Article

The International Journal of Robotics Research
1–23

DOI: 10.1177/0278364919845047
journals.sagepub.com/home/ijr
SAGE

Detection and correction of subtle context-dependent robot model inaccuracies using parametric regions

Juan Pablo Mendoza, Reid Simmons and Manuela Veloso

Abstract

Autonomous robots frequently rely on models of their sensing and actions for intelligent decision making. Unfortunately, in complex environments, robots are bound to encounter situations in which their models do not accurately represent the world. Furthermore, these context-dependent model inaccuracies may be subtle, such that multiple observations may be necessary to distinguish them from noise. This paper formalizes the problem of detection and correction of such subtle contextual model inaccuracies in autonomous robots, and presents an algorithm to address this problem. The solution relies on reasoning about these contextual inaccuracies as parametric regions of inaccurate modeling (RIMs) in the robot's planning space. Empirical results from various real robot domains demonstrate that, by explicitly searching for RIMs, robots are capable of efficiently detecting subtle contextual model inaccuracies, which in turn can lead to task performance improvement.

Keywords

Execution monitoring, anomaly detection, online adaptation, model learning

1. Introduction

Robots frequently use models of the behavior of the world to make intelligent decisions throughout execution. Models enable robots to reason about the effects of their actions, and thus to generalize their capabilities to different tasks.

Unfortunately, in many realistic environments, it is infeasible to have the perfect knowledge and exhaustive data required to create globally accurate models. These models, created by some combination of human design and robot experience, may reflect the true dynamics of the world relatively accurately in most circumstances, but fail to capture the dynamics of the world in some specific sets of similar contexts. This work analyzes the need to detect and correct these *contextual model inaccuracies*, and presents an algorithm to do so.

1.1. Illustrative example

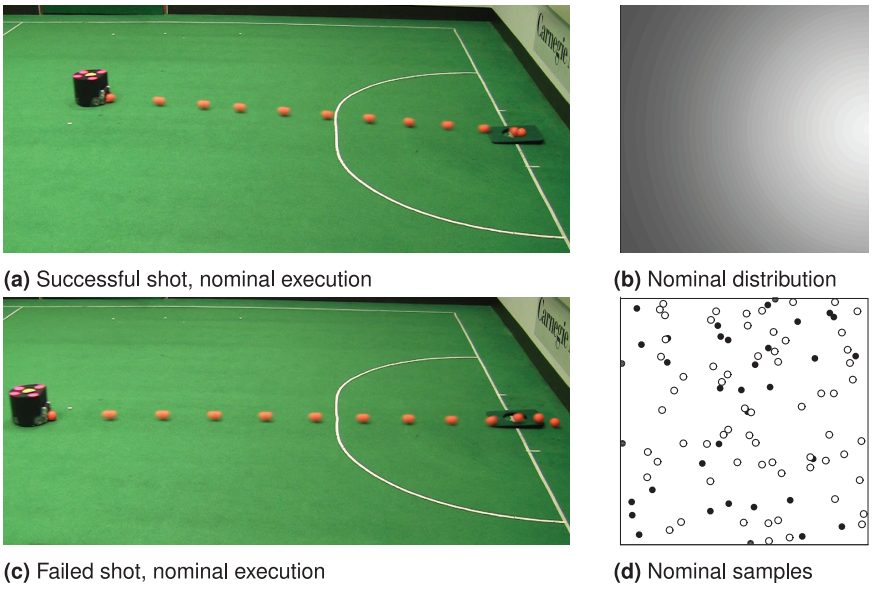
Figures 1 and 2 illustrate the concept of *contextual model inaccuracies* with a simple example: a golf-putting robot needs to repeatedly shoot a golf ball into a hole from different locations on the field. The robot sometimes succeeds and sometimes fails, and it has a model of the distribution of these stochastic outcomes as a function of the shot source location (Figure 1b). However, during one of its deployments, an unexpected and imperceptible bump on the field (Figures 2a and 2c) significantly reduces its chances of successfully putting the ball whenever it shoots from behind the bump (Figure 2b), creating a significant but subtle difference between its model's predictions and the observed execution from behind the bump (Figure 2d). The goal of our work is to enable the robot to autonomously detect these discrepancies between its model and the world, and to adapt its model appropriately. Improving its model enables the robot to make better decisions, e.g., it may choose to shoot from more advantageous locations on the field than from behind the bump.

1.2. General problem characteristics

The simplified golf example illustrates the general class of problems that this work addresses. First and foremost, it

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Corresponding author:
Juan Pablo Mendoza, The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15206, USA.
Email: jpmendoza@ri.cmu.edu



(a) Successful shot, nominal execution

(b) Nominal distribution

(c) Failed shot, nominal execution

(d) Nominal samples

Fig. 1. Golf-putting task. Shots stochastically (a) succeed or (c) fail. (b) The robot builds or receives a nominal model of success probability over the field (lighter shows higher probability). (d) Simulated success and failure samples (white and black circles).

addresses domains in which the robot's model of the world may be partially inaccurate, usually due to practical restrictions: the training environment may not be exactly the same as the deployment environment, or the deployment space may be too large to explore thoroughly during training.

We focus on domains in which, even during nominal execution, the robot's actions produce *stochastic outcomes*; thus, it is necessary to analyze collections of outcome observations to distinguish between nominal stochasticity and model inaccuracies. For example, the golfing robot succeeds or fails stochastically. More generally, we focus on stochastic outcomes that may be discrete or continuous multi-dimensional vectors.

Furthermore, we focus on addressing *context-dependent* model inaccuracies. For example, the golfing robot's context is given by the location on the field from which the robot shoots, and the inaccuracy affects a region of this space; more generally, the context is given by the multi-dimensional space of possible states and actions.

1.3. Approach overview

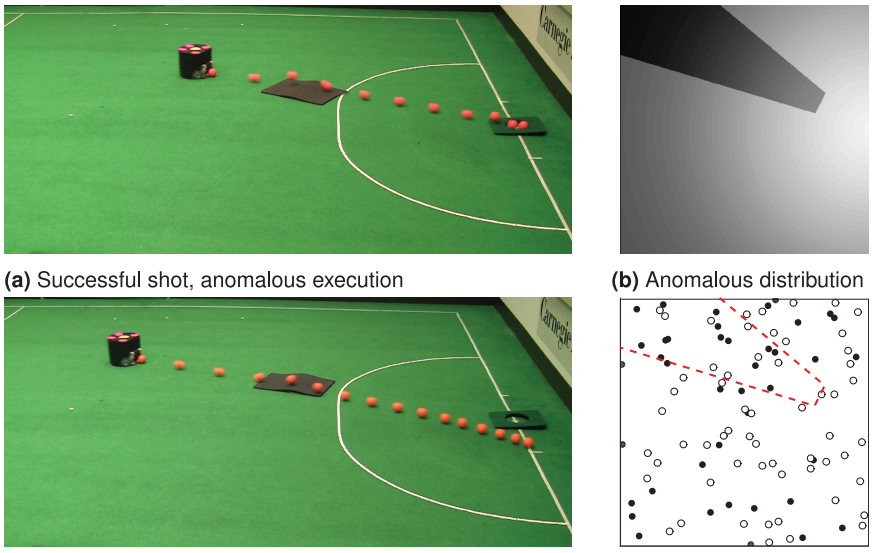
To detect and react to context-dependent model inaccuracies in domains with stochastic outcomes, our approach is to explicitly search for regions of inaccurate modeling (RIMs) in a feature space of the state–action space of the robot. For example, in the golf domain, the robot would be able to determine that the region enclosed by the dashed red lines in Figure 2d is anomalous, and it would correct its model appropriately. There are various RIMs consistent with the observations of Figure 2d; our approach searches over parametric families of regions, e.g., ellipsoids in our experiments, to find the most likely RIMs. These regions are then classified as a true RIMs or not, based on an anomaly score given by the observations contained in them. Finally, the robots correct their models in these RIMs by treating them as context-dependent behavioral modes with different outcome distributions.

1.4. Empirical evaluation

We evaluate our algorithms on the illustrative golf domain as well as two real-robot platforms: the CoBot mobile service robots and the CMDragons team of autonomous soccer robots. In both of these domains, it is infeasible to create perfect models before deployment, because of the unconstrained nature of real world in the case of the CoBot, and because of the diversity of opponent behaviors in the case of the CMDragons.

Empirical results show high detection effectiveness in the golf domain and in the CoBot's motion model, given a variety of injected context-dependent subtle inaccuracies.

In a keep-away domain of the CMDragons playing against themselves, detecting and correcting inaccuracies in the opponent's ball interception model lead to significant improvement in ball-passing performance in timescales comparable with a single soccer game. Remarkably, the algorithm detected various inaccuracies in the CMDragons' models that were not injected, and were previously unknown to the researchers. For example, the robots autonomously learned their opponents could accidentally



(a) Successful shot, anomalous execution

(b) Anomalous distribution

(c) Failed shot, anomalous execution

(d) Anomalous samples

Fig. 2. (a), (c) An imperceptible bump on the field causes a significant inaccuracy in the model of Figure 1b with respect to the true distribution of (b). (d) Synthetic samples behind the imperceptible obstacle show a significant deviation from the nominal model.

(a) CoBot service robots

(b) CMDragons soccer robots

Fig. 3. Robots that serve as motivation and application domains: (a) CoBot service robots; (b) CMDragons soccer robots.

intercept their passes in certain situations, as further described in Section 5.

1.5. Article organization

Section 2 formally introduces the problem of context-dependent model inaccuracy detection and correction, and analyzes the need to solve it through various real robot domain examples. Section 3 situates our work within the execution monitoring, diagnosis, and anomaly-detection communities. Section 4 presents the core theory and algorithms of our approach: RIM -detection and a monitoring framework that uses this information to correct the robot's models; Section 5 presents empirical support for our proposed approach, by applying it to real robot domains. Finally, Section 6 summarizes the contributions and discusses directions for future work.

2. Problem motivation and formulation

This section motivates the need to address the problem of detecting and reacting to subtle context-dependent model inaccuracies with various real robot domains (Section 2.1) and formalizes the problem (Section 2.2).

2.1. Motivating domains

2.1.1. CoBot motion. Autonomous navigation is a key component of all mobile autonomous robots, and thus having an accurate motion model is essential for task performance. We therefore explore the problem of context-dependent model inaccuracies in the motion of the CoBot autonomous service robots (Veloso et al., 2012) of Figure 3a. The CoBots autonomously perform tasks for the inhabitants of multiple buildings at Carnegie Mellon University

(a) Collision with imperceptible obstacle **(b)** Rough tile floor

Fig. 4. Example sources of inaccurate modeling for the CoBot robot's motion model: (a) collision with imperceptible obstacle; (b) rough tile floor.

(CMU). To accomplish high-level tasks, each CoBot navigates autonomously around the building, which entails moving and localizing properly (Biswas and Veloso, 2013).

To move properly, each CoBot uses a model that maps a desired velocity $\mathbf{v}_{\text{cmd}}$ into currents to each of its four wheels. Then, after a short latency time Δt, the CoBot can obtain feedback about its actual observed velocity $\mathbf{v}_{\text{obs}}$ motion from its wheel encoder sensors. Even during nominal execution, these two velocities will not match exactly, owing to noise in actuation and sensing; this noise is approximately normally distributed $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Thus, the relationship between the CoBot's commanded velocity and its measured velocity is given by

$$\boldsymbol{v}_{\text{obs}}(t) = \boldsymbol{v}_{\text{cmd}}(t - \Delta t) + \boldsymbol{\epsilon} \tag{1}$$

Although this simple model accurately reflects reality in a large majority of situations, there are particular contexts in which it fails to do so. Figure 4a shows an infrequent scenario: the CoBot's sensors cannot perceive a table in the environment, which leads to a collision. Although the collision cannot be avoided, it can be quickly detected by sharp inaccuracies in the CoBot's motion model predictions. Furthermore, if the table's position is permanent, the CoBot should learn that, in that particular place of the building, it cannot move correctly.

Figure 4b shows a more subtle model inaccuracy: in particular regions of the Gates–Hillman Center (GHC) a rough tile floor causes small disturbances in the CoBot's motion. Although these disturbances may be too small to detect from a single run, the CoBot could learn over multiple runs that, in those particular regions of the building, when it has a non-zero velocity (and especially at high speeds), its motion model is somewhat inaccurate.

Our goal with the CoBot robot is to enable it to autonomously detect such subtle context-dependent inaccuracies in its motion model. Although the CoBot experiments in this article focus on inaccuracy detection rather than correction, we argue that correcting the motion model could straightforwardly lead to improved performance: the CoBot could autonomously learn to avoid regions of its building where its motion model is inaccurate, thus improving its performance reliability.

2.1.2. Opponent modeling in autonomous robot soccer. Figure 3b shows the CMDragons team of autonomous soccer-playing robots (Mendoza et al., 2016b). These robots participate in the RoboCup Small Size League (SSL), competing against other teams of soccer robots from institutions around the world. Each game of robot soccer lasts 20 minutes, and each opponent team is usually only faced once.

Research on the adversarial nature of the SSL has covered a wide spectrum of topics, from overall team architecture (Browning et al., 2005) to coordinating multiple robots in the presence of opponents (Mendoza et al., 2016a). Here, we focus on the particularly difficult problem of adapting the model of our opponents within a single game. To be successful, our team needs to have generally accurate models of the opponent from the beginning of the game, as state-of-the-art techniques for learning from scratch need millions of observations to learn to play well (e.g. Stone et al., 2005b). Thus, we usually assume some reasonable opponent model, e.g., they intercept a moving ball the same way our team would.

