Focused Optimization for Online Detection of Anomalous Regions*

Juan Pablo Mendoza[1], Manuela Veloso[2] and Reid Simmons[3]

***Abstract*— This paper presents an online algorithm for early detection of anomalies in robot execution, where the anomalies occur in a particular region of the robot's state space. Assuming that a model of normal execution is given, the algorithm detects regions of space where data significantly deviate from normal. It achieves this by focusing optimization over a fixed-parameter family of shapes to find the one among them that is most likely anomalous, and then using this region to decide whether execution is anomalous. Experiments using synthetic and real robot data support the effectiveness of the approach.**

I. Introduction

Autonomous robots often encounter unexpected situations and failures as they perform tasks. The goals of *execution monitoring* [1] are to enable robots to autonomously *detect* failures or anomalies in execution, *diagnose* the characteristics of these failures, and *recover* from them. This paper focuses on the first two of these subproblems, for failures in which (1) the anomaly occurs in only a subset of the state space X, (2) the anomaly is too subtle to detect from a single observation, but is apparent only when considering a set of nearby observations, and (3) the relevant observations are not necessarily sequential in time. Examples of these regional failures in robotics domains may include:

- A mobile robot whose motion is inaccurate at certain speeds or in some areas of its environment.
- A golf putting robot that misses more shots than expected from a particular region of the green.
- A manipulating robot whose robustness is lower than expected when using a set of similar grasps.
- A service robot whose task success rate is lower than normal at times of the day when the hallways of its building are crowded.

Our approach to detect such failures is to conduct a Focused Anomalous Region Optimization (FARO) to actively search for regions where *expectations of normal behavior are not met*. Our target domain is the first example above: we wish to monitor the motion of the CoBot service robot [2] in a 6-dimensional domain. However, for ease of explanation and visualization, we use a simpler 2-dimensional domain, inspired by the golf example above, as a motivating example.

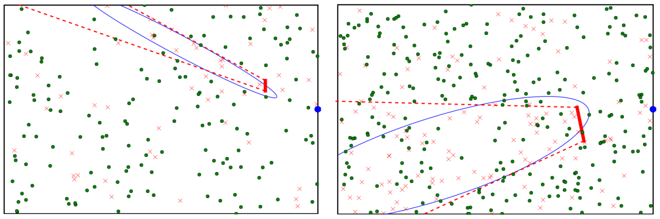

(a) abnormal success rate = 0.2 (b) abnormal success rate = 0.5

Fig. 1: Robot repeatedly aiming at a target on the far right. FARO finds approximations (blue ellipses) to anomalous regions (defined by red lines) where the observed distribution of success (green circles) vs failure (red exes) does not match the expected distribution $P(\text{success}) = 0.8$.

Our motivating example robot, with state $\boldsymbol{x} = [x\ y]^\top \in X$, needs to hit a target from various points in X, as shown in Figure 1. The robot expects to succeed with some known, possibly state-dependent probability $p(\boldsymbol{z}|\boldsymbol{x})$, where $\boldsymbol{z} \in \{0, 1\}$ indicates whether the robot succeeded.

Figure 1 also shows an example of the type of abnormal situation that we wish to detect: In a particular region of the field, the robot's performance is inferior to its expectation. This anomaly could be caused by various problems, such as some unevenness on the field for the putting robot. Regardless of the reason, simply knowing the region in which the anomaly happens gives the robot valuable information; the robot might, for example, choose to avoid the abnormal region if possible.

Formally, our algorithm assumes a robot that receives a sequence of state-observation pairs $(\boldsymbol{x}_t \in X, \boldsymbol{z}_t \in \mathbb{R}^k), t \in \{0, 1, 2, \ldots\}$ during execution. It also assumes this robot has an expectation about what observations $\boldsymbol{z}_t$ it will receive in each state $\boldsymbol{x}_t$ it visits[1], defined by a given distribution $p(\boldsymbol{z}|\boldsymbol{x})$. *We address the problem of early detection of regions $R \subseteq X$ in which the robot's observations deviate significantly from their expectation $\boldsymbol{\mu}(\boldsymbol{x}) \equiv E[\boldsymbol{z}|\boldsymbol{x}]$.* The difficulty in detecting this type of anomaly comes from two main sources:

1) Data from abnormal regions may not happen sequentially in time. In our motivating example, shot location x_{i+1} is independent from shot location x_i. Thus, attempting to detect anomalies as temporal changes in data streams [3] is not effective in our scenarios.
2) The shape of the abnormal region could be complicated. While, given enough data, existing machine

*This research is sponsored by DARPA under agreement FA8750-12-2-0291, by the National Science Foundation under award NSF IIS-1012733, and by AFOSR under grant FA2386-10-14138. The views and conclusions contained in this document are those of the authors only.

