region \( X_{\text{tar}} \), waypoints \( W \), controllers as defined in Section II-C, and a set of ET thresholds \( \Delta \), compute optimal ET strategy \( \omega^* \) such that

\[
\omega^* = \arg \min_{\omega} \left( C_{E}(\omega), P_{\text{coll}}(\omega), 1 - P_{\text{tar}}(\omega) \right),
\]

where \( \min \) is a simultaneous minimization of every element.

Note that (15) is a multi-objective optimization problem, and since the objectives are (often) competing, there may not exist a single solution that simultaneously optimizes each objective. For this reason, we study the set of all optimal trade-offs between the objectives (Pareto front). Then, given a point from this set, we synthesize the corresponding \( \omega^* \).

It is computationally intractable to directly calculate the set of all trade-offs for a continuous system. We instead approach this problem by constructing an abstraction of the continuous system as a finite MDP. Established algorithms for multi-objective optimization of MDPs are polynomial with the size of the MDP and implemented in tools such as PRISM [17], which we use for our analysis.

III. FINITE ABSTRACTION

Finite abstraction is often performed via a discretization of the continuous state space. In absence of sensing uncertainty, this is easily addressed by polytopic partitions, e.g., a grid. However, with measurement noise, the true state of the system is unknown; hence, we must reason over the belief space. Recall from (4) that the belief at a waypoint is dependent on the complete history of measurements and threshold choices. Consequently, the belief evolution is an exponentially growing graph [16]. To avoid analysis of the full graph, we propose two abstraction methods via (i) enforcing convergence to a single belief at each waypoint, and (ii) grouping the beliefs at each waypoint (analogous to a polytopic discretization of state space). With these methods, the transition from a belief \( b \) at waypoint \( w_i \) to another belief at \( w_{i+1} \) becomes dependent only on \( b \) and the choice \( \delta \) at \( w_i \). This naturally leads to an MDP abstraction where the MDP states are formed by the converged belief or belief sets.
A. MDP Abstraction

An MDP $M = (S, s_0, A, T, C)$ is a tuple consisting of a finite set of states $S$, an initial state $s_0 \in S$, a set of actions $A$, a probabilistic transition function $T : S \times A \rightarrow [0, 1]$, and a cost function $C : S \times A \rightarrow \mathbb{R}_{\geq 0}$ that assigns to each state-action pair a non-negative cost.

In our MDP abstraction, each state $s \in S$ is defined as a set of beliefs at each waypoint, and the action set is the set of ET thresholds, i.e., $A = \Delta$. The cost $C(s, \delta)$, where $\delta \in \Delta$, is the expected communication cost from the waypoint $w$ corresponding to $s$, to the next waypoint, and is given by

$$C(s, \delta) = \bar{k}(w, \delta) \gamma(\delta), \quad (16)$$

where $\bar{k}(w, \delta)$ is the expected number of time steps required to navigate from $w$ to the the next waypoints under $\delta$. Further, we include three states $s_{coll}$, $s_{tar}$, and $s_{free}$ in addition to the belief states in $S$ to represent termination in an obstacle, target region, and neither, respectively.

Below, we detail two methods to construct the MDP states: enforced KF convergence and discretized belief states. For both methods, we use Monte Carlo sampling to approximate the transition probabilities, $T$ [16].

B. Method 1: Enforced Belief Convergence

This method relies on convergence to a pre-computed Gaussian belief at waypoint $w$, and is defined by the KF steady state covariance $P_{KF} \in \mathbb{R}^{n \times n}$, i.e., $\mathcal{N}(w, P_{KF})$.