Inevitably, some of our assumptions about the opponents are wrong, which often leads to context-dependent inaccuracies. The opponents may show strengths and

weaknesses that we could not foresee before the game. For example, their robots might use a ball-interception skill that our robots do not possess, but which is only applicable in particular world configurations; on the other hand, their defense might be particularly weak against some type of attack. Both of these would cause our models in particular sets of similar contexts to be significantly inaccurate. Furthermore, correcting these can lead to significant performance improvement, e.g., a more accurate model of the opponent's interception capabilities leads to better decision making when our robots choose among different possible passes.

Both the CMDragons and the CoBots are domains with stochastic outcome observations, and subtle context-dependent model inaccuracies. Section 2.2 formalizes these domains and the problem we address.

2.2. Problem formulation

In our problem, a robot ρ needs to complete tasks in a continuous state space $\boldsymbol{S} \subseteq \mathbb{R}^{d_s}$, by choosing from a space of actions $\boldsymbol{A} \subseteq \mathbb{R}^{d_a}$ to apply on the world. Applying action $\boldsymbol{a} \in \boldsymbol{A}$ in state $\boldsymbol{s} \in \boldsymbol{S}$ yields some outcome $\boldsymbol{z} \in \boldsymbol{Z}$, according to some unknown probability distribution $P^*(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a})$. The domain $\boldsymbol{Z}$ of these outcomes may vary depending on the application: If the robot's behavior were modeled as a Markov decision process (MDP) (Bellman, 1957), for example, $\boldsymbol{z} \in \boldsymbol{Z}$ would be the state $\boldsymbol{s}'$ resulting from applying action $\boldsymbol{a}$ in state $\boldsymbol{s}$; alternatively, if the robot's behavior were modeled as a contextual bandit problem (Slivkins, 2014), $\boldsymbol{z}$ would be the reward observed when applying action $\boldsymbol{a}$ in state $\boldsymbol{s}$. Our detection model applies to both of these formulations as well as others, so we choose the formulation-agnostic term *outcome* $\boldsymbol{z}$.

The robot has a model $\boldsymbol{\theta}^0$ of the outcome distribution; ideally, the distribution $P(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^0)$ generated by this model would accurately reflect the true distribution for all states and actions: $\forall(\boldsymbol{s}, \boldsymbol{a}) \in (\boldsymbol{S} \times \boldsymbol{A}).P(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^0) = P^*(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a})$. However, it is often the case that a model is not a good predictor for every state and action. In particular, we are interested in domains in which there is some subset $\mathcal{R}_{\mathrm{SA}}$ of the state–action space in which this model $\boldsymbol{\theta}^0$ does not accurately describe execution:

$$\exists \mathcal{R}_{\mathrm{SA}} \subseteq (\boldsymbol{S} \times \boldsymbol{A}).\left[(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{R}_{\mathrm{SA}} \leftrightarrow P(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^0) \not\approx P^*(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a})\right] \tag{2}$$

To maximize the generality of our approach, we define this inaccurate subset over *contextual features* extracted from the state–action space by a function $\chi : \boldsymbol{S} \times \boldsymbol{A} \to \mathcal{X} \subseteq \mathbb{R}^d$. In this work, we focus on domains in which this function is given to the robot, and thus instead of directly finding $\mathcal{R}_{\mathrm{SA}}$, we seek to find $\mathcal{R}^* \subseteq \mathcal{X}$ such that

$$\chi(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{R}^* \leftrightarrow P(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^0) \not\approx P^*(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a}) \tag{3}$$

Table 1. Notation commonly used in this paper.

Notation	Meaning
ρ	Robot
$\boldsymbol{s} \in \boldsymbol{S}$	Robot state
$\boldsymbol{a} \in \boldsymbol{A}$	Robot action
$\boldsymbol{x} \in \mathcal{X}$	State–action context
$\chi : \boldsymbol{S} \times \boldsymbol{A} \to \mathcal{X}$	Context feature-extracting function
$\boldsymbol{z} \in \boldsymbol{Z}$	Outcome observation
$\boldsymbol{z}^{\boldsymbol{x}} = (\boldsymbol{x}, \boldsymbol{z}) \in \boldsymbol{Z}^{\boldsymbol{x}}$	Contextual observation
$\boldsymbol{\theta}^0 \in \boldsymbol{\Theta}$	Nominal behavior model
$\boldsymbol{\theta}^+ \in \boldsymbol{\Theta}$	Corrected model
$P^*(\boldsymbol{z}\|\boldsymbol{x})$	True (unknown) outcome observation distribution
$P(\boldsymbol{z}\|\boldsymbol{x}, \boldsymbol{\theta} \in \boldsymbol{\Theta})$	Observation distribution according to model $\boldsymbol{\theta}$

We focus on cases in which $\mathcal{R}^*$ is made up of a *finite set of regions* of state–action space. Many real-world domains have this characteristic, as motivated in Section 2.1.

Thus, the problem is defined as follows.

Given a list of contextual observations $\boldsymbol{Z}^{\boldsymbol{x}} = [(\boldsymbol{x}_t, \boldsymbol{z}_t)|t = 0, \ldots, N]$, and a model $\boldsymbol{\theta}^0$ of nominal execution

Detect the existence of a set of contexts $\mathcal{R}^*$ as defined in Equation (3), and

Correct Model $\boldsymbol{\theta}^0$ into $\boldsymbol{\theta}^+$, such that

$$\forall(\boldsymbol{s}, \boldsymbol{a}) \in (\boldsymbol{S} \times \boldsymbol{A}).[P(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^+) \approx P^*(\boldsymbol{z}|\boldsymbol{s}, \boldsymbol{a})] \tag{4}$$

Table 1 summarizes the most commonly used notation throughout this paper.

3. Related work

Our work is concerned with online detection and adaptation to subtle context-dependent model inaccuracies. Owing to its online monitoring of the robot's performance, we situate it among the execution monitoring community in Section 3.1. Owing to the contextual nature of the robot's observations, and our desire to detect anomalous sets of observations, we situate it among the anomaly detection community in Section 3.2.

3.1. Execution monitoring

The problem of execution monitoring, also called fault detection and identification (FDI), or diagnosis (DX), depending on the community (Cordier et al., 2004), is concerned with detecting, identifying, and recovering from failures in execution. Execution monitoring is a well-established problem in various areas of scientific research, and the complex and unpredictable nature of robotics domains has led to increased exploration of the subject in robotics (Antonelli, 2003; Pettersson, 2005).

Execution monitoring can be divided into *model-based* methods, which achieve monitoring using models of the system, and *model-free* methods, which detect failures

using only observed data (Hwang et al., 2010). We are interested in domains in which robots have access to partially accurate models of the world, and thus we focus on *model-based* monitoring; however, model-free methods have also been successfully applied to robotics (Pettersson et al., 2003, 2005).

We are interested in the problem of detecting contextual model inaccuracies given a list of stochastic contextual observations in which the robot's state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$ can be mapped into a context feature point $\boldsymbol{x}_t$, and the outcome $\boldsymbol{z}_t$ is the observation to be monitored. Most work in execution monitoring has focused on fault detection given a single observation $\boldsymbol{z}_t$ or a sequence of observations $[\boldsymbol{z}_i | i = 0, 1, \ldots, t]$, i.e., using only *time* as a context that correlates various observations. Several algorithms have been developed to address this problem, and their properties, such as speed of detection and detection power, have been studied extensively.

Two of the most studied algorithms, sequential probability ratio test (SPRT) (Wald, 1945) and cumulative sum control charts (CUSUM) (Page, 1954), detect faults using thresholds on the likelihood ratio of residual observations, given a nominal model $\boldsymbol{\theta}^0$, and an alternative fault model $\boldsymbol{\theta}^1$. These algorithms can be very efficiently computed by maintaining an aggregate statistic S_t and updating it in constant time with a new observation $\boldsymbol{z}_t$. However, they require prior knowledge about the fault model $\boldsymbol{\theta}^1$, or sets of models $\{\boldsymbol{\theta}^i\}$ for multiple fault detection (Nikiforov, 1995). In contrast, we are interested in problems in which the anomalous distributions are not known a priori.

The generalized likelihood ratio test (GLRT) approach (Willsky and Jones, 1976) uses the maximum likelihood estimate of $\boldsymbol{\theta}^1$ for detection of faults with unknown parameters. Faults are again detected by thresholding a statistic S_t. However, in general domains S_t requires computation that is not constant, but linear in t.

In our approach, we seek to use similar statistical techniques as these well-established methods. However, the key difference between our work and these and other time-series monitoring approaches (Bu et al., 2007; Keogh et al., 2005, 2002) is that we need to detect inaccuracies that occur in particular regions of context space. Thus, we must use methods that consider the full contextual observation $(\boldsymbol{x}_t, \boldsymbol{z}_t)$, rather than just the outcome $\boldsymbol{z}_t$.

3.2. Anomaly detection

The anomaly detection community has extensively explored the problem of finding anomalies in contextual data.

According to a classification proposed in a previous survey of anomaly detection research (Chandola et al., 2009), some of the important characteristics of our approach are: (i) it works on multi-dimensional continuous context data; (ii) it is semi-supervised, i.e., it assumes that either a nominal model or nominal execution data is given; and (iii) apart from detecting anomalies, our approach also returns a measure of confidence on that detection. However, the main distinguishing characteristic of our work, according to this classification, is that it detects *spatial collective* anomalies, i.e., anomalies that occur in regions of context space, and which require collections of data for detection, rather than individual points.

The problem of detecting spatial collective anomalies has received significant attention from the computer vision (CV) community, as it can be used to detect anomalous regions of images, or simply to segment regions that stand out. Unfortunately, the algorithms developed for CV are not directly applicable to our problem: images provide dense observations, in the sense that every pixel in the image provides an observation that can be used for detection. Thus, CV techniques often exploit the lattice structure of image pixels to use graph-based algorithms to extract anomalous regions of the image (e.g., Felzenszwalb and Huttenlocher, 2004; Shi and Malik, 2000). Thus, we cannot apply these techniques for our continuous domains with potentially sparse observations. Furthermore, because CV is highly focused on 2D or 3D images, CV algorithms are usually not tractably applicable to problems of higher dimensions, such as our robotics problems.

3.3. Spatial scan statistics

The spatial scan statistic (Kulldorff, 1997) is an approach for detecting regions of a multi-dimensional point process in which the number of observed points is significantly different from the number expected from a given model. This statistic has a wide range of applications, from forestry to astronomy (Kulldorff, 1997); however, it has been most often studied in the context of early disease outbreak detection. The core idea of the algorithm is to search over a set of regions of the process domain to find the region $R_{\max}$ that, given the set of contextual observations $\boldsymbol{Z}^{\boldsymbol{x}}$ and the expected nominal model $\boldsymbol{\theta}^0$, maximizes the following likelihood ratio $\mathrm{anom}(R | \boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0)$:

$$\mathrm{anom}(R | \boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0) = \frac{\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} P(\boldsymbol{Z}^{\boldsymbol{x}}(R) | \boldsymbol{\theta})}{P(\boldsymbol{Z}^{\boldsymbol{x}}(R) | \boldsymbol{\theta}^0)} \tag{5}$$

where $\boldsymbol{\Theta}$ is the space of distribution parameters and $\boldsymbol{Z}^{\boldsymbol{x}}(R)$ are the set of observations seen in R. This approach searches for the region $R_{\max}$ most likely to be anomalous, after which it can perform inference to determine whether there exists an anomalous region of the domain, and where the anomaly is located. As originally developed (Kulldorff, 1997), the algorithm for searching over the space of possible subspaces R is not scalable to higher dimensions, because it performs an exhaustive search over the space of circular subspaces to find the one most likely to be anomalous; this search algorithm is feasible for the original context of the algorithm, in which only a 2D space had to be searched.

More recent work has extended the spatial scan statistics approach in several directions. A more efficient search algorithm has been proposed for axis-aligned rectangles

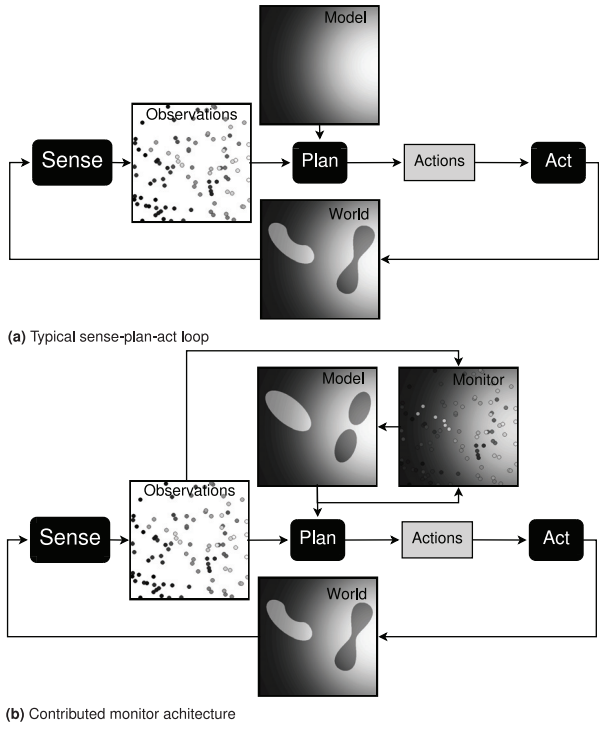


Fig. 5. Robot architecture diagrams. (a) Typical sense–plan–act loop: without monitoring, model inaccuracies persist through execution. (b) Contributed monitor architecture: our monitor detects inaccuracies through an execution monitor, and corrects the robot's model approximately with parametric RIMs.

(Neill et al., 2004), but it does not scale well with the dimensionality of the domain. Graph-based approaches (Duczmal and Assuncao, 2004; Tango and Takahashi, 2005) assume some connectivity between the data, which cannot be easily obtained from sparse observations in higher dimensions. Fast Subset Scan (Neill, 2012; Neill et al., 2013), while efficient, limits its search to only subsets of regions of fixed radius around each observation. In contrast, our method searches over regions with arbitrary locations, sizes, and orientations.