[1] Juan Pablo Mendoza is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA jpmendoza@ri.cmu.edu

[2] Manuela Veloso is with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA mmv@cs.cmu.edu

[3] Reid Simmons is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA reids@cs.cmu.edu

[1] The problem of generating observation expectations is beyond the scope of this paper. They could be given by a human expert, or they could be generated from data of past normal execution.

learning methods (e.g., kernel-based [4] or density ratio estimation [5] methods could approximate this region very accurately, this is guaranteed only with large bodies of data. In many applications, it is important to detect anomalies early (i.e., with few data points) to stop or rectify execution.

To address these difficulties, our algorithm *approximates the shape of the region $R_{\max} \subseteq X$ most likely to be abnormal by optimizing over a fixed-parameter family of shapes*. This active optimization enables it to find abnormal regions early, at the cost of only being to approximate the shape of abnormal regions as ell as the chosen family of shapes permits. The algorithm scales tractably with dimensionality, and experiments with synthetic and real robot data show that it can find abnormal regions with high accuracy.

II. Related work

Various fields of research, such as computer security and medical imaging, have extensively studied the problem of detecting failures or anomalies during execution. Approaches to do so can be generally divided into three categories: detecting modeled deviations from normal execution models, detecting unmodeled deviations from normal execution models, and detecting anomalies in data without previous models of either normal or abnormal execution. Which approach is more suited to a problem depends on the availability of models or labeled data from normal and abnormal behavior, and they have all been applied in robotics [1] and other fields [6]. We focus on the problem of detecting abnormal behavior given that a model of normal behavior is available.

We are interested in finding anomalies that cannot be detected from a single data point. Instead, the approach needs to analyze *collections of data* to find anomalies. Several techniques have been used to detect anomalies from collections of data, such as finding anomalous subsequences of time-ordered data [7] or detecting when a particular data point or series is observed more or less frequently than expected [8]. Our approach differs from those in that it is built to detect anomalous data that may happen far apart in time but close along *some* dimensions of state space X, and that may conform to overall frequency expectations, but violates them in a particular region of X (e.g., in our recurring example, the overall probability of success may not deviate significantly from normal, but the probability of success from the anomalous region does).

In some ways, our problem is related to image segmentation: We try to find groups of data points that may not each be special on their own, but whose spatial proximity makes them a significant region (like an object in an image). However, image segmentation algorithms (e.g., [9]) generally require a dense representation of the image, while our problem often presents relatively sparse data in space.

Our algorithm is closely related to the spatial scan statistic algorithm presented in [10], in which a scanning window is used to try to find abnormal regions of data in space. This work has been extended to elongated and rotated regions [11], and to increase scanning efficiency [12]. However, these methods were developed for low-dimensional data, and become intractable with an increase in dimensionality. A scan statistic algorithm linear in the number of data points is presented in [13], but this algorithm requires no spatial relation among the abnormal data points, leading to false positive abnormal region classifications when the anomaly is truly confined to a particular region of state space.

III. Online Detection of Anomalous Regions

The core ideas behind our algorithm are the following: We can find anomalous regions during execution by repeatedly searching for the region $R_{\max}$ most likely to be anomalous, and then deciding whether it is abnormal enough to be declared an anomaly. During execution, this search can be performed tractably by keeping it *focused* in two ways: (1) We restrict the optimization to a fixed-parameter family of shapes, and (2) We initialize the optimization only with a small set of regions $\mathcal{R}$ which are likely to be quickly optimized toward $R_{\max}$. At every time t, $\mathcal{R}$ consists of two regions: the most abnormal-seeming region from $(t-1)$, and a small region $r(\boldsymbol{x}_t)$ surrounding the new data point $\boldsymbol{x}_t$. We choose these two regions because $R_{\max}$ is likely to evolve either from the previous most promising region or from information provided by the new data point.