To ensure convergence to this belief, we force the system to switch from ET to KF estimation within some neighborhood of $w$. Let $\varepsilon_{KF} \in \mathbb{R}_{>0}$ be the radius of this neighborhood. The system switches to KF when $\|\hat{x}_k - w_{i+1}\|_2 \leq \varepsilon_{KF}$. Then, it continues under KF until both the covariance convergence condition, $\|P_k - P_{KF}\|_{\text{max}} \leq \varepsilon_P$, where $\varepsilon_P \in \mathbb{R}_{>0}$, and (10), are met. The belief state at each waypoint is thus known to be Gaussian with a covariance within $\varepsilon_P$ of $P_{KF}$ and a mean within $\varepsilon_x$ of the waypoint. So, the MDP state under the Enforced KF Convergence Method is defined as a tuple: $s = (w, \varepsilon_x, P_{KF}, \varepsilon_P)$.

Note that, under every choice of $\delta$, the system converges to the same MDP belief state at each waypoint, hence the total number of states is $N + 3$, where $N$ is the number of waypoints. Despite its low complexity, this method suffers from overly constraining the system, and thus does not fully exploit the benefits of ET estimation.

C. Method 2: Discretized Belief State

The second method operates only in the ET mode and uses only the mean convergence criterion in (10) for termination. Stochastic $y_k$ induces random switching between the two possible cases for the $P_k$ information update (implicit and explicit), precluding convergence to steady state $P_k$. This means $P_k$ at each waypoint can be quite variable. Therefore, we form the MDP states from sets of $b_k$ at each waypoint with $\hat{x}_k$ captured by criterion in (10) and $P_k$ captured by a discretized covariance state (described below). The MDP obtained using this technique leads to history-independent state transitions only up to the resolution used for the covariance discretization, called abstraction error. In the full version of this work [20], we show that the abstraction error goes to zero as the discretization size goes to zero.

A naive approach would be to directly discretize the individual elements of the $P_k$ matrix. However, as we show later, this approach is inefficient and exacerbates the state explosion problem. Instead, we rely on the spectral decomposition of $P_k$, which results in $n$ eigenvectors describing the orientation of the axes and $n$ eigenvalues describing the magnitudes of an associated uncertainty (hyper)ellipsoid, as shown in Figure 2a. The covariance state is then taken as a combination of discrete regions across each of the $n$ axes.

A discrete region for each axis is defined by 3 elements: a nominal vector, an angle range, and a magnitude range. The nominal vector, $v_{nom} \in \mathbb{R}^n$, is a unit vector that provides the orientation of the region. While the nominal vector can be chosen as any vector in $\mathbb{R}^n$, a good choice is to base it on empirically sampled data. The angle range between $\theta_{low}, \theta_{high} \in (0, \pi/2]$, is defined relative to the $v_{nom}$, with the angle calculated via the Euclidean inner product. The magnitude range is between $\lambda_{low}, \lambda_{high} \in \mathbb{R}_{>0}$. Various angle and magnitude choices are shown in Figures 2b and 2c. A region described by $v_{nom}$, angles $\theta_{low}$ and $\theta_{high}$, and magnitudes $\lambda_{low}$ and $\lambda_{high}$, is defined as the set

$$R = \{(\lambda, \theta) \mid \lambda \in [\lambda_{low}, \lambda_{high}] \land \theta \in [\theta_{low}, \theta_{high})\}, \quad (17)$$

for eigenvalue $\lambda$ and angle $\theta$ from the eigenvector to $v_{nom}$ (the combinations of angle and magnitude ranges to form regions are shown as colored areas in Figure 2d).
The full covariance state, \( c_R \) is taken as a combination of regions for each dimension: \( c_R = (R_1, \ldots, R_n) \). Figure 2e shows the decomposition of an example covariance, and the corresponding \( c_R \) it falls in. The MDP state under the Discretized Belief State Method is defined as a tuple: \( s = (w, \varepsilon_C, c_R) \).

Defining the states as described relies on some knowledge of the upper and lower bounds of the covariance. While the lower bound can be trivially obtained via the KF steady state covariance, we currently have no method of determining the theoretical upper bound for the covariance. Although this is a potential avenue for future work, a good empirical bound can be obtained by propagating the covariance through the update equation and enforcing \( \gamma_k = 0 \) at every time step. Additionally, the angle and magnitude ranges are also empirically chosen such that the abstraction error (discuss in the following section) is low, but the number of samples required for Monte Carlo sampling is not too large.