3.4. Reinforcement learning

The model-correction problem addressed in this article is closely related to model-based reinforcement learning (RL): even though the problem of exploration lies beyond the scope of this article (but is addressed in separate research (Mendoza, 2017)), this article presents an approach to correcting models to improve performance, which is the goal of model-based RL. Traditional model-based RL approaches such as the E3 (Kearns and Singh, 2002) and R-MAX (Brafman and Tennenholtz, 2003) algorithms guarantee near-optimal RL in polynomial time in the case of finite states and actions. However, they do not address the problem of generalization to unseen state–actions, and therefore are not applicable to our domains of interest, in which only sparse observations are available. Other algorithms for model-based RL have addressed the problem of generalization, usually via function approximation techniques (Hester and Stone, 2013; Jong and Stone, 2007; Nouri and Littman, 2009). In contrast to these algorithms, we seek to leverage knowledge about the fact that our model is accurate with the exception of a few unknown subspaces of the domain; this knowledge may enable our algorithms to achieve higher data efficiency than previous algorithms.

The problem of Multiple Model-based RL (MMRL) of the world has been explored by previous work, mostly for cases in which the number of different models is known a priori (Choi et al., 2001; Doya and Samejima, 2002). Reinforcement Learning with Context Detection (RL-CD) (da Silva et al., 2006) addresses a similar problem to ours, in that they seek to automatically generate new models throughout execution while performing RL. However, the switch among their models is temporal, in which only one model is active throughout space at a time. In contrast, we wish to infer spatial regions of different behavior, and create new models accordingly. Thus, the novelty of our approach with respect to RL is the discovery of new modes of behavior in state–action space as regions of anomalous behavior, and the active characterization of the extent and behavior of these modes, with the end goal of task performance optimization.

4. Approach: search for, detect and correct RIMs

This section presents our approach to the problem of detecting and correcting subtle context-dependent model inaccuracies. Our approach explicitly searches for parametric RIMs in the robot's context space, during execution. The overall algorithm results in an execution monitoring framework that enables robots to detect and correct RIMs online. At a very high level, the algorithm is sketched in Figure 5. Normally, robots may make decisions based on a partially inaccurate model of the world (Figure 5a). With our approach, a monitor compares observations to expectations extracted from the model; this monitoring can then be used to correct the model through the parametric RIMs found (Figure 5a).

Algorithm 1 shows a description of this monitoring process, extended from our previous work (Mendoza et al., 2015). At every time step t of execution, this monitor receives as input the nominal model of execution $\boldsymbol{\theta}^0$, the state $\boldsymbol{s}_t$ visited by the robot, the action $\boldsymbol{a}_t$ taken by the robot, and the observed outcome $\boldsymbol{z}_t$ of that action. First, the algorithm extracts relevant features from state–action space (line 2). In this work, we assume the feature extraction function χ is given to the robot. Next, the contextual observation $(\boldsymbol{x}_t, \boldsymbol{z}_t)$ is added to the set $\boldsymbol{Z}^{\boldsymbol{x}}$ of all observations (line 3). The monitor proceeds to find RIMs from $\boldsymbol{Z}^{\boldsymbol{x}}$ and $\boldsymbol{\theta}^0$ (line 4), and correct the model accordingly (line 5).

Algorithm 1. Execution monitor procedure run every time step t of execution.

Input: model $\boldsymbol{\theta}^0$ of nominal execution, world state $\boldsymbol{s}_t$, chosen action $\boldsymbol{a}_t$, and observed outcome $\boldsymbol{z}_t$.
Output: corrected model $\boldsymbol{\theta}^+$.
1:**function** MONITOR ($\boldsymbol{\theta}^0, \boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{z}_t$)
2: $\boldsymbol{x}_t \leftarrow \chi(\boldsymbol{s}_t, \boldsymbol{a}_t)$ ▷ Extract relevant features
3: $\boldsymbol{Z}^{\boldsymbol{x}} \leftarrow \boldsymbol{Z}^{\boldsymbol{x}} \cup (\boldsymbol{x}_t, \boldsymbol{z}_t)$ ▷ Update set of observations (initially, $\boldsymbol{Z}^{\boldsymbol{x}} = \emptyset$)
4: $\mathcal{R} \leftarrow$ FindAnomalies($\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0$) ▷ Find anomalies
5: $\boldsymbol{\theta}^+ \leftarrow$ UpdateModel($\boldsymbol{\theta}^0, \mathcal{R}$) ▷ Correct model
6:**end function**

The core of the algorithm thus lies in detecting and correcting model inaccuracies. We divide this problem into four components.

Measure of inaccuracy: Intuitively, we wish to detect and correct regions of context space in which observations do not match the robot's model. Because observations are stochastic, we define this measure statistically: we use a *likelihood ratio* measure, widely used in statistics and anomaly detection (Chan-dola et al., 2009). Section 4.1.1 describes the computation of this measure.

Search space of RIMs: We would like to search over all possible regions of state space to find inaccurate ones. Although some non-parametric methods can in fact represent arbitrarily-shaped boundaries, here we opt for parametric approaches, which converge to meaningful results from sparser observations, at the cost of expressiveness power. Section 4.1.2 describes the parametric space we use.

RIM search algorithm: Given a search measure and space, we define the algorithm for searching over this space. Here, we diverge from previous related work, which conducts exhaustive searches, and take an optimization-based approach, better suited for robotics domains. Section 4.2 describes the focused anomalous region optimization (FARO) (Mendoza et al., 2014) algorithm for domains with up to one RIM, and Section 4.3 describes the DMAPS (Mendoza et al., 2015) extension for domains with multiple RIMs.

RIM correction: Once RIMs are detected, our algorithm corrects these inaccuracies by treating each detected RIM as an alternate behavioral mode of the system, with a different outcome observation distribution Section 4.4 describes how we address this in our work (Mendoza et al., 2015).

4.1. Background: inaccuracy measure and search space

The problem of detecting RIMs has been explored in non-robotics contexts, such as disease outbreak detection. From this previous work (Kulldorff, 1997), we borrow the measure used to decide whether a RIM is truly inaccurate, and the idea of using a particular family of parametric regions as the search space. However, here we derive the specifics to fit our problem domains.

4.1.1. Inaccuracy measure anom($R|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0$). Following previous work, we define a RIM as a region R such that the *likelihood of the data* observed in that region *given the robot's model* is much lower than the likelihood of it *given the best fitting model*. Thus, we use a *likelihood ratio* to define the inaccuracy measure[1] anom($R|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0$) of a region R given a set of observations $\boldsymbol{Z}^{\boldsymbol{x}} = (\boldsymbol{x}_i, \boldsymbol{z}_i)$ and a nominal model $\boldsymbol{\theta}^0$, as given in Equation (5). This statistical measure has advantageous detection properties, described in detail in previous work (Kulldorff, 1997), including a variant of the uniformly most powerful property, and its independence from data that lies outside of R.

This inaccuracy measure depends only on the observations $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ contained in R, and not explicitly on the shape of R itself. This implicitly assumes that we have no prior information about the likelihood of observing RIMs of particular shapes, which is a valid assumption in many robotics domains. Adding such prior information is possible, as shown in Section 4.3.

Assumptions. To compute the anomaly value of a region, we make two assumptions.

- We assume a particular set of alternate models $\hat{\boldsymbol{\theta}}$: for all the states in a RIM, the mean $\boldsymbol{\mu}$ of the observations has been shifted by an unknown constant vector $\boldsymbol{\delta}$

$$\mathbb{E}\left[\boldsymbol{z}|\boldsymbol{x}, \hat{\boldsymbol{\theta}}\right] = \boldsymbol{\mu}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}) = \begin{cases} \boldsymbol{\mu}(\boldsymbol{x}|\boldsymbol{\theta}^0) + \boldsymbol{\delta} & \text{if } \boldsymbol{x} \in R \\ \boldsymbol{\mu}(\boldsymbol{x}|\boldsymbol{\theta}^0) & \text{otherwise} \end{cases} \quad (6)$$

Although other work (e.g., Neill, 2006) assumes a multiplicative offset, an additive offset fits our domains best. In many monitoring applications, zero-mean observations (e.g., residuals) indicate normal execution; thus, a multiplicative constant would not distinguish between normal and anomalous execution. This assumption results in a simple analytical form for Equation (5).[2]

- We assume independence between observations $\boldsymbol{z}_i$ given the context $\boldsymbol{x}_i$. This assumption, which is valid given an expressive enough context description, allows us to decompose the probabilities of Equation (5) into individual probabilities $P(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta})$.

With these assumptions, Equation (5) becomes

$$\text{anom}(R|\boldsymbol{Z}^{\boldsymbol{x}},\boldsymbol{\theta}^0)\equiv\frac{\max_{\boldsymbol{\delta}}\prod_{\boldsymbol{x}_i\in R}P(\boldsymbol{z}_i|\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}^0)+\boldsymbol{\delta})\prod_{\boldsymbol{x}_i\notin R}P(\boldsymbol{z}_i|\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}^0))}{\prod_{\boldsymbol{x}_i}P(\boldsymbol{z}_i|\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}^0))}$$
$$=\frac{\max_{\boldsymbol{\delta}}\prod_{\boldsymbol{x}_i\in R}P(\boldsymbol{z}_i|\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}^0)+\boldsymbol{\delta})}{\prod_{\boldsymbol{x}_i\in R}P(\boldsymbol{z}_i|\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}^0))} \tag{7}$$

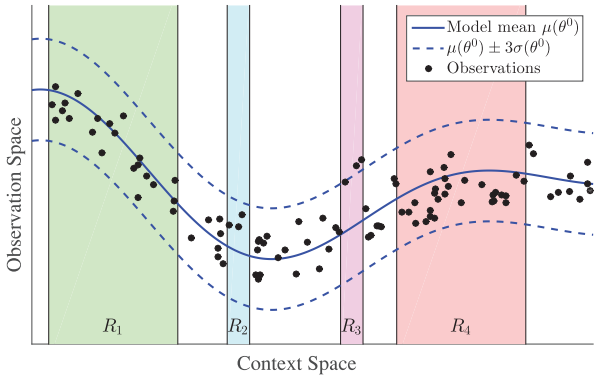
The above expression is a general measure of anomaly for arbitrarily shaped regions R and for arbitrary distributions $P(\boldsymbol{z}|\boldsymbol{x})$. To make it more concrete, we now derive the expression for $\text{anom}(R|\boldsymbol{Z}^{\boldsymbol{x}},\boldsymbol{\theta}^0)$ for the normal distribution $P(\boldsymbol{z}_i|\boldsymbol{x}_i,\boldsymbol{\theta})\sim\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}),\boldsymbol{\Sigma}(\boldsymbol{x}_i|\boldsymbol{\theta}))$. We show the derivation for the normal distribution because it is one of the most commonly used. Analogous derivations for other distributions, and for multiplicative shifts (instead of our additive shift $\boldsymbol{\delta}$), can be found in related work (Neill, 2006). For brevity, when discussing the nominal model $\boldsymbol{\theta}^0$, we use the abbreviations $\boldsymbol{\mu}_i\equiv\boldsymbol{\mu}(\boldsymbol{x}_i|\boldsymbol{\theta}^0)$, $\boldsymbol{\Sigma}_i\equiv\boldsymbol{\Sigma}(\boldsymbol{x}_i|\boldsymbol{\theta}^0)$.

For normal distributions, it is mathematically simpler to work with the logarithm of the likelihood ratio rather than with the likelihood ratio itself. As logarithm is monotone, the region that maximizes one maximizes the other. Defining an auxiliary variable $\Delta\boldsymbol{z}_i\equiv\boldsymbol{z}_i-\boldsymbol{\mu}_i$, function $F(R|\boldsymbol{Z}^{\boldsymbol{x}},\boldsymbol{\theta}^0)\equiv\ln\text{anom}(R|\boldsymbol{Z}^{\boldsymbol{x}},\boldsymbol{\theta}^0)$ is given by[3]

$$\begin{aligned}
F(R|\boldsymbol{Z}^{\boldsymbol{x}},\boldsymbol{\theta}^0)&=\max_{\boldsymbol{\delta}}\ln\left(\frac{\prod_{\boldsymbol{x}_i\in R}P(\boldsymbol{z}_i|\boldsymbol{\mu}_i+\boldsymbol{\delta})}{\prod_{\boldsymbol{x}_i\in R}P(\boldsymbol{z}_i|\boldsymbol{\mu}_i)}\right)\\
&=\max_{\boldsymbol{\delta}}\sum_{\boldsymbol{x}_i\in R}[\ln(P(\boldsymbol{z}_i|\boldsymbol{\mu}_i+\boldsymbol{\delta},\boldsymbol{\Sigma}_i)-\ln(P(\boldsymbol{z}_i|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)]\\
&=\max_{\boldsymbol{\delta}}\sum_{\boldsymbol{x}_i\in R}\left[\ln\left(\exp\left(-\frac{1}{2}\left((\Delta\boldsymbol{z}_i-\boldsymbol{\delta})^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}(\Delta\boldsymbol{z}_i-\boldsymbol{\delta})\right)\right)\right)-\right.\\
&\left.\ln\left(\exp\left(-\frac{1}{2}\left(\Delta\boldsymbol{z}_i^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i\right)\right)\right)\right]\\
&=\max_{\boldsymbol{\delta}}\sum_{\boldsymbol{x}_i\in R}\frac{1}{2}\left[\Delta\boldsymbol{z}_i{}^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i-(\Delta\boldsymbol{z}_i-\boldsymbol{\sigma})^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}(\Delta\boldsymbol{z}_i-\boldsymbol{\delta})\right]\\
&=\max_{\boldsymbol{\delta}}\sum_{\boldsymbol{x}_i\in R}\left[\boldsymbol{\delta}^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i-\frac{1}{2}\boldsymbol{\delta}^{\mathsf{T}}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\delta}\right]\\
&=\max_{\boldsymbol{\delta}}\left[\boldsymbol{\delta}^{\mathsf{T}}\sum_{\boldsymbol{x}_i\in R}\left(\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i\right)-\frac{1}{2}\boldsymbol{\delta}^{\mathsf{T}}\sum_{\boldsymbol{x}_i\in R}\left(\boldsymbol{\Sigma}_i^{-1}\right)\boldsymbol{\delta}\right]
\end{aligned} \tag{8}$$

To find $\boldsymbol{\delta}$ that maximizes this likelihood ratio we find the point where the derivative with respect to $\boldsymbol{\delta}$ is equal to 0:

$$\begin{aligned}
\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i\right)-\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\right)\boldsymbol{\delta}_{\max}&=\quad 0\\
\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\right)^{-1}\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i\right)&=\quad\boldsymbol{\delta}_{\max}
\end{aligned} \tag{9}$$



(a) Nominal model θ^0, observations received, and four test regions

Region	Ground truth (deviation δ)	Num. data points	Observed deviation	$\ln(\text{anom}(R_i))$
R_1	Nominal ($\delta=0$)	22	-0.4σ	3.3
R_2	Nominal ($\delta=0$)	3	1.7σ	4.3
R_3	Anomalous ($\delta=3.0\sigma$)	3	3.3σ	16.8
R_4	Anomalous ($\delta=-0.8\sigma$)	29	-1.0σ	15.3

(b) The anomaly value for each region depends on both the number of data points in it and the observed deviation of these from the nominal model θ^0

Fig. 6. Illustration of the anomaly value function in a 1D context and observation space. Data is generated from a nominal Gaussian model $\boldsymbol{\theta}^0$, except for points in region R_3, which deviate significantly from $\boldsymbol{\theta}^0$, and those in R_4, which deviate subtly from $\boldsymbol{\theta}^0$.