Algorithm 1 presents the core of our Focused Anomalous Region Optimization (FARO) detector, which is run each time a new data pair is observed. An iterative optimization runs while the algorithm has not exceeded an allowed running time $t_{\max}$ and no anomaly has been found. Every iteration, each region in $\mathcal{R}$ is optimized (i.e. reshaped) into regions that are more likely to be abnormal, based on an anomaly likelihood measure $\operatorname{anom}(R)$, discussed extensively in Section III-A. For each optimized region R', $\operatorname{anom}(R')$ is compared to a threshold value $a_{\max}$. If $\operatorname{anom}(R') > a_{\max}$, R' is returned as anomalous. The algorithm finalizes when an anomalous region is found, or the allowed time runs out.

Algorithm 1 Focused Anomalous Region Optimization.

```
function FARO($(\boldsymbol{x}_i, \boldsymbol{z}_i)_{i \in \{0,\ldots,t\}}, \mathcal{R}, a_{\max}, t_{\max}$)
    $\mathcal{R} \leftarrow \mathcal{R} \cup r(\boldsymbol{x}_t)$
    while running time $< t_{\max}$ do
        for $R \in \mathcal{R}$ do
            $R \leftarrow \operatorname{OptimizerStep}(R)$        ▷ Algorithm 2
            if $\operatorname{anom}(R) \geq a_{\max}$ then
                return $R$ as an anomalous region
            end if
        end for
    end while
    return No anomalous region found
end function
```

Ideally, during each time step, each of the regions in $\mathcal{R}$ would (a) be optimized until convergence, and would (b) converge to a common global optimum. These two goals are unrealistic in most systems, as $\operatorname{anom}(R)$ is often a complicated, nonlinear function that is hard to optimize, and $t_{\max}$

may be short. Initializing the optimization with regions from $\mathcal{R}$ thus allows convergence to happen throughout multiple time steps for the most promising regions.

While in this paper we only keep the most promising region in $\mathcal{R}$ for the next time step, the algorithm generalizes straightforwardly to keeping the k most promising regions around instead, at the cost of having to maintain and optimize those k regions. Using such extension to detect multiple anomalous regions is beyond the scope of this paper, however, as issues such as overlapping regions and loss of diversity in $\mathcal{R}$ would probably need to be addressed.

A. Computing an anomaly measure

To calculate the anomaly value of a region R, we will assume the following: For all the states in region R, the mean of the observations has been shifted by a constant vector $\boldsymbol{\delta}$:

$$E[\boldsymbol{z}|\boldsymbol{x}] = \begin{cases} \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\delta} & \text{if } \boldsymbol{x} \in R \\ \boldsymbol{\mu}(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{1}$$

While other work [14] has used a multiplicative constant, an additive constant serves our purposes best. In many monitoring applications, zero-mean observations (e.g., residuals) indicate normal execution; thus a multiplicative constant would not distinguish between normal and abnormal execution.

We search for the region R that maximizes the ratio of likelihoods $\mathrm{anom}(R) = \frac{P(\mathrm{Data}\,|\,\exists\,\mathrm{Anomaly})}{P(\mathrm{Data}\,|\,\nexists\,\mathrm{Anomaly})}$. Assuming independence between observations given the state of the robot, these probabilities can be broken down into products of individual probabilities $P(\boldsymbol{z}_i|\boldsymbol{x}_i)$. Furthermore, since we do not know beforehand the shift value $\boldsymbol{\delta}$, we use the most likely value for $\boldsymbol{\delta}$, i.e., the one that maximizes the likelihood of the data, as is standard in the literature [3], [14]. We now have a double maximization, over R and $\boldsymbol{\delta}$; however, given a region R, the maximization over $\boldsymbol{\delta}$ can be done analytically, as shown below. The ratio of likelihoods we wish to maximize then becomes

$$\mathrm{anom}(R) \equiv \frac{\max\limits_{\boldsymbol{\delta}} \prod\limits_{\boldsymbol{x}_i \in R} P(\boldsymbol{z}_i|\boldsymbol{\mu}_i + \boldsymbol{\delta}) \prod\limits_{\boldsymbol{x}_i \notin R} P(\boldsymbol{z}_i|\boldsymbol{\mu}_i)}{\prod\limits_{\boldsymbol{x}_i} P(\boldsymbol{z}_i|\boldsymbol{\mu}_i)}$$

$$= \max_{\boldsymbol{\delta}} \frac{\prod\limits_{\boldsymbol{x}_i \in R} P(\boldsymbol{z}_i|\boldsymbol{\mu}_i + \boldsymbol{\delta})}{\prod\limits_{\boldsymbol{x}_i \in R} P(\boldsymbol{z}_i|\boldsymbol{\mu}_i)} \tag{2}$$