1) Discretization Error: Abstraction errors due to discretization manifest in the transition probabilities: belief states originating from some parts of the same region (as defined above) have different transition probabilities from those originating in different parts of the same region. We provide a proof in the full version of this work [20] that as the volume of the discretized belief state, i.e., the size of the ranges \((\theta_{\text{low}}, \theta_{\text{high}})\) and \((\lambda_{\text{low}}, \lambda_{\text{high}})\), goes to zero, the abstraction error goes to zero.

2) Scalability: A naive method of covariance discretization could involve setting ranges on the individual elements of the covariance matrix, rather than relying on the spectral decomposition. The full MDP state could then be described as a combination of regions across each element of the covariance matrix. Since the covariance matrix is symmetric, the number of regions, \( N_R \), required to describe a state is \( N_R = n(n+1)/2 \), where \( n \) is the dimension of the state space. Whereas the spectral decomposition requires only a single region for each axis, so the number of regions required to describe a state is simply \( N_R = n \).

Now assume that each matrix element is discretized into \( d \) regions, and similarly that each axis in the spectral decomposition is also discretized into \( d \) regions. The total number of combinations across all the regions is \( d^{N_R} \). While this number explodes for both cases, it explodes less rapidly for the discretization based on the spectral decomposition. If we assume the same discretization at every waypoint, the total number of MDP states is then \( N_R d^{N_R} + 3 \). While this is a worst case scenario (in practice, not every combination of regions is occupied), the described techniques do suffer from a state explosion problem, which is particularly troublesome as the dimensionality of the system increases. Finally, the spectral decomposition also benefits from visual interpretability; it is easy to visualize the uncertainty of the system in terms of vectors defining a covariance ellipse.

Off the shelf tools are available to solve the multi-objective problem on an MDP [17]. These use value iteration to construct a Pareto Front [7], from which specific strategies are generated through linear programming [6]. The MDP analysis algorithms are polynomial with the size of the MDP.

IV. EVALUATIONS

Here, we demonstrate the benefits of trade-off analysis by showing that small sacrifices in task performance can lead to large energy savings. We present Pareto Fronts for a 2D and 3D system and simulations of strategies of interesting points. In all cases, the error between theory and simulation was less than 2% for the 2D and less than 3.5% for 3D system. For more case studies and detailed discussions see [20].

2D System: We considered the robotic system from [21] with dynamics

\[
x_{k+1} = x_k + u_k + w_k, \quad y_k = x_k + w_k, \quad (18)
\]

where \( w_k \sim \mathcal{N}(0, 0.07^2 I) \) and \( v_k \sim \mathcal{N}(0, 0.03^2 I) \). For clarity, the communication cost is \( c_m = 1 \) so that the energy is simply the raw number of triggers. We considered a trajectory through a set of obstacles as shown in Figure 3a. For this trajectory, we built an MDP using both of the two abstraction methods described above. The resulting Pareto fronts are shown in Figure 4.

Each front shows a characteristic shape; relatively large energy savings are achievable for the highest probabilities of reaching the target, with smaller savings as the probability of target goes to its minimum. This shape is common across every Pareto Front generated in these studies, and supports the argument that this kind of analysis can provide valuable energy savings for little sacrifice in performance objectives.

The second note from these two fronts is the contrast between the two methods. At the highest energy consumption, both methods offer similarly high probability of reaching the target. However, the discretized belief method drops to far lower energy values than the KF convergence method.
indicating that method is better able to leverage ET and provide greater energy efficiency.