Substituting $\boldsymbol{\delta}_{\max}$ back into Equation (8) gives the final expression for the quantity to maximize:

$$F(R|\boldsymbol{Z}^{\boldsymbol{x}},\boldsymbol{\theta}^0)=\frac{1}{2}\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i\right)^{\mathsf{T}}\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\right)^{-1}\left(\sum_{\boldsymbol{x}_i\in R}\boldsymbol{\Sigma}_i^{-1}\Delta\boldsymbol{z}_i\right) \tag{10}$$

This expression depends only on the nominal model $\boldsymbol{\theta}^0$ (through $\boldsymbol{\Sigma}_i$ and the $\boldsymbol{\mu}_i$ in $\Delta\boldsymbol{z}_i$) and sufficient statistics of the observed data (the two distinct sums of Equation (10)). This is particularly useful when the context space is discretized and only sufficient statistics of each discrete bin, instead of all the data points, need to be stored.

Figure 6 and Table 2 illustrate this anomaly measure for various regions in a one-dimensional context and one-dimensional observation space. The data in Figure 6 is generated directly from the nominal Gaussian model $\boldsymbol{\theta}^0$ defined by $\boldsymbol{\mu}(\boldsymbol{\theta}^0)$ and $\boldsymbol{\sigma}(\boldsymbol{\theta}^0)$, except for the data in regions R_3 and R_4. In R_3, the generated data has been shifted by $3\boldsymbol{\Sigma}$, whereas the data in R_4 has been shifted by $-0.8\boldsymbol{\Sigma}$. Given enough data, the anomaly value should therefore be able to differentiate between nominal regions R_1 and R_2, and anomalous regions R_3 and R_4. Table 2 reveals that this is indeed the case: the nominal regions have similar values $\ln(\text{anom}(R_1))=3.3$ and $\ln(\text{anom}(R_2))=4.3$, whereas the anomalous regions have significantly higher anomaly values $\ln(\text{anom}(R_3))=16.8$ and $\ln(\text{anom}(R_4))=15.3$.

Table 2. Anomaly value function in a 1D context and observation space. The anomaly value for each region depends on both the number of data points in it and the observed deviation of these from the nominal model $\boldsymbol{\theta}^0$.

Region	Ground truth (deviation δ)	Number of data points	Observed deviation	$\ln(\text{anom}(R_i))$
R_1	Nominal ($\delta = 0$)	22	$-0.4\boldsymbol{\sigma}$	3.3
R_2	Nominal ($\delta = 0$)	3	$1.7\boldsymbol{\sigma}$	4.3
R_3	Anomalous ($\delta = 3.0\boldsymbol{\sigma}$)	3	$3.3\boldsymbol{\sigma}$	16.8
R_4	Anomalous ($\delta = -0.8\boldsymbol{\sigma}$)	29	$-1.0\boldsymbol{\sigma}$	15.3

4.1.2. Search space of RIMs : parametric regions. Although we would like to search over the space of all possible subregions of state–action space to find RIMs, we choose to bound the search space to families of parametric shapes. Previous work on spatial anomaly detection has used various parametric shapes (e.g., circles (Kulldorff, 1997) or ellipses (Kulldorff et al., 2006)). Previous work has argued that, although these regions only represent a small subset of all possible regions, their detection power extends significantly beyond the chosen shapes. In addition, parametric shapes have the advantage of having an explicit concise parameter space over which to conduct a search.

Throughout our work, we do not assume a particular choice of parametric shapes, and our algorithms apply to any parametric shape (rectangles, ellipsoids, cylinders, etc.). The choice of shape depends on the domain at hand.

In our implementations, we have chosen to use ellipsoids as the search space for RIMs. Our motivation is that ellipsoids are a generic convex shape that support arbitrary rotations, translations, and scalings, while requiring only $O(d^2)$ parameters. In a d -dimensional context space, an ellipsoid can be parameterized by a d -vector $\boldsymbol{u}$ and a $d \times d$ positive-definite matrix $\boldsymbol{A}$ as the set of points that satisfy

$$(\boldsymbol{x} - \boldsymbol{u})^\intercal \boldsymbol{A}^{-1} (\boldsymbol{x} - \boldsymbol{u}) < 1 \tag{11}$$

Thus, the parameter vector $\boldsymbol{\psi}(\boldsymbol{u}, \boldsymbol{A})$ describing a particular ellipsoid is the linearized form of $\boldsymbol{u}$ and $\boldsymbol{A}$. As $\dim(\boldsymbol{u}) = d$ ($\boldsymbol{u}$ is a d-vector), and $\dim(\boldsymbol{A}) = \frac{d(d+1)}{2}$ ($\boldsymbol{A}$ is a symmetric d-matrix), $\boldsymbol{\psi}$ has dimensionality of $d + \frac{d(d+1)}{2} = \frac{1}{2}(d^2 + 3d)$. The search space is then the space of such vectors $\boldsymbol{\psi} \in \Psi$ such that the derived matrix $\boldsymbol{A}$ is positive definite.

4.2. Detection of a single RIM

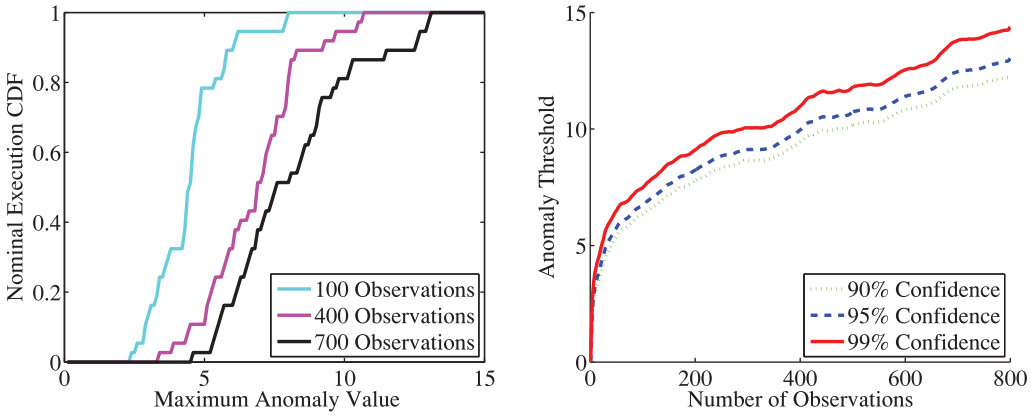
In this section, we use the anomaly measure[4] $\text{anom}(R)$ and the parametric search space defined in Section 4.1 to describe the FARO algorithm for detection of up to one RIM . Although previous work in spatial anomaly detection has used similar measures and parametric shapes, we propose a different algorithm, more suited for online monitoring of execution. As Section 3.3 discusses, previous search methods include exhaustive search over discretizations of the optimization space, and other methods that scale exponentially with the dimensionality of the data. Instead, we propose an optimization-based approach.

Algorithm 2 describes this optimization process. It is an iterative optimization process that maintains a set $\mathcal{R}$ of most promising candidate RIMs throughout time, bounding the size of this set to a constant $|\mathcal{R}|_{\max}$; the value chosen

Algorithm 2. FARO algorithm the for detection of a single RIM .

Input: List of contextual observations $\boldsymbol{Z}^{\boldsymbol{x}}$, nominal model $\boldsymbol{\theta}^0$, and maximum set size $|\mathcal{R}|_{\max}$.
Output: A RIM $R_{\max}$, or $\emptyset$ if no RIM is detected.

function FARO ($\boldsymbol{Z}^{\boldsymbol{x}} = [(\boldsymbol{x}_i, \boldsymbol{z}_i) | i = 0, \ldots, t]$, $\boldsymbol{\theta}^0$, $|\mathcal{R}|_{\max}$)
  $\mathcal{R} \leftarrow \mathcal{R} \cup \text{ball}(\boldsymbol{x}_t)$ ▷ Seed around latest $\boldsymbol{x}$. Initially, $\mathcal{R} = \emptyset$.
  $R_{\max} \leftarrow \emptyset$
  for $R \in \mathcal{R}$ **do**
    $R \leftarrow \text{Optimize}(R, \text{anom}(\cdot, \boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0))$ ▷ Nonlinear optimization
    if $\text{anom}(R) \geqslant a_{\max}(\boldsymbol{\theta}^0, |\boldsymbol{Z}^{\boldsymbol{x}}|)$ **then** ▷ Statistical test for anomaly
      $R_{\max} \leftarrow R$
    end if
  end for
  if $|\mathcal{R}| > |\mathcal{R}|_{\max}$ **then**
    Reduce $\mathcal{R}$ to its $|\mathcal{R}|_{\max}$ elements of highest anom
  end if
  return $R_{\max}$
end function



(a) CDF of the maximum anomaly value observed during nominal execution runs

(b) Anomaly detection threshold for desired detection confidence

Fig. 7. Using repeated simulations of nominal execution, we obtain an empirical distribution of maximum anomaly values. This distribution is used to set the threshold anomaly value for detection, given a desired detection confidence. (a) CDF of the maximum anomaly value observed during nominal execution runs. (b) Anomaly detection threshold for desired detection confidence.

for $|\mathcal{R}|_{\max}$ provides a trade-off between the probability of efficiently converging to the optimal RIM by having more parallel optimizations, and the computational efficiency of having fewer optimizations. At every timestep of execution, the approach begins by seeding a new RIM candidate with a small region around the most recent observation (line 2). Then, the algorithm optimizes each candidate RIM using a nonlinear optimization algorithm (line 5). For this thesis, we have used the cross-entropy method (CEM) of optimization (Rubinstein, 1999), but we could replace it with another optimization algorithm Optimize, that takes as input a region R (or its parameter vector $\boldsymbol{\psi}(R)$) and a value function $\text{anom}(\cdot, \boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0)$ to be optimized.

Once every candidate RIM has been optimized, the algorithm must decide whether any of the candidates are statistically anomalous, and return that candidate as a RIM. This decision is based on comparing $\text{anom}(R)$ with a threshold value $a_{\max}(\boldsymbol{\theta}^0, |\boldsymbol{Z}^{\boldsymbol{x}}|)$, which depends only on the domain during nominal execution, and on the number of observations (line 6). This threshold value can be approximately mapped to a desired rate of false positives in detection through Monte Carlo simulation (Kulldorff, 1997): we run FARO in N simulated executions using nominal model $\boldsymbol{\theta}^0$, and count the number of simulations $n(a_{\max}|\boldsymbol{\theta}^0, |\boldsymbol{Z}^{\boldsymbol{x}}|)$ in which $\text{anom}(R_{\max}) > a_{\max}$ after $|\boldsymbol{Z}^{\boldsymbol{x}}|$ observations:

$$P\big(\text{anom}(R_{\max}) > a_{\max}(\boldsymbol{\theta}^0, |\boldsymbol{Z}^{\boldsymbol{x}}|)|\boldsymbol{\theta}^0\big) \approx \frac{n(a_{\max}|\boldsymbol{\theta}^0, |\boldsymbol{Z}^{\boldsymbol{x}}|)}{N} \tag{12}$$

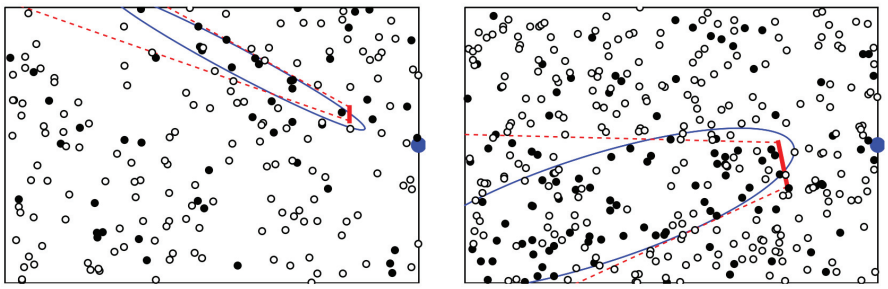
Figure 7 illustrates the establishment of this empirical threshold value. Figure 7a shows the distribution of maximum anomaly values obtained from simulations under a nominal model $\boldsymbol{\theta}^0$. The figure shows that, for longer runs of the system, i.e., more observations, the algorithm finds regions with higher anomaly values even during nominal execution. This is analogous to observing longer streaks of heads (or tails) in longer coin flip sequences, and is the reason why the threshold $a_{\max}$ depends on the number of observations received. Figure 7b shows the empirical cumulative distribution function (CDF) of the maximum anomaly value found for three different observation set sizes. From these empirical CDFs, the algorithm chooses an appropriate threshold value for a desired detection error.