The above expression is a general measure of anomaly for arbitrarily shaped regions R and for arbitrary distributions $P(\boldsymbol{z}|\boldsymbol{x})$. To make it more concrete, we now derive the expression for $\mathrm{anom}(R)$ for the Gaussian distribution $P(\boldsymbol{z}_i|\boldsymbol{x}_i) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}_i), \boldsymbol{\Sigma}(\boldsymbol{x}_i)) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. We show the derivation for the Gaussian distribution since it is one of the most commonly used. For example, the data in the robot experiments of Section IV-B is close to Gaussian-distributed, and even the binomial distribution generated by our motivating example can be approximated by a Gaussian distribution. Analogous derivations for other distributions, and for multiplicative shifts (instead of our additive shift $\boldsymbol{\delta}$), can be found in [14].

For Gaussian distributions, it is simpler to work with the logarithm of the likelihood ratio rather than with the likelihood ratio itself. Since logarithm is a monotone function, the region that maximizes one will maximize the other. Defining an auxiliary variable $\Delta\boldsymbol{z}_i \equiv \boldsymbol{z}_i - \boldsymbol{\mu}_i$, function $F(R) \equiv \ln \mathrm{anom}(R)$ is given by [2]

$$= \max_{\boldsymbol{\delta}} \sum_{\boldsymbol{x}_i \in R} \left[\ln(P(\boldsymbol{z}_i|\boldsymbol{\mu}_i + \boldsymbol{\delta}, \boldsymbol{\Sigma}_i) - \ln(P(\boldsymbol{z}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right]$$

$$= \max_{\boldsymbol{\delta}} \sum_{\boldsymbol{x}_i \in R} \frac{1}{2}\left[\Delta\boldsymbol{z}_i^\top \boldsymbol{\Sigma}_i^{-1} \Delta\boldsymbol{z}_i - (\Delta\boldsymbol{z}_i - \boldsymbol{\delta})^\top \boldsymbol{\Sigma}_i^{-1} (\Delta\boldsymbol{z}_i - \boldsymbol{\delta})\right]$$

$$= \max_{\boldsymbol{\delta}} \sum_{\boldsymbol{x}_i \in R} \left[\boldsymbol{\delta}^\top \boldsymbol{\Sigma}_i^{-1} \Delta\boldsymbol{z}_i - \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\delta}\right]$$

$$= \max_{\boldsymbol{\delta}} \left[\boldsymbol{\delta}^\top \sum_{\boldsymbol{x}_i \in R} \left(\boldsymbol{\Sigma}_i^{-1} \Delta\boldsymbol{z}_i\right) - \frac{1}{2}\boldsymbol{\delta}^\top \sum_{\boldsymbol{x}_i \in R} \left(\boldsymbol{\Sigma}_i^{-1}\right) \boldsymbol{\delta}\right] \tag{3}$$

To find the maximizing $\boldsymbol{\delta}$, we differentiate the expression inside the max and equate it to 0 to obtain

$$\boldsymbol{\delta}_{\max} = \left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1}\right)^{-1} \left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1} \Delta\boldsymbol{z}_i\right) \tag{4}$$

Substituting $\boldsymbol{\delta}_{\max}$ back into Equation 3 gives the final expression for the quantity to maximize:

$$F(R) = \frac{1}{2}\left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1} \Delta\boldsymbol{z}_i\right)^{\!\top} \left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1}\right)^{-1} \left(\sum_{\boldsymbol{x}_i \in R} \boldsymbol{\Sigma}_i^{-1} \Delta\boldsymbol{z}_i\right) \tag{5}$$

This expression depends only on sufficient statistics of the observed data and assumed previous knowledge of the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{x}_i)$. This is particularly useful when the state space is discretized and only sufficient statistics of each discrete state, instead of all the data points, need to be stored, as in our implementation (see Section IV).

B. Cross-entropy optimization for maximum anomaly regions

Once equipped with an appropriate anomaly measure, we use the Cross-Entropy Method (CEM) [15] to optimize for the most abnormal parameterized region $R(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^n$ is the parameter vector for R. Algorithm 2 shows a step of CEM in our context. Given $\boldsymbol{\theta}$, and its estimated covariance matrix $\boldsymbol{\Sigma}_\theta$ (initially set to some small covariance $\boldsymbol{\Sigma}_0$), this randomized optimization algorithm re-estimates $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}_\theta$ to a value closer to its optimal anomaly value.