To more closely examine interesting trade-offs for the Discretized Belief Method, consider the table below. It shows 3 selected Pareto Points of interest in addition to expected performance if a regular KF is used at every time step.

| Pareto Point | $P_{\text{coll}}$ | $P_{\text{et}}$ | $E_c$ |
|--------------|-----------------|----------------|-------|
| 1            | 0.969           | 0.031          | 367.99|
| 2            | 0.950           | 0.047          | 185.07|
| 3            | 0.835           | 0.147          | 92.35 |
| Full KF      | 0.957           | 0.397          | 682.32|

From Pareto Point 1, the point with the highest probability of reaching the target, Pareto point 2 sacrifices only 1.9% probability of target, but uses half the energy. However, while Pareto Point 3 represents another halving of the energy, the probability of target drops by 11.5%. Also note the stark comparison with the full KF performance; at its highest probability of reaching the target, ET achieves the similar performance for 54% the energy use.

Figure 3 shows 3000 simulated runs of each strategy, with the triggering threshold indicated by the color of the trajectory. Each threshold can be directly mapped to an expected triggering rate (see [5]). The efficiency of the strategies can be seen in how the triggering rate only increases (lower $\delta$’s), contracting the trajectory, in proximity to obstacles. Trivially, we can see that the lower energy Pareto points correspond to lower triggering rates (higher $\delta$’s) around the obstacles.

**3D System:** Finally, to demonstrate the flexibility of our abstraction method, we consider the same system as (18), but extended to three dimensions. A selection of interesting Pareto Points and a single simulated strategy for a winding trajectory using the Discretized Belief Method is shown in Figure 5. Note that the same general shape of the Pareto front holds for the 3D system, with generous trade-offs at higher energies that taper as the energy decreases. The simulated trajectory corresponds to the highest probability of target and shows similar behavior to the 2D system, with higher triggering rates closer to the obstacles.

**Fig. 5:** (a) Pareto front and simulated strategy for 3D System; (b) ‘Super G’ trajectory, using discretized belief method.

**V. CONCLUSION**

In this work, we introduced a method of calculating optimal trade-offs between communication cost of a distributed sensor network via ET, and task performance of an active agent. Our method relies on a novel geometric discretization using the spectral decomposition of a covariance matrix
to group belief states of similar size and orientation. This technique allows us to fully leverage the benefits offered by ET. A limitation of this work is scalability. A future research direction is to employ adaptive discretization techniques to gain efficiency, extending the method to higher dimensions.

**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 (EBCDSP)*, 2015, pp. 1–8.

[3] M. Ouiimet, 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 performance,” *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.

[6] K. Etessami, M. Kwiatkowska, M. Y. Vardi, and M. Yannakakis, “Multi-objective model checking of markov decision processes,” in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Berlin Heidelberg, 2007, pp. 30–65.

[7] V. Forejt, M. Kwiatkowska, and D. Parker, “Pareto curves for probabilistic model checking,” in *Automated Technology for Verification and Analysis*. Springer Berlin Heidelberg, 2012, pp. 317–332.

[8] E. M. Hahn, V. Hashemi, H. Hermans, M. Lahijanian, and A. Turrini, “Multi-objective robust strategy synthesis for interval Markov decision processes,” in *Int’l Conf. on Quantitative Evaluation of Systems*. Berlin, Germany: Springer, Sep. 2017, pp. 207–223.

[9] ——, “Interval Markov decision processes with multiple objectives: From robust strategies to pareto curves,” *ACM Transactions on Modeling and Computer Simulation*, vol. 29, no. 4, pp. 1–31, 2019.

[10] I. Romanova, *Pareto Optimal Solutions and Their Application in Designing Robots and Robotic Systems*. Cham: Springer International Publishing, 2020, pp. 25–39.

[11] A. Lavin, “A pareto front-based multiobjective path planning algorithm,” 2015, [Online]. Available: https://arxiv.org/abs/1505.05947

[12] J. Lee, D. Yi, and S. S. Srinivasa, “Sampling of pareto-optimal trajectories using progressive objective evaluation in multi-objective motion planning,” in *2018 IEEE/RSJ Int’l Conf on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[13] J. Zhao and M. Zhu, “Pareto-optimal multirobot motion planning,” *IEEE Trans. on Autom. Control*, vol. 66, no. 9, pp. 3984–3999, 2021.