Figure 8 illustrates two examples of FARO detecting RIMs in the simulated golf-robot domain. The larger difference between nominal and anomalous behavior in Figure 8a translates to detection from fewer anomalous samples ($n = 16$) than those required in Figure 8b ($n = 114$). Section 5 describes thorough evaluations of the detection capabilities of FARO.

4.3. Detection of multiple RIMs

Section 4.2 describes our work on the FARO algorithm, created to detect RIMs in domains with up to one RIM . Here, we present the theory and algorithm needed to extend these ideas to domains that may have multiple RIMs, as first explained in previous work (Mendoza et al., 2015). Section 4.3.1 derives the theoretical framework required for detection of multiple RIMs, whereas Section 4.3.2 puts this theory into the DMAPS algorithm for detection of multiple RIMs.

4.3.1. Multiple RIM detection theory. To extend the FARO algorithm to domains with multiple RIMs, we need to address two issues: how to address overlapping RIM candidates, and how to encode our prior knowledge that simpler hypotheses should be favored over more complex ones.



(a) $P(z|x \in \text{RIM}) = 0.2. P(z|x \notin \text{RIM}) = 0.8$ **(b)** $P(z|x \in \text{RIM}) = 0.5. P(z|x \notin \text{RIM}) = 0.8$

Fig. 8. Synthetic data for a RIM (red dashed lines) created by an imperceptible bump (red solid line) in the golf-robot scenario. (a) $P(z|x \in \text{RIM}) = 0.2$. $P(z|x \notin \text{RIM}) = 0.8$. (b) $P(z|x \in \text{RIM}) = 0.5$. $P(z|x \notin \text{RIM}) = 0.8$. The robot has different constant probabilities of success inside and outside of the RIM. White and black circles show successful and failed shots, respectively. Blue ellipses show the RIMs found by FARO.

Overlapping RIMs Trying to directly apply FARO to domains with multiple anomalies, we quickly run into the problem of overlapping regions. If each *RIM* candidate is treated independently, then many of them will converge to the same region. To address this, we state the following assumption.

Assumption 1. *The robot's world may contain several behavioral modes: one nominal mode defined by $\boldsymbol{\theta}^0$, and zero or more unmodeled modes, defined by unknown distributions $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^m$. Each observation in $\boldsymbol{Z}^{\boldsymbol{x}}$ is produced by one of these behavioral modes, and the probability of each mode being active depends on the observation's context.*

Thus, we will try to find the set of regions $\mathcal{R}$ that maximizes a new anomaly measure $\text{Anom}(\mathcal{R}|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0)$ in such a way that each observation in $\boldsymbol{Z}^{\boldsymbol{x}}$ is not attributed to multiple regions in $\mathcal{R}$.

Occam's razor Our second assumption is a specific statement of Occam's razor.

Assumption 2. *Given two possible sets of RIMs $\mathcal{R}_1$ and $\mathcal{R}_2$, both equally well-supported by the observed data, the smaller set is more likely to be the one generating the observed data.*

For example, a set $\mathcal{R}_1$ with a single region that contains 100 anomalous-seeming observations should attain a higher anomaly value than a set $\mathcal{R}_2$ of two regions, each containing 50 of these observations.

This requirement is achieved when the appropriate prior knowledge is added to the anomaly value function: given an alternate distribution $\hat{\boldsymbol{\theta}} \neq \boldsymbol{\theta}^0$, the prior probability of such a distribution is smaller than that of the nominal distribution $P(\hat{\boldsymbol{\theta}}|R) < P(\boldsymbol{\theta}^0|R)$. Then, our new anomaly value function for *sets of regions* is given by

$$\text{Anom}\left(R|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0\right) \equiv \prod_{R \in \mathcal{R}} \frac{\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} P(\boldsymbol{\theta}|\boldsymbol{Z}^{\boldsymbol{x}}, R)}{P(\boldsymbol{\theta}^0|\boldsymbol{Z}^{\boldsymbol{x}}, R)} \tag{13}$$

This function implicitly assumes that for any two regions $R_i \subseteq \mathcal{X}$ and $R_j \subseteq \mathcal{X}$, "$R_i$ is an anomalous region" is conditionally independent from "R_j is an anomalous region," given the list of observations $\boldsymbol{Z}^{\boldsymbol{x}}$. In domains in which $\boldsymbol{Z}^{\boldsymbol{x}}$ is the only source of information about the presence of anomalous regions, this assumption holds. However, there may exist domains in which this assumption does not hold: for example, a domain in which the robots knew in advance that there exist exactly n anomalous regions. Addressing these domains is beyond the scope of this work.

The proposed formulation of Equation (13), leads to a more desirable optimization cost function, which naturally favors simpler hypotheses, as shown by the following theorem.

Theorem 1. *If all the possible anomalous distributions $\hat{\boldsymbol{\theta}} \in \boldsymbol{\Theta}$ are equally likely a priori, then the set $\mathcal{R}_{\max}$ of regions that maximizes $\text{Anom}(\mathcal{R}|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0)$ is also the set that maximizes the simpler function*

$$F\left(\mathcal{R}|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0\right) \equiv \left[\sum_{R \in \mathcal{R}} \log\left(\text{anom}(R|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0)\right)\right] - \sum_{R \in \mathcal{R}} \lambda(R) \tag{14}$$

where $\lambda(R) = \log\left(\frac{P(\boldsymbol{\theta}^0|R)}{P(\hat{\boldsymbol{\theta}}|R)}\right)$ is a function only of R.

Proof. We can expand Equation (13) using Bayes' theorem:

$$\begin{aligned} \text{Anom}(\mathcal{R}|\boldsymbol{Z}^{\boldsymbol{x}}, \boldsymbol{\theta}^0) &= \prod_{R \in \mathcal{R}} \frac{\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} P(\boldsymbol{\theta}|\boldsymbol{Z}^{\boldsymbol{x}}, R)}{P(\boldsymbol{\theta}^0|\boldsymbol{Z}^{\boldsymbol{x}}, R)} \\ &= \frac{\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} P(\boldsymbol{Z}^{\boldsymbol{x}}|\boldsymbol{\theta}, R) P(\boldsymbol{\theta}|R)}{P(\boldsymbol{Z}^{\boldsymbol{x}}|\boldsymbol{\theta}^0, R) P(\boldsymbol{\theta}^0|R)} \\ &= \left[\prod_{R \in \mathcal{R}} \text{anom}(R)\right] \left[\prod_{R \in \mathcal{R}} e^{-\lambda(R)}\right] \end{aligned} \tag{15}$$

Algorithm 3. DMAPS algorithm for detection of multiple RIMs .

Input: List of contextual observations $\boldsymbol{Z}^{\boldsymbol{x}}$, and nominal model $\boldsymbol{\theta}^0$.
Output: A set R of potential RIMs, along with their corresponding anomaly values.

```
1:function DMAPS ( Z^x = [(x_i, z_i)|i = 0, ..., t], θ^0)
2:   R ← R ∪ ball(x_t)                          ▷ Seed around latest x. Initially, R = ∅.
3:   ℝ ← Sort(R)                                ▷ Descending order of anom(R)
4:   Z^¬R ← Z^x                                 ▷ Observations not yet in a region R ∈ R
5:
6:   R ← ∅                                      ▷ Reset R and F(R)
7:   F ← 0
8:   for R ∈ ℝ do
9:     R ← Optimize(R, anom( · , Z^¬R, θ^0))    ▷ Nonlinear optimization.
10:    if log(anom(R, Z^¬R)) ≥ λ(R) then
11:      R ← R ∪ R
12:      F ← F + log(anom(R, Z^¬R)) − λ(R)
13:      Z^¬R ← Z^¬R − Z^x(R)                   ▷ Remove observations in R
14:    end if
15:  end for
16:  return R
17:end function
```

As $\log(\text{Anom}(\mathcal{R}))$ is a monotonically increasing function of $\text{Anom}(\mathcal{R})$, we maximize $\log(\text{Anom}(\mathcal{R}))$ instead of $\text{Anom}(\mathcal{R})$:

$$\begin{aligned}
&\underset{\mathcal{R}\subseteq 2^{\mathcal{X}}}{\arg\max}[\text{Anom}(\mathcal{R})] = \underset{\mathcal{R}\subseteq 2^{\mathcal{X}}}{\arg\max}[\log(\text{Anom}(\mathcal{R}))] \\
&= \underset{\mathcal{R}\subseteq 2^{\mathcal{X}}}{\arg\max}\left[\left[\sum_{R\in\mathcal{R}}\log(\text{anom}(R))\right] - \sum_{R\in\mathcal{R}}\lambda(R)\right] \qquad (16)\\
&= \underset{\mathcal{R}\subseteq 2^{\mathcal{X}}}{\arg\max}[F(\mathcal{R})]
\end{aligned}$$

□

Corollary 1. *In a domain in which the prior probability of region R being anomalous is uniform for all $R \subseteq \mathcal{X}$, $F(\mathcal{R})$ reduces to*

$$F(\mathcal{R}) = \left[\sum_{R\in\mathcal{R}}\log(\text{anom}(R))\right] - \lambda|\mathcal{R}| \qquad (17)$$

where λ is a constant.

For this work, we assume λ to be a constant for all regions, but incorporating a non-uniform informative prior is a subject of interest for future work.

We note that, in domains in which RIMs are less probable than nominal behavior, i.e., $\lambda(R)>0$, the cost function $F(\mathcal{R})$ naturally favors simpler hypotheses with fewer anomalous regions.

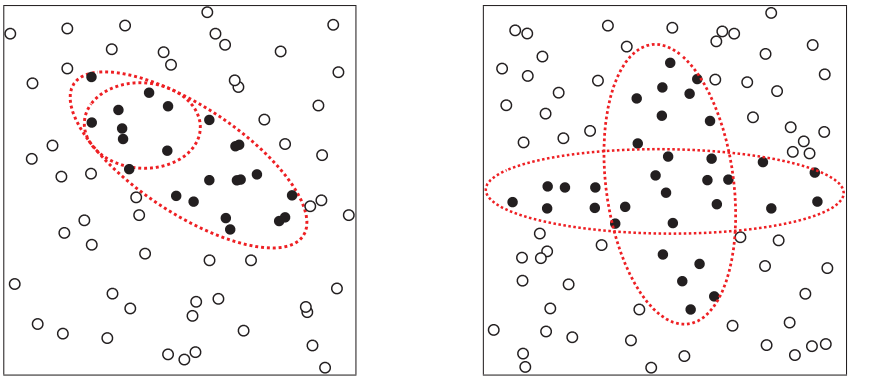
4.3.2. DMAPS algorithm for detection of multiple RIMs. Having defined the appropriate anomaly value function $\text{anom}(\mathcal{R})$, this section describes our procedure for searching for the maximizing region set $\mathcal{R}^*$. As with the FARO algorithm for finding a single RIM, DMAPS uses an iterative optimization approach with various seeds to optimize the candidate regions in the set. However, the candidate regions are not treated as independent from each other: each observation only contributes to the anomaly value of a single candidate region. Furthermore, to encourage the consolidation of simpler hypotheses, i.e., fewer larger RIMs, DMAPS greedily optimizes the current candidates sequentially in non-ascending order of $\text{anom}(R)$.

Algorithm 3 describes the sequential optimization process of DMAPS. First, the algorithm adds a small candidate region around the most recent observation to $\mathcal{R}$ in line 2, similarly to the FARO algorithm. This set of regions is then sorted in non-increasing order of value $\text{anom}(R)$ for sequential optimization. At this point, in line 8, the sequential optimization over candidate regions begins.

Although this optimization bears resemblance to the FARO algorithm of Section 4.2, the results derived in Section 4.3.1 enable the algorithm to keep candidates only if they add to the total value of the set R.

After optimizing each region in line 9, the decision of whether to add this region to the final output set is made in line 10. A region R is only added if it adds value to the optimization cost function F. If a region is added to the candidate set, then the observations in that region are removed from the set $\boldsymbol{Z}^{\neg\mathcal{R}}$ of available observations (line 13) to prevent double-counting of observations. Therefore, for example, if the small region created in line 2 is subsumed by a larger, more anomalous region, the set of observations in $\text{ball}(\boldsymbol{x}_t)$ would contain no available observations; it would thus have a log anomaly value of zero, and would not be added to the set of candidates.

The end result of the DMAPS algorithm is a set of regions $\mathcal{R}_{\max}$ found to be the most likely RIMs in the robot's context space. This set is not a global optimum of cost function $F(\mathcal{R})$ for two reasons. First, the CEM optimization of each region $R \in \mathcal{R}$ finds a locally optimal



(a) Redundant overlapping regions **(b)** Non-redundant overlapping regions

Fig. 9. Examples of (a) a redundant and (b) a non-redundant set of overlapping regions in domains with non-subtle anomalies. DMAPS would discard the smaller region in (a), but would correctly keep both regions in (b).

solution, rather than a globally optimal one. Second, the greedy sequential optimization over multiple anomalies is not a globally optimal assignment either, and counter-examples to its optimality can be found. Nonetheless, detecting RIMs using DMAPS and correcting the model as shown in Section 4.4 can significantly improve performance in complex real-robot domains, such as that in Section 5.3.3.