While the original CEM algorithm gives the top k samples (for some constant k) a weight of $w_i = \frac{1}{k}$, and all other samples a weight of 0, other weighing schemes may also be used [16]. Our normalized value function already represents the relative likelihood of different candidate regions, and therefore we use it as the weighing function.

[2] Unfortunately, capital letter Σ is the standard symbol for both summations and covariance matrices. While we have kept this notation, we try to avoid confusion by always placing summation indices under Σ, while covariance matrices are indexed by a subscript to the right of Σ.

Algorithm 2 Cross Entropy Maximization Step

```
function OptimizerStep($R(\boldsymbol{\theta})$)
    Sample $m$ points $\boldsymbol{\theta}_i \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}_\theta)$    ▷ Sample
    $\forall i, w_i \leftarrow \frac{\operatorname{anom}(R(\boldsymbol{\theta}_i))}{\sum_i \operatorname{anom}(R(\boldsymbol{\theta}_i))}$    ▷ Weigh
    $\boldsymbol{\theta} \leftarrow \sum_i w_i \boldsymbol{\theta}_i$    ▷ Re-estimate $\boldsymbol{\theta}$
    $\boldsymbol{\Sigma}_\theta \leftarrow \sum_i w_i (\boldsymbol{\theta}_i - \boldsymbol{\theta})(\boldsymbol{\theta}_i - \boldsymbol{\theta})^\top$    ▷ Re-estimate $\boldsymbol{\Sigma}_\theta$
    return $R(\boldsymbol{\theta})$
end function
```

The CEM algorithm re-estimates both the mean and the covariance matrix of the distribution. This, along with some noise added as a diagonal matrix $\boldsymbol{\Sigma}_\varepsilon$ (as in [16]), allows our algorithm to naturally perform a broader search in portions of X where the value function is close to constant, and a most focused search where the function changes rapidly, requiring a finer optimization. We chose the value of both the first guess of covariance $\boldsymbol{\Sigma}_0$ and the noise matrix $\boldsymbol{\Sigma}_\varepsilon$ to be about 5% of the size of the world along each dimension.

For the examples and experiments in this paper, our chosen parameterized family of shapes were ellipsoids. In an n-dimensional state space, an ellipse can be parameterized by an n-vector $\boldsymbol{v}$ and a $n \times n$ positive definite matrix $\boldsymbol{A}$ as the set of points that satisfy:

$$(\boldsymbol{x} - \boldsymbol{v})^\top \boldsymbol{A}^{-1} (\boldsymbol{x} - \boldsymbol{v}) = 1 \tag{6}$$

For the purposes of optimization, our vector $\boldsymbol{\theta}$ will thus be the linearized form of $\boldsymbol{v}$ and $\boldsymbol{A}$, consisting of $n + \frac{n(n+1)}{2} = \frac{1}{2}\left(n^2 + 3n\right)$ dimensions.

IV. Experiments and Results

To validate the effectiveness of our FARO-based detector, we ran experiments in two domains: synthetic data from the recurring example discussed in previous sections, and real-world data from the CoBot service robots [2]. The objectives of experimenting in these two domains are that the first provides a simple and easily visualizable domain, while the second provides a real-world application with a higher dimensional state space; the two combined support the generality of the algorithm.

To greatly decrease the number of data points to be analyzed, the state space was discretized into a grid. For each cell in this grid, we store only sufficient statistics of the observed data. Discretizing the state space involves a trade-off between efficiency and accuracy in detection.

A. Synthetic Data Experiments

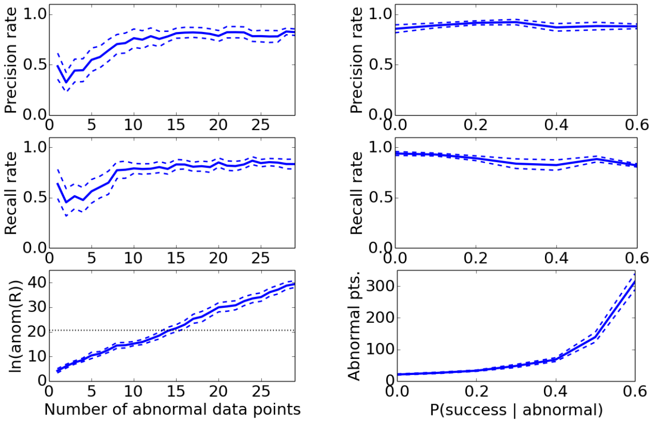
We first show experiments and results using synthetic data from our recurring example. The parameters in this world are the shape and location of the unperceived obstruction obs, the probability p of success during normal execution, and the probability q of success from locations that are blocked by obs (abnormal execution). For each test of each of the experiments below, obs was sampled uniformly from the set of line obstructions where the distance between the target and the center of obs is between 0.1 and 1.5 meters (the field measures 6×4 meters); the center of obs is in the field, and the angle subtended by the endpoints of obs and the target is between $\frac{\pi}{16}$ and $\frac{\pi}{4}$ radians. This randomization created test regions of various shapes, sizes, and orientations.