[14] R. Ghrist and S. M. Lavalle, “Nonpositive curvature and pareto optimal coordination of robots,” *SIAM Journal on Control and Optimization*, vol. 45, no. 5, pp. 1697–1713, 2006.

[15] R. Cui, B. Gao, and J. Guo, “Pareto-optimal coordination of multiple robots with safety guarantees,” *Autonom. Robots*, vol. 32, no. 3, 2012.

[16] M. Lahijanian, M. Svorenova, A. A. Morve, B. Yeomans, D. Rao, I. Posner, P. Newman, H. Kress-Gazit, and M. Kwiatkowska, “Resource-performance tradeoff analysis for mobile robots,” *IEEE Robotics and Automation Letters*, no. 3, pp. 1840–1847, 2018.

[17] M. Kwiatkowska, G. Norman, and D. Parker, “Prism 4.0: Verification of probabilistic real-time systems,” in *Computer Aided Verification*. Springer Berlin Heidelberg, 2011, pp. 585–591.

[18] J. Lee, D. Yi, and S. S. Srinivasa, “Sampling of pareto-optimal trajectories using progressive objective evaluation in multi-objective motion planning,” in *2018 IEEE/RSJ Int’l Conf on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[19] ——, “Interval Markov decision processes with multiple objectives: From robust strategies to pareto curves,” *ACM Transactions on Modeling and Computer Simulation*, vol. 29, no. 4, pp. 1–31, 2019.

[20] I. Romanova, *Pareto Optimal Solutions and Their Application in Designing Robots and Robotic Systems*. Cham: Springer International Publishing, 2020, pp. 25–39.

[21] A. Lavin, “A pareto front-based multiobjective path planning algorithm,” 2015, [Online]. Available: https://arxiv.org/abs/1505.05947

[22] J. Lee, D. Yi, and S. S. Srinivasa, “Sampling of pareto-optimal trajectories using progressive objective evaluation in multi-objective motion planning,” in *2018 IEEE/RSJ Int’l Conf on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[23] J. Zhao and M. Zhu, “Pareto-optimal multirobot motion planning,” *IEEE Trans. on Autom. Control*, vol. 66, no. 9, pp. 3984–3999, 2021.

[24] R. Ghrist and S. M. Lavalle, “Nonpositive curvature and pareto optimal coordination of robots,” *SIAM Journal on Control and Optimization*, vol. 45, no. 5, pp. 1697–1713, 2006.

[25] R. Cui, B. Gao, and J. Guo, “Pareto-optimal coordination of multiple robots with safety guarantees,” *Autonom. Robots*, vol. 32, no. 3, 2012.

[26] M. Lahijanian, M. Svorenova, A. A. Morve, B. Yeomans, D. Rao, I. Posner, P. Newman, H. Kress-Gazit, and M. Kwiatkowska, “Resource-performance tradeoff analysis for mobile robots,” *IEEE Robotics and Automation Letters*, no. 3, pp. 1840–1847, 2018.

[27] M. Kwiatkowska, G. Norman, and D. Parker, “Prism 4.0: Verification of probabilistic real-time systems,” in *Computer Aided Verification*. Springer Berlin Heidelberg, 2011, pp. 585–591.

[28] J. Lee, D. Yi, and S. S. Srinivasa, “Sampling of pareto-optimal trajectories using progressive objective evaluation in multi-objective motion planning,” in *2018 IEEE/RSJ Int’l Conf on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[29] J. Zhao and M. Zhu, “Pareto-optimal multirobot motion planning,” *IEEE Trans. on Autom. Control*, vol. 66, no. 9, pp. 3984–3999, 2021.