Notes on overlapping regions. We have noted that DMAPS only counts each data point toward a single RIM candidate. Therefore, for example, the smaller region in the non-subtle-anomaly domain of Figure 9a would automatically be discarded by the algorithm, as it contains no points that are not contained in the larger region with a higher anomaly value. However, this does not mean that DMAPS prevents overlap among RIM candidates. In particular, Figure 9b shows another similar domain, but in which the two overlapping regions are not redundant with each other. Even though the points in the intersection between the regions will only belong to that with the higher anomaly value, each region individually covers a relevant region of space. This is also how DMAPS is capable of approximating RIMs of arbitrary shapes with various parametric regions.

Comparison with FARO. DMAPS presents an extension of the FARO algorithm to domains with an unknown number of RIMs, and thus is most readily applicable to most real robot domains. However, in domains where no more than one RIM can exist, FARO could present a convergence advantage, because overlapping hypotheses can evolve towards different local optima of the same RIM during optimization.

4.4. RIM correction

Section 4.3 describes the DMAPS algorithm to find a set $\mathcal{R}$ of RIMs. This section describes how robots can use this information to correct their planning models. This corresponds to line 6 in Algorithm 1. In essence, we treat each RIM as a context-dependent behavioral mode, distinct from the nominal behavioral mode, in which outcome observations are distributed differently. Previous work has explored the problem of learning different behavioral modes as nodes of an hidden Markov model (Fox et al., 2006) or a dynamic Bayesian network (Infantes et al., 2006), depending on the observability of controllable variables. In contrast with these methods, in which the robots learn the entire structure of the underlying graph, our robots begin execution with a model of nominal behavior, and then learn new behavioral modes in specific contexts where their models are not accurate.

We seek to compute a corrected model $\boldsymbol{\theta}^+$ based on the nominal model $\boldsymbol{\theta}^0$ and the set $\mathcal{R}$ of detected RIMs :

$$P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^+) \equiv P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^0,\mathcal{R}) \tag{18}$$

We assume that when the robot performs action a in state s, the world behaves according to its nominal model $\boldsymbol{\theta}^0$ or one of the $|\mathcal{R}|$ behaviors $\boldsymbol{\theta}^i$ detected by DMAPS. We denote the set of plausible models as $\boldsymbol{\Theta}^{\mathcal{R}} = \{\boldsymbol{\theta}^0\} \cup \{\boldsymbol{\theta}^i | i \in 1, \ldots, |\mathcal{R}|\}$. By the law of total probability, we obtain

$$P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^+) = \sum_{\boldsymbol{\theta}^i \in \boldsymbol{\Theta}^{\mathcal{R}}} \left[P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^i,\mathcal{R}) P(\boldsymbol{\theta}^i|\boldsymbol{s},\boldsymbol{a},\mathcal{R}) \right] \tag{19}$$

The distribution of observations is thus a mixture model of the plausible models in $\boldsymbol{\Theta}^{\mathcal{R}}$, appropriately weighted depending on how likely the context $\boldsymbol{x} = \chi(\boldsymbol{s},\boldsymbol{a})$ is to be in each region in $\mathcal{R}$. For clarity, we rename the second factor in the sum of Equation (19), which corresponds to these weights, as $\alpha_i(\boldsymbol{s},\boldsymbol{a},\mathcal{R}) \equiv P(\boldsymbol{\theta}^i|\boldsymbol{s},\boldsymbol{a},\mathcal{R})$. As $\boldsymbol{z}$ is independent of $\mathcal{R}$ given the appropriate model $\boldsymbol{\theta}^i$, we obtain that

$$P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^+) = \sum_{i=0}^{|\boldsymbol{\Theta}^{\mathcal{R}}|} \left[\alpha_i(\boldsymbol{s},\boldsymbol{a},\mathcal{R}) P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^i) \right] \tag{20}$$

Section 4.4.1 addresses the problem of estimating the model $\boldsymbol{\theta}^i$ for each RIM, whereas Section 4.4.2 discusses the estimation of α_i. Then, Section 4.4.3 describes how to bring these together into the final model correction equations.

4.4.1 Estimating observation distributions. Here $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^i)$ describes the distribution of observations $\boldsymbol{z}$ given that the world is in a RIM defined by model $\boldsymbol{\theta}^i$. For nominal execution, this is simply the model-given predicted distribution $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^0)$.

In the absence of prior knowledge about the distribution of anomalies, we select the alternate model $\boldsymbol{\theta}^i$ as that from a set of plausible alternate models $\hat{\boldsymbol{\theta}}$ that maximizes the likelihood of the observed data in region R_i:

$$\boldsymbol{\theta}^i \equiv \underset{\hat{\boldsymbol{\theta}}\in\hat{\boldsymbol{\Theta}}}{\operatorname{argmax}} \left[\prod_{(\boldsymbol{x}_j,\boldsymbol{z}_j)\in\boldsymbol{Z}^x} P(\boldsymbol{z}_j|\hat{\boldsymbol{\theta}}(\boldsymbol{x}_j))^{\mathbb{1}(\boldsymbol{x}_j\in R_i)} \right] \tag{21}$$

In particular, for the examples of this article, we assume that the distribution in each RIM $R_i \in \mathcal{R}$ has the same shape as $\boldsymbol{\theta}^0$ but a shifted mean $\boldsymbol{\mu}+\boldsymbol{\delta}$. In the case of Gaussian distributions, for example, the maximum likelihood distribution $\boldsymbol{\theta}^i$ is given by

$$P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^i) = \mathcal{N}(\boldsymbol{\mu}_0(\boldsymbol{s},\boldsymbol{a})+\boldsymbol{\delta}(R_i),\boldsymbol{\Sigma}_0(\boldsymbol{s},\boldsymbol{a})) \tag{22}$$

where $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ are the parameters predicted by the nominal model $\boldsymbol{\theta}^0$, and $\boldsymbol{\delta}(R_i)$ is the maximum likelihood shift in mean of Equation (9). Although this article focuses on inaccuracies in the form of a shifted mean, the maximum likelihood variance changes could also be estimated.

4.4.2. Estimating active behavior distributions. Coefficients $\alpha_i \equiv P(\boldsymbol{\theta}^i|\boldsymbol{s},\boldsymbol{a},\mathcal{R})$ describe the probability of the world being in each behavioral mode, including nominal behavior, given that the robot takes action $\boldsymbol{a}$ in state $\boldsymbol{s}$. The nominal behavior probability is constrained by the probabilities of the anomalous modes:

$$P(\boldsymbol{\theta}^0|\boldsymbol{s},\boldsymbol{a},\mathcal{R}) = 1 - \sum_{i=1}^{|\mathcal{R}|} P(\boldsymbol{\theta}^i|\boldsymbol{s},\boldsymbol{a},\mathcal{R}) \tag{23}$$

so we focus on computing the probability of the anomalous behavioral modes.

The activation probability of each RIM $\boldsymbol{\theta}^i$ depends both on whether the context $\boldsymbol{x}=\chi(\boldsymbol{s},\boldsymbol{a}) \in \mathcal{X}$ lies inside of region R_i, and on our confidence that R_i is actually a RIM that behaves according to $\boldsymbol{\theta}^i$. Thus, the activation probability is given by

$$\alpha_i(\boldsymbol{s},\boldsymbol{a},\mathcal{R}) \equiv P(\boldsymbol{\theta}^i|\boldsymbol{s},\boldsymbol{a},\mathcal{R}) = P(\boldsymbol{\theta}^i|\chi(\boldsymbol{s},\boldsymbol{a})\in R_i)P(\chi(\boldsymbol{s},\boldsymbol{a})\in R_i) \tag{24}$$

For this work, $P(\chi(\boldsymbol{s},\boldsymbol{a})\in R_i)$ simply indicates whether a state–action point belongs to region R_i:

$$P(\chi(\boldsymbol{s},\boldsymbol{a})\in R_i) = \mathbb{1}(\chi(\boldsymbol{s},\boldsymbol{a})\in R_i) \tag{25}$$

In domains with noisy measurements of $\boldsymbol{s}$, or with anomalous regions with fuzzy boundaries, a softer definition of "belonging" to region R_i could be more appropriate.

To calculate the confidence value $P(\boldsymbol{\theta}^i|\chi(\boldsymbol{s},\boldsymbol{a})\in R_i)$, i.e., the probability that R_i is actually a RIM with distribution $\boldsymbol{\theta}_i$, we again employ Monte Carlo simulation, similarly to how we used it in Section 4.2: we note that this confidence is a monotonically increasing function of $\text{anom}(R)$, so there exists a function $\mathbb{P}:\mathbb{R}\to\mathbb{R}$ such that

$$\mathbb{P}(\text{anom}(R_i)) = P(\boldsymbol{\theta}^i|\boldsymbol{s},\boldsymbol{a},\mathcal{R}) = P(\boldsymbol{\theta}^i|\chi(\boldsymbol{s},\boldsymbol{a})\in R_i) \tag{26}$$

We estimate this function empirically using the the empirical CDF of Figure 7 and denote it by $\hat{\mathbb{P}}$. That is, the estimated probability $\hat{\mathbb{P}}$ of R_i being a RIM is determined by the number of runs, during nominal execution simulations, that the maximum anomaly value of a region exceeded $\text{anom}(R_i)$.

Combining these results, Equation (24) becomes

$$\alpha_i(\boldsymbol{s},\boldsymbol{a},\mathcal{R}) = \mathbb{1}(\chi(\boldsymbol{s},\boldsymbol{a})\in R_i)\hat{\mathbb{P}}(\text{anom}(R_i)) \tag{27}$$

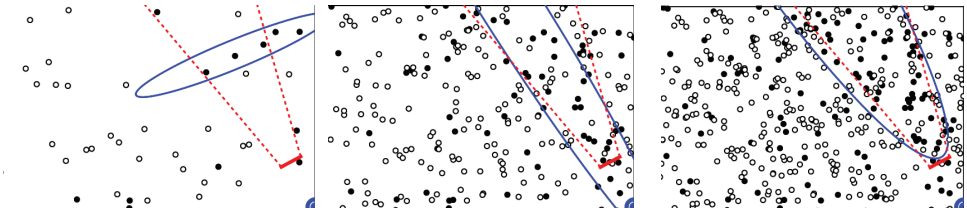
4.4.3 Applying model corrections. Having estimated the distribution $P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^i)$ for each region R_i, and its activation weight α_i, Equation (20) becomes fully defined. Using normal distributions as described in Section 4.4.1 and α_i as defined in Section 4.4.2, we obtain the corrected model:

$$\begin{aligned} P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^+) = {} & P(\boldsymbol{z}|\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0) \\ & \left(1 - \sum_{R\in\mathcal{R}} \left[\mathbb{1}(\chi(\boldsymbol{s},\boldsymbol{a})\in R)\ \hat{\mathbb{P}}\ (\text{anom}(R))\right]\right) + \\ & \sum_{R\in\mathcal{R}} \left[P(\boldsymbol{z}|\boldsymbol{\mu}_0+\boldsymbol{\delta}(R),\boldsymbol{\Sigma}_0)\mathbb{1}(\chi(\boldsymbol{s},\boldsymbol{a})\in R)\hat{\mathbb{P}}(\text{anom}(R))\right] \end{aligned} \tag{28}$$

Given that $\chi(\boldsymbol{s},\boldsymbol{a})$ can only be part of one anomalous region by Assumption 1, we obtain the final expression for the distribution of observations:

$$P(\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^+) = \begin{cases} P(\boldsymbol{z}|\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)(1-\hat{\mathbb{P}}(\text{anom}(R))) + & \text{if } \exists R\in\mathcal{R} \text{ s.t.} \\ P(\boldsymbol{z}|\boldsymbol{\mu}_0+\boldsymbol{\delta}(R),\boldsymbol{\Sigma}_0)\hat{\mathbb{P}}(\text{anom}(R)) & \chi(\boldsymbol{s},\boldsymbol{a})\in R \\ P(\boldsymbol{z}|\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0) & \text{otherwise} \end{cases} \tag{29}$$

From this expression, the planner can estimate different statistics of the predicted distribution of $\boldsymbol{z}$, such as its expected value:



(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. 10. Most anomalous ellipse found during three stages of execution: (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. During nominal execution, the robot succeeds with probability $p = 0.8$. When shooting from behind the bump, it succeeds with probability $q = 0.5$.

$$\mathbb{E}[\boldsymbol{z}|\boldsymbol{s},\boldsymbol{a},\boldsymbol{\theta}^{+}] = \begin{cases} \boldsymbol{\mu}_0 + \boldsymbol{\delta}(R)\hat{\mathbb{P}}(\text{anom}(R)) & \text{if } \exists R \in \mathcal{R} \text{ s.t. } \chi(\boldsymbol{s},\boldsymbol{a}) \in R \\ \boldsymbol{\mu}_0 & \text{otherwise} \end{cases} \tag{30}$$

The re-estimated model $\boldsymbol{\theta}^{+}$ enables robots to adapt online to detected RIMs in their world. Section 5 empirically demonstrates the benefits of correcting these RIMs online.

5. Evaluation

Section 4 describes our theory and algorithm contributions to online detection and correction of RIMs. This section describes how we evaluate each of these contributions, with extended analysis of results presented in previous preliminary work (Mendoza et al., 2014, 2015).

First, we describe an evaluation of the FARO algorithm for detection of domains with no more than one RIM . We test the detection capability of FARO on synthetically generated data from the golf-putting robot domain (Section 5.1) and real-world data from the CoBot service robots (Veloso et al., 2012) (Section 5.2). Although the former provides an easily visualizable testing 2D domain, the latter provides a more difficult realistic testing domain.

We then proceed to evaluate the DMAPS algorithm for detection and correction of multiple RIMs in the robot soccer domain (Section 5.3.3). We briefly describe the interception–keepaway soccer domain, and proceed to describe the results of applying DMAPS to it.

Our evaluation shows that (a) FARO and DMAPS are capable of detecting subtle context-dependent model inaccuracies in a wide variety of domains, and that (b) detecting and correcting these RIMs online can substantially improve task performance.