The goal of the first experiment was to determine the effect of the number of available abnormal data points on the effectiveness of detection. To do this, we held both p and q constant and kept track of the precision and recall rates of the algorithm as the number of abnormal data points increased. In the context of this work, given a ground truth anomalous region S, precision and recall of the estimated region R are defined as

$$\text{precision} = \frac{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in R \cap S\}|}{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in R\}|} \tag{7}$$

$$\text{recall} = \frac{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in R \cap S\}|}{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in S\}|} \tag{8}$$

Figure 2a shows the results of these experiments for $p = 0.8$, $q = 0.2$. Apart from an initially surprising high precision and recall when the number of abnormal data points $n_a = 1$, we see a consistent increase in both as n_a increases. With $n_a \approx 10$, precision and recall both get to about 80%, and they keep slowly increasing. The apparently surprising precision and recall observed for $n_a = 1$ happens because we use $r(\boldsymbol{x}_t)$ as a seed to the optimizer; thus, $\boldsymbol{x}_t$ is often correctly selected as the only abnormal point when $n_a = 1$.



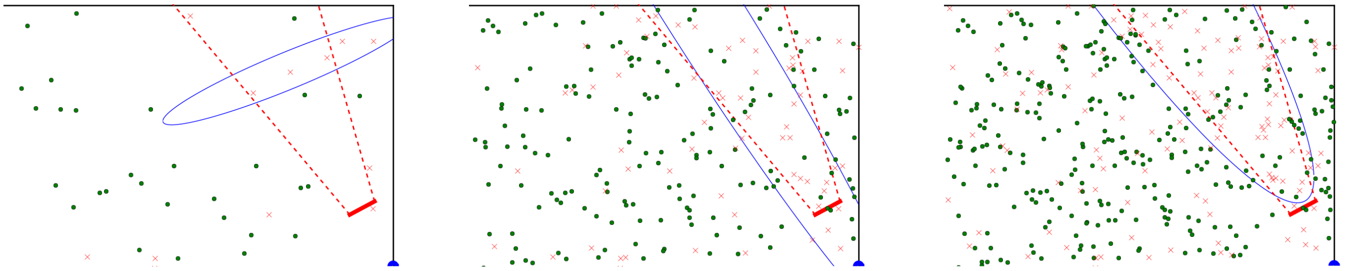
(a) Precision, Recall and anom value as the number of abnormal data points varies.

(b) Precision, Recall and Number of abnormal points when threshold anomaly was passed.

Fig. 2: Synthetic data experiments results. Blue dashed lines indicate standard error bars. The black dashed line is the highest anomaly value observed during normal execution.

Anomaly value $\operatorname{anom}(R)$ in Figure 2a grows exponentially as more abnormal data arrives ($\ln \operatorname{anom}(R)$ grows approximately linearly); with $n_a \approx 15$, the anomaly value surpasses the highest value observed during normal execution. With relatively few data points, then, FARO finds the abnormal regions with high precision and recall.

Figure 3 shows an example of how an ellipse evolves as more data becomes available. Both $\operatorname{anom}(R)$ and the



(a) $\ln \text{anom}(R) = 10.0$, 7 observations (b) $\ln \text{anom}(R) = 20.6$, 43 observations (c) $\ln \text{anom}(R) = 40.8$, 84 observations

Fig. 3: Most abnormal ellipse found as more observations arrive. The sub-figures show the state of the algorithm when three anomaly thresholds are reached. Parameter values are $p = 0.8$, $q = 0.5$.

accuracy of the ellipse approximation increase with the number of abnormal data points. Note that even when the ellipse approximates the true anomaly S very well, as in Figure 3c, precision and recall rates still do not reach 1.0. The first reason for this is that S cannot be arbitrarily-well approximated by any ellipse; the efficiency of choosing a relatively small parameterization (i.e., ellipses) comes with the cost of the inability to represent regions arbitrarily well. Additionally, if there are missed shots outside of S, an ellipse that includes those shots has a higher value than an ellipse that does not, and conversely with successful shots inside of S; thus, *for finite data sets*, the region that maximizes anomaly is not necessarily the one that best matches S.