5.1. FARO golf-domain evaluation

We first show experiments and results using synthetic data from the golf-robot domain described in Section 1. Figure 10 shows this domain, in which the robot's task is to repeatedly shoot a ball into a hole on the far right-hand side of the field. The domain has one RIM created by an imperceptible bump on the field, which causes the robot to miss its target more often than expected when shooting from behind the bump.

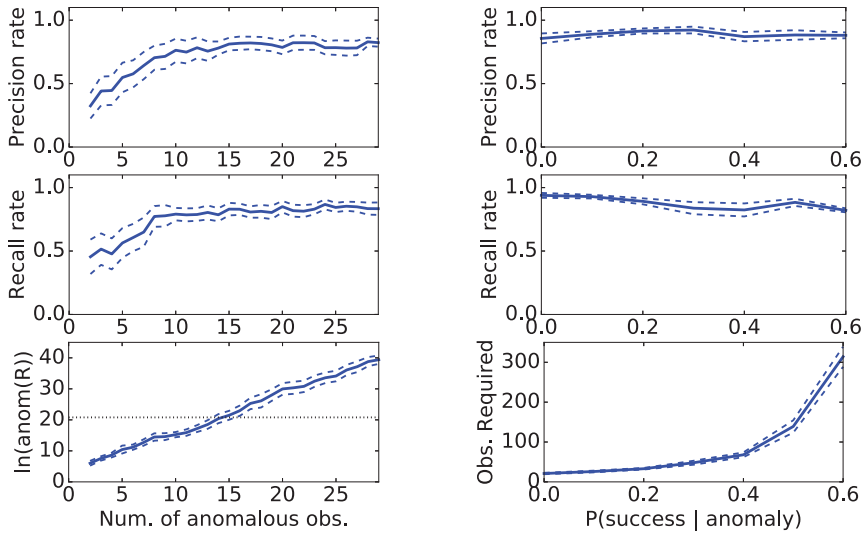
Given a bump location, the simulated data from this domain is generated iteratively as follows: (a) sample a context 2D point, from the uniform distribution over the field; (b) if the bump is not between the sampled point and the hole on the right, sample the shot success from the nominal distribution $\boldsymbol{\theta}^0$; (c) if the bump is between the sampled point and the hole, sample the shot success from the RIM distribution $\boldsymbol{\theta}^R$. For these experiments, both the nominal and anomalous distributions were constant bernoulli distributions across the field, with success probabilities p and q, respectively; however, the FARO algorithm does not require these probabilities to be constant across the domain.

Experimental setup. The field measures $6\text{m} \times 4\text{m}$. The parameters in this world are the shape and location of the unperceived bump obs, the probability p of success during normal execution, and the probability q of success from locations that are blocked by obs. For each test, obs is uniformly sampled from a set of linear bumps such that: (a) the distance between the target and the center of obs is between 0.1 and 1.5 m; (b) the center of obs is in the field; and (c) the angle subtended by the endpoints of obs and the target is between $\frac{\pi}{16}$ and $\frac{\pi}{4}$ rad. This randomization creates test RIMs of various shapes, sizes, and orientations.

The observations are given to the robot sequentially, and FARO is run after every new observation, as specified in Algorithm 2. Figure 10 shows an example of the evolution of the most likely RIM as the robot gathers more data.

To evaluate how well FARO approximates RIMs on the field, we define *precision* and *recall* values. Precision computes the fraction of points in the detected region R that are in the ground truth RIM S; recall computes the fraction of points in the ground truth RIM S that were detected by R:

$$\text{precision} = \frac{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in R \cap S\}|}{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in R\}|} \tag{31}$$



(a) Precision, Recall and Anomaly value with varying number of anomalous data points

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

Fig. 11. Synthetic data experiments results: (a) precision, recall and anomaly value with varying number of anomalous data points; (b) precision, recall and number of anomalous points when threshold anomaly was passed. Blue dashed lines indicate standard error bars. The black dashed line is the highest anomaly value observed during nominal execution.

$$\text{recall} = \frac{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in R \cap S\}|}{|\{\boldsymbol{x}_i : \boldsymbol{x}_i \in S\}|} \tag{32}$$

Experiment 1: size of the dataset. The goal of the first experiment is to determine the effect of the number of available anomalous data points on the effectiveness of the detection. To do this, we hold both p and q constant and keep track of the precision and recall rates of the algorithm as the number of anomalous data points increases.

Figure 11a show the results of these experiments for $p = 0.8$, $q = 0.2$. Precision and recall both increase rapidly to about 0.8 with around 10 observations, and then keep slowly increasing with more data. The anomaly value, which correlates with the confidence in an anomaly, grows exponentially as more anomalous data arrives ($\ln \text{anom}(R)$ grows linearly). Both $\text{anom}(R)$ and the accuracy of the ellipse approximation increase with the number of anomalous data points.

Precision and recall do not reach 1.0 even when the ellipse approximates the true anomaly very well, as in Figure 10c. There are two reasons for this apparently suboptimal performance. (1) The true shape of each region cannot be arbitrarily well-approximated by any single ellipse; the efficiency of choosing a relatively small parameterization (i.e., ellipses) comes with the cost of the inability to represent regions arbitrarily well: we note, however, that DMAPS partially addresses this problem, because multiple ellipses can significantly improve the approximation. (2) If there are missed shots outside of the truly anomalous region, an ellipse that includes those shots has a higher value than an ellipse that does not, and conversely with successful shots that are inside of the anomalous region; thus, *for finite numbers of observations*, the ellipse that maximizes anomaly is not necessarily the one that best matches the true anomaly region.

Experiment 2: magnitude of inaccuracy. To determine the effect of the magnitude of the inaccuracy on the algorithm's performance, we analyze the number of data points required for the detector to reach a particular threshold of high anomaly confidence, as the anomalous distribution q approached the normal distribution $p = 0.8$. Figure 11b 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 highly super-linear 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 is approximately independent of the magnitude of the failure, given an anomaly threshold $a_{\max}$.

5.2. FARO CoBot motion evaluation

We also evaluate FARO by introducing inaccuracies in the CoBot's motion model domain described in Section 2.1.1. We define the motion state of the CoBot by its translational and rotational position and velocities: $\boldsymbol{x} \equiv [x \quad y \quad \phi \quad \dot{x} \quad \dot{y} \quad \dot{\phi}]^\intercal$; the observations of execution are $\boldsymbol{z} \equiv [\Delta\dot{x} \quad \Delta\dot{y} \quad \Delta\dot{v}]$, the vector difference between the CoBot's measured velocity, obtained from its wheel encoders, and its expected velocity, based on its velocity

Table 3. CoBot experiment results. For each experiment, we present 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.

RIM	Anomaly threshold = 40.2			End of experiment		
	$\lvert\boldsymbol{Z}\rvert$	Precision	Recall	anom	Precision	Recall
Encoder	58±8.2	1.0±0.0	0.76±0.15	337±14	1.0±0.0	0.92±0.03
Collision	4.5±2.1	1.0±0.0	0.81±0.09	1,535±54	1.0±0.0	0.85±0.03
Corridor	60±11	0.94±0.0	0.77±0.15	199±28	0.97±0.01	0.90±0.02
Left turn	31±5.1	1.0±0.0	0.79±0.07	203±22	0.80±0.18	0.47±0.12

command and its state. These observations, which are execution residuals, are expected to be near zero during nominal execution. We obtain the variance of these observations during nominal execution.

To evaluate the FARO detector, we inject four types of RIMs in the CoBot's motion, as the real robot completes navigation tasks around the building.

Encoder failure. Every time the CoBot moves, one of its wheel encoders observes $(1-\epsilon)d$, at each timestep, where d is the displacement of the wheel returned during nominal execution. This failure mode evaluates FARO for RIMs that occupy the whole domain physical domain, and any points in velocity space that are non-zero.

Collision. During an otherwise normal run of the robot, a sudden collision happens. This failure mode evaluates FARO for very small, localized RIMs of context space.

Corridor failure. The wheel encoders fail by returning $(1-\epsilon)d$, as above, but only in a particular corridor of the building. Thus, the context in which this RIM is active is the particular corridor in physical space, and any non-zero velocities.

Left turn failure. The robot's execution is nominal 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 velocity command v. As the robot turns only at intersections or when it needs to face a doorway, this failure mode tests the algorithm when small clusters of anomalous points happen infrequently and far apart.

With the exception of the collision anomaly, these failures were injected into the software of the CoBot's wheel encoder firmware. For each of these scenarios, we commanded the robot to autonomously navigate to several offices in its building, while keeping FARO running. The route that the robot chooses depends on its autonomous navigation algorithms (Biswas and Veloso, 2013). Table 3 summarizes the results of running the algorithm under the different failure modes, with $\epsilon = 0.05$.

The algorithm shows high precision and recall for each of the experiments. The variance in the precision and recall rates is small but not insignificant, indicating that some runs were probably more difficult than others. For example, recall may be higher for a run in which the robot goes down the bad corridor in the same direction multiple times than for a run in which it went down and up the bad corridor: in the latter case, the ellipse had to expand over a gap between regions with $\phi \approx \alpha$ and those with $\phi \approx \pi + \alpha$, where α is the angle of the corridor itself.

The experiment that stands out in terms of results is the *Bad turn left*, for which the recall rate is significantly lower than the others. This is because the anomaly happens in very disjoint regions of context 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 anomalous regions were visited after the time of detection, and some of them were not joined to the main ellipse. This problem is due to the local optimum limitation of our approach, discussed in Section 4. Related research in the field has addressed the problem of RIM detection in high-dimensional spaces (Mendoza et al., 2016c).

5.3. Detecting and correcting multiple RIMs in robot soccer

We evaluate the DMAPS detection algorithm of Section 4.3 and the benefits of model correction as presented in Section 4.4 on a complex sub-problem of the robot soccer domain introduced in Section 2.1.2. The domain, which we call interception–keepaway, is inspired by the keepaway domain introduced for 2D robot soccer simulation (Stone et al., 2005b). We proceed to describe the interception–keepaway problem in Section 5.3.1, then we describe the application of DMAPS in this domain in Section 5.3.2, and finally we evaluate its efficacy in Section 5.3.3.

5.3.1. Interception–keepaway domain. In the keepaway benchmark domain (Stone et al., 2005a), a team of *n keepers* tries to keep the soccer ball within a bounded 2D region, and away from a team of *m takers*, who try to gain possession of the ball. The task is divided into *episodes*, each beginning with the robots and the ball in particular semi-random positions on the field (Stone et al., 2005a). An episode ends when the ball leaves the bounded region or it is held by one of the takers for a significant period of time, at which point a new episode begins. Robots are rewarded for each time step an episode persists.

Keepaway has been used extensively in the simulation league of robot soccer, but we extend it to a real robot domain with the SSL robots. Unlike simulation, real robots

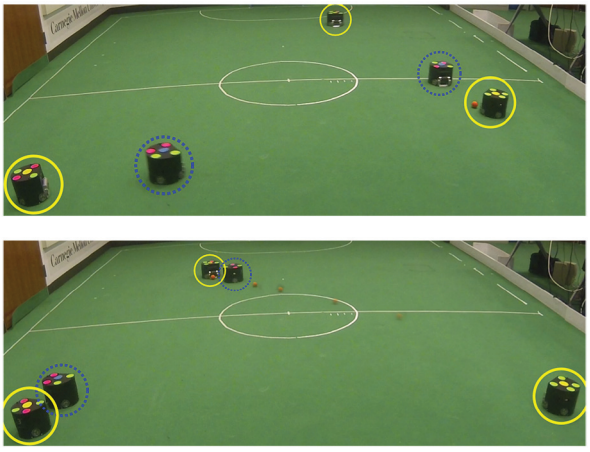


Fig. 12. Interception–keepaway domain. The keepers (yellow solid circles) attempt to maintain possession by passing to each other, while the takers (blue dashed circles) try to steal possession by intercepting passes.

cannot run millions of trials to approach an optimal policy. Running real robot trials takes human effort, causes wear on the robots (changing their dynamics in the process), and cannot be sped up. Robots thus need to adapt online and from sparse observations, especially to perform well against unknown opponents in realistic timescales.

Interception–keepaway robot behavior. Figure 12 shows this interception–keepaway domain with three *keepers* and two *takers*. The keepers try to maintain possession of the ball by performing passes that they believe will not be intercepted by the takers. The takers focus on intercepting passes from the keepers. While the ball is not in motion, one taker positions itself between the keeper k_0 in possession of the ball and its most open teammate, at a small distance from k_0; the remaining takers position themselves between k_0 and its remaining teammates k_i, at a small distance from k_i. When a pass is performed, the taker with the shortest interception time tries to intercept the pass.

Performance evaluation. Performance is measured by the completion rate of passes as $\frac{n(\text{success})}{n(\text{success})+n(\text{failure})}$. We define a successful pass as one in which the ball touches a keeper before it touches a taker or goes out of bounds, whereas a failed pass is one in which the ball touches a taker before it touches a keeper or the ball goes out of bounds.

5.3.2 Model correction for keepaway. We focus on the passer's decision-making problem. Each time a keeper receives the ball, it must choose where to pass next to maintain possession. The physical state of the world $\boldsymbol{s}$ is a vector of size $\dim(\boldsymbol{s}) = 6n + 6m + 4$ containing the 2D translational coordinates $\boldsymbol{l}_i$ and 1D rotational coordinates ϕ_i of each robot, their first time derivatives $\boldsymbol{v}_i$, and the ball's 2D location $\boldsymbol{l}_b$ and velocity $\boldsymbol{v}_b$.[5] The actions available to the robot are the legal velocities $\boldsymbol{v}_b$ ($|\boldsymbol{v}_b| \leqslant 8\,\frac{m}{s}$) at which it can kick the ball, discretized by magnitude and direction.