The goal of the second experiment was to determine the effect of the magnitude of the anomaly on the algorithm's performance. To do this, we analyzed the number of data points required for the detector to reach $\ln(\text{anom}(R)) = 41.6$ (twice as much as the maximum observed during normal execution), as the abnormal distribution q approached the normal execution distribution $p = 0.8$. Figure 2b shows the results of this experiment. The number of data points required to be certain of an anomaly appears to increase exponentially as the anomalous distribution gets closer to the normal distribution. This result is expected as the number should go to infinity as $q \to p$, at which point the distributions are indistinguishable. Furthermore, it is noteworthy that the precision and recall rate stay close to constant for a given anomaly threshold. This suggests that the precision and recall performance of the detector are not heavily dependent of the magnitude of the failure, given an anomaly threshold.

B. Real world experiments: motion anomalies on the CoBot

The CoBot mobile robots autonomously perform tasks for inhabitants of the Gates-Hillman Center at CMU. The CoBots' autonomy is very robust, and it has traversed several hundred kilometers without supervision as it completes online-requested tasks, with very few failures in execution [2]. These rare failures are precisely those we want the robot to be able to detect autonomously, and in these experiments we focus on motion failures. Concretely, we define the motion state of the CoBot by translational and rotational positions and velocities as $\boldsymbol{x} \equiv [x\ y\ \phi\ \dot{x}\ \dot{y}\ \dot{\phi}]^\top$, and let our observations be defined as $\boldsymbol{z} \equiv [\Delta\dot{x}\ \ \Delta\dot{y}\ \ \Delta\dot{v}]$, the vector difference between the CoBot's measured velocity, obtained from its wheel encoders, and its expected velocity, based on its velocity command and its state.

After running the CoBot for about 20 minutes of normal execution, and determining that the highest anomaly value during normal execution of $\ln(\text{anom}(R)) = 20.1$, we proceeded to inject anomalies into its execution and test the detector. We introduced four types of anomalies separately:

Failing encoder

Everywhere in X, one of the CoBot's wheel encoders returns $(1 - \epsilon)d$, at each timestep, where d is the displacement of the wheel that would be returned during normal execution. This failure mode tests FARO under anomalies that happen globally in X.

Collision

During an otherwise normal run of the robot, a sudden collision happens. This failure mode tests the detector under anomalies that are local in X.

Bad corridor

The wheel encoders fail by returning $(1 - \epsilon)d$, as above, but only in a particular corridor of the building. This failure mode tests the detector under failures that happen in a particular region of state space. Note that this region encompasses a particular sub-region of two of the dimensions of X, and it is global in the others.

Bad left turns

The robot's performance is normal except when it turns left (i.e., $\dot{\phi} > 0$), in which case each of its wheels moves only at $(1 - \epsilon)v$ for a commanded velocity v. Since the robot turns only at intersections or when it needs to face a doorway, this failure mode tests FARO when small clusters of abnormal points happen infrequently and far apart.

For each of these experiments, the robot was commanded at a high level to navigate to various offices around the building, using its autonomous navigation algorithms [17], while FARO ran simultaneously. The route the robot took was different each time, as it was simply commanded to go to various points in the building. Table I summarizes the results of running the algorithm under the different failure modes, with $\epsilon = 0.05$.

Experiment	Anomaly Thresh = 40.2			End of experiment		
	Data Points	Precision	Recall	Anomaly	Precision	Recall
Failing encoder	58 ± 8.2	1.0 ± 0.0	0.76 ± 0.15	337.08 ± 14.37	1.0 ± 0.0	0.92 ± 0.03
Collision	4.5 ± 2.05	1.0 ± 0.0	0.81 ± 0.09	1535.05 ± 54.02	1.0 ± 0.0	0.85 ± 0.03
Bad corridor	60 ± 11	0.94 ± 0.02	0.77 ± 0.15	198.50 ± 28.15	0.97 ± 0.01	0.90 ± 0.02
Bad left turn	31 ± 5.1	1.0 ± 0.0	0.79 ± 0.07	202.74 ± 22.04	0.80 ± 0.18	0.47 ± 0.12

TABLE I: CoBot experiment results. For each experiment, the table shows statistics at the time the anomaly threshold (two times the largest anomaly observed during normal execution) was surpassed, and statistics at the time the robot was stopped.

The algorithm shows high precision and recall for each one of the experiments. The variance in precision and recall rates is small but not insignificant, indicating that the difficulty of detecting anomalies in some runs was not route-independent.

The experiment that stands out in terms of results is the *Bad left turn*, for which the recall rate is significantly lower than the others. This is because the anomaly happens in very disjoint regions of state space (only when the robot needs to turn left), with no data between them. This also explains why the final recall in this failure mode is lower than at the time of detection: more disjoint abnormal regions were visited after the time of detection, and some of them were not joined to the main ellipse. This result shows promise for two potential future extensions of our algorithms not addressed in this paper: (1) detection of multiple anomalous regions, and (2) active exploration between potentially abnormal regions to determine whether they are truly two potentially abnormal regions or one large abnormal region.

V. Conclusion

We have presented the FARO optimization-based algorithm for early detection of abnormal regions. The explicit and focused search for a parameterized abnormal shape that best models the data allows the algorithm to detect anomalies of various orientations and scales with few data points.

The algorithm has shown to reliably detect anomalies that happen in convex regions of spaces of up to 6 dimensions. This paper focused on detecting and approximating single failure regions; it is not hard to imagine an extension to the algorithm in which multiple ellipsoids are tracked as candidate regions of anomaly. Problems such as merging of multiple anomalous regions have not been analyzed yet. Further work is thus necessary to draw conclusions about the applicability of our approach to higher dimensions and to multiple anomalies.

As the results in Section IV-B show, the algorithm had some problems finding the full anomaly region when large sub-regions of this region were completely devoid of data. This suggests a research opportunity in the domain of active learning of these anomaly regions, given that the robot has an active role in which states of state space X it visits. For example, a robot may observe potential anomalies when turning left at either end of a corridor, but may have never needed to turn left in the middle; it may then decide to explore turning left in the middle simply to confirm that those anomalies should be merged or denying that hypothesis.

In addition to presenting a detection algorithm, a goal of this paper was to demonstrate the usefulness, within the context of anomaly detection, of actively optimizing for the *shape* of the potential anomalies. For example, kernel-based density or density ratio estimation approaches to anomaly detection [5] usually choose the kernel bandwidth parameters to be independent along the various dimensions, and to maximize the prediction capabilities within the training set of data. Future work could explore the potential advantages of using kernel parameters that actively optimize for anomaly instead, to test the more general applicability this type of focused optimization.

References

[1] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, Nov. 2005.

[2] M. Veloso, J. Biswas, B. Coltin, S. Rosenthal, S. Brandao, T. Mericli, and R. Ventura, "Symbiotic-Autonomous Service Robots for User-Requested Tasks in a Multi-Floor Building," *IROS 2012 Workshop on Cognitive Assistive Systems*, 2012.

[3] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[4] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier Detection with Kernel Density Functions," *Machine Learning and Data Mining in Pattern Recognition*, vol. 4571, pp. 61–75, 2007.

[5] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowledge and Information Systems*, vol. 26, no. 2, pp. 309–336, Feb. 2011.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, no. September, pp. 1–72, 2009.

[7] E. Keogh and J. Lin, "Hot sax: Efficiently finding the most unusual time series subsequence," in *ICDM*, 2005, pp. 226–233.

[8] E. Keogh, S. Lonardi, and B. Chiu, "Finding surprising patterns in a time series database in linear time and space," *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 550, 2002.

[9] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, Nov. 2006.

[10] M. Kulldorff, "A spatial scan statistic," *Communications in Statistics-Theory and methods*, 1997.

[11] M. Kulldorff, L. Huang, L. Pickle, and L. Duczmal, "An elliptic spatial scan statistic." *Statistics in medicine*, vol. 25, no. 22, pp. 3929–43, Nov. 2006.

[12] D. B. Neill and A. W. Moore, "Detecting significant multidimensional spatial clusters," *Advances in Neural Information Processing Systems*, vol. 17, pp. 969–976, 2004.

[13] D. B. Neill, "Fast subset scan for spatial pattern detection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 337–360, Mar. 2012.

[14] ——, "Detection of spatial and spatio-temporal clusters," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 2006.

[15] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and computing in applied probability*, vol. 1, no. 2, pp. 127–190, 1999.

[16] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *ICML*, 2012.

[17] J. Biswas and M. Veloso, "Localization and navigation of the CoBots over long-term deployments," *The International Journal of Robotics Research*, pp. 1–16, 2013.