We monitor the expected probability of success of passes, $P(z = 1|\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^0)$, where an observation of $z = 1$ means the pass succeeded and $z = 0$ means the pass failed. The nominal model $\boldsymbol{\theta}^0$ is based on the planner's estimate of the time $t_i(\boldsymbol{l}_i, \boldsymbol{v}_i, \boldsymbol{l}_b, \boldsymbol{v}_b)$ that each robot i would need, starting at location $\boldsymbol{l}_i$ with velocity $\boldsymbol{v}_i$, to intercept a moving ball starting at location $\boldsymbol{l}_b$ with velocity $\boldsymbol{v}_b$. Given such an estimate (e.g., Biswas et al., 2014)), the model computes the keepers' time to ball t_k:

$$t_k(\boldsymbol{l}_b, \boldsymbol{v}_b) = \min_{i \in \text{keepers}} t_i(\boldsymbol{l}_i, \boldsymbol{v}_i, \boldsymbol{l}_b, \boldsymbol{v}_b) \tag{33}$$

and similarly t_t for takers. We thus define the nominal model $\boldsymbol{\theta}^0$ as

$$P(z|\boldsymbol{s}, \boldsymbol{a} = \boldsymbol{v}_b, \boldsymbol{\theta}^0) = \Phi\left(\frac{t_t(\boldsymbol{l}_b, \boldsymbol{v}_b) - t_k(\boldsymbol{l}_b, \boldsymbol{v}_b)}{\boldsymbol{\sigma}_t}\right) \tag{34}$$

where Φ denotes the standard normal cumulative distribution, and $\boldsymbol{\sigma}_t$ is an uncertainty factor. That is, we model the probability of success as being entirely dependent on which robot has the shortest interception time, plus normal noise.

To conclude the definition of the monitor from Algorithm 1, we must define a feature extraction function $\chi(\boldsymbol{s}, \boldsymbol{a})$ (see line 2). For each expectation e_i about taker robot i, we extract a five-dimensional feature vector:

$$\chi(\boldsymbol{s}, \boldsymbol{a}) = (\boldsymbol{l}'_i, \boldsymbol{v}'_i, |\boldsymbol{v}_b|) \tag{35}$$

where $\boldsymbol{l}'_i$ and $\boldsymbol{v}'_i$ are the location and velocity of robot i relative to the ball at the time the pass starts, rotated by the direction of the pass. By excluding features of the keepers and the other takers, to keep the context space of a manageable dimensionality, we implicitly assume that the only source of inaccurate modeling is the intercepting taker robot's behavior.

Algorithm 1 is thus fully defined for keepaway, and runs each time a pass ends, providing corrections to the planner.

The planning model used here is a simple one, as the focus of this paper is performance improvement through execution monitoring, rather than optimal planning. We use a greedy planner that chooses the action that maximizes the expected immediate reward. In keepaway, that means the passing robot maximizes the expected probability of success of its next pass, and does not plan for multiple passes in the future. In practice, then, the planner always chooses the action that maximizes Equation (30), where $\boldsymbol{\mu}_0$ is given by the expectation of Equation (34). Section 5.3.3 presents an empirical evaluation of this monitor.

5.3.3. DMAPS evaluation on interception–keepaway. To empirically demonstrate our model correction framework, we implemented it on the robot soccer domain of Section 2.1.2. The algorithm was extensively evaluated using a realistic PhysX-based simulator, which employs the same interface to the AI as the real world does, models the robots at the component level, and simulates physics to high detail

(e.g., it models the angular momentum imparted on the ball when a robot touches it with its spinning dribbling mechanism). As we seek to improve high-level robot decisions, rather than low-level controllers, simulation is a particularly useful means of obtaining statistically significant results, which can then be corroborated on the real robots.

Evaluation metrics. The first metric by which we evaluated the model-correction framework is task performance (TP). The ultimate goal of our monitor is to improve TP in environments with RIMs, which makes it a natural metric to evaluate our framework. TP was measured by the average pass completion rate of a particular model $\boldsymbol{\theta}$ as

$$\mathrm{TP}(\boldsymbol{\theta}) = \frac{n(\mathrm{success}|\boldsymbol{\theta})}{n(\mathrm{success}|\boldsymbol{\theta}) + n(\mathrm{failure}|\boldsymbol{\theta})} \tag{36}$$

Although TP is an intuitive evaluation metric, it is highly dependent on the task at hand. For example, improving TP by 1% in a task that originally had 50% success rate is much less meaningful than improving TP in a task that originally had 98% success rate. An evaluation metric that more objectively measures model correction performance is the failure reduction rate (FRR) of the corrected model $\boldsymbol{\theta}^+$ with respect to the baseline model $\boldsymbol{\theta}^0$:

$$\mathrm{FRR}(\boldsymbol{\theta}^+, \boldsymbol{\theta}^0) = 1 - \frac{1 - \mathrm{TP}(\boldsymbol{\theta}^+)}{1 - \mathrm{TP}(\boldsymbol{\theta}^0)} \tag{37}$$

FRR measures the expected percentage of failures that were eliminated by correcting the model. A perfect learner, in a task for which perfect performance is achievable, would eventually reach $\mathrm{FRR} = 1$.

A different metric used for evaluation is model prediction accuracy (MPA). Although improvement in TP is the main goal, we are also interested in evaluating how well the predictions made by the corrected model match the execution. Modeling the world accurately is important to enable robots to generalize to different tasks in the same domain. For example, models acquired during keepaway learning could generalize to passing in the wider problem of full soccer. MPA was measured by the average likelihood of observations given the model used for prediction of that observation:

$$\mathrm{MPA}(\boldsymbol{\theta}) = \frac{1}{|\boldsymbol{Z}|} \sum_{(\boldsymbol{x}, z) \in \boldsymbol{Z}} P(z|\boldsymbol{x}, \boldsymbol{\theta}) \tag{38}$$

For all of our performance metrics, and throughout our experiments, we use as a baseline the original model $\boldsymbol{\theta}^0$.

When analyzing results, we keep in mind the timescale of the learning process in which we are most interested. A game of robot soccer lasts 20 minutes (1,200 seconds). During our keepaway tests, we measured that robots completed, on average, around 0.45 passes per second. Therefore, the upper limit number of passes they could perform in a game is around 540. A considerable portion of that time will be spent not passing (e.g., opponent possession, dead time between plays, shots on goal). However, this estimate lets us narrow our timescale of interest to the order of 100 passes. The experiments conducted here thus focused on such timescales.

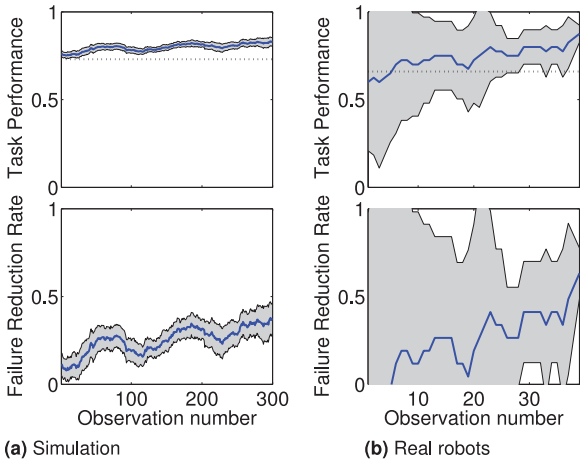


Fig. 13. Moving average of passing performance evaluation as a function of number of passes performed: (a) simulation; (b) real robots. The shaded area shows the 95% confidence interval; the dotted black horizontal line indicates average baseline performance.

Experimental results. First, we conducted extensive simulation tests to determine in a statistically significant way the evolution of $\mathrm{TP}(\boldsymbol{\theta}^+)$ and $\mathrm{MPA}(\boldsymbol{\theta}^+)$ as the monitor acquires new observations. Figure 13a shows a moving average (window size 50) measurement of TP and FRR as a function of how many passes the robots have performed, demonstrating an evident performance improvement as the model is corrected with experience. Performance quickly improves with the first few observations: the first data point shows $\mathrm{FRR} \approx 0.1$, i.e., there is a 10% reduction in failures within the first 50 observations. Furthermore, as new pass results are observed, performance keeps improving, achieving a FRR of about 40% within the first few hundred passes. Conducting less-extensive experiments on the real robots showed a similar trend, as shown in Figure 13b. Baseline and adaptation performance were both higher in simulation than in the real robots, owing to the complications added by the real world.

To evaluate MPA adequately, we ran experiments in which the robots ignored model corrections suggested by the execution monitor. This allowed us to evaluate the predictive performance of the monitor without the confounding factor of altered robot behavior. We evaluate MPA exclusively for points inside of detected RIMs, as this is where model improvement is expected to occur due to our monitor. For points that lie outside of the detected RIMs, the model (and, thus, MPA) remains unchanged by the monitor.

Figure 14a shows $\mathrm{MPA}(\boldsymbol{\theta}^+)$ evaluation for simulation. With model correction, observations show a significantly better fit to the model than without correction. The ideal

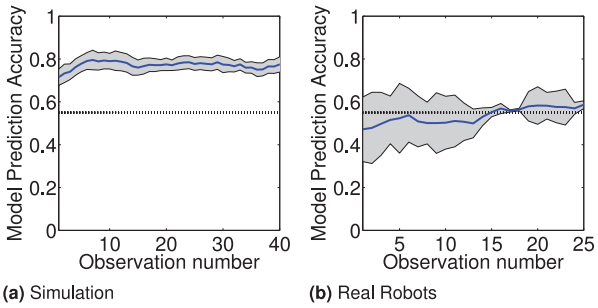


Fig. 14. Moving average of prediction accuracy MPA($\boldsymbol{\theta}^+$), as a function of observations inside of RIMs : (a) simulation; (b) real robots. The shaded area shows the 95% confidence interval; the dotted black horizontal line indicates average baseline MPA($\boldsymbol{\theta}^0$).

MPA = 1 is only achievable by distributions with zero variance, which is certainly not the case in our keepaway domain. The realistic goal of the monitor is not to reach MPA = 1, but to show significant improvement over the initial global model. Figure 14b shows less-extensive real-robot test results, which show a similar trend as simulation.

These results support the efficacy of our framework for short-term task performance and prediction accuracy improvement. Anecdotal evidence from this domain also suggests a different benefit of our approach: system designers may interpret the context and effect of these RIMs to understand the limitations and flaws in the designed models. After conducting the experiments above, the model designers found that the discovered RIMs corresponded to flaws in the design of their simple prediction model of Equation (34). Some RIMs corresponded to inadequacies in the computation of navigation times t_k and t_t, whereas others corresponded to inadequacies in the simplified assumption that the probability of success corresponds exclusively to a smoothly varying function of $t_t - t_k$. An example of this oversimplification occurs when an taker is directly between the ball and a keeper, but very close to the keeper: the simple model indicates that a pass to that position would very slightly favor the taker, as their interception times are almost equal, i.e., $t_t - t_k$ is very small and negative; in reality, because the taker is directly blocking the keeper, its probability of interception is much larger than the keeper's probability of success. This type of knowledge, acquired solely due to autonomous RIM detection, has been incorporated into the source code of the CMDragons, making them a stronger team for future competitions.

6. Conclusion

This paper presents an approach for online detection and correction of subtle context-dependent inaccuracies in robots' models. Subtle context-dependent model inaccuracies arise in a wide variety of complex autonomous robot problems. Here, we have motivated the study of this problem with three domains: an illustrative simple golf-putting robot, whose model has inaccuracies due to imperceptible bumps on the field; a mobile service robot whose motion model is inaccurate due to a variety of internal and external reasons that affect it only in particular contexts; and a team of soccer-playing robots whose model of the opponent's likelihood of pass interception is inaccurate in some contexts, previously unknown to the researchers.

Owing to the subtlety of these model inaccuracies, the robots need to analyze sets of observations to find them, rather each observation in isolation. Owing to the context-dependency of the inaccuracies, the robots need to reason about the correlation between observations along various dimensions of their context space, such as time, position, and velocity.

We have presented an approach that explicitly searches for parametric RIMs to determine whether the robot's model is inaccurate in particular regions of its context space. This online approach relies on nonlinear optimization to find the region of context space that maximizes an anomaly value function anom, given the observations contained in that region. Once this region is found, the algorithm decides whether this region is truly anomalous based on a threshold, tuned to achieve a particular detection power. We have then presented an algorithm to detect multiple such anomalous regions, and to correct the robot's model based on such detections.

Empirical evidence shows the effectiveness of this algorithm at detecting subtle context-dependent model inaccuracies in our three test domains. In the simple golf-putting scenario, the algorithm successfully detected bumps in various locations of the field. In the CoBot's motion scenario, the algorithm successfully detected a failing wheel encoder, a collision, a corridor in which the CoBot's motion is flawed and a robot that has subtle problems when trying to turn left. Finally, in the robot soccer domain, the algorithm was able to detect inaccuracies in the passing model of the team, and to improve performance significantly within the time span of one soccer game, by correcting these detected inaccuracies. Remarkably, these inaccuracies were previously unknown to the researchers, and their detection has led to improvements in the CMDragons' models and algorithms.

Our three target domains provide evidence for the general applicability of our approach. Our algorithm can be applied to other model-based autonomous robot domains in which the robot's stochastic models can be verified by execution observations, and in which subtle context-dependent model inaccuracies may exist. Our future work will address the problem of applying these techniques to domains with very high-dimensional context spaces, such as early detection of anomalies in spacecraft with hundreds of sensors.

Funding

This material is based on work partially supported by the NSF (grant number IIS-1012733), DARPA (grant number FA87501220291), and MURI (subcontract 138803 of award N00014-09-1-1031). The presentation reflects only the views of the authors. This material is based upon research supported by (while Dr. Simmons was serving at) the National Science Foundation.

Notes

1. Because our work is strongly inspired by work on anomaly detection, we interchangeably use the terms *inaccuracy measure*, which reflects our specific purposes, and *anomaly measure*, which reflects the broader context for which this measure has been used.
2. In future work, we will also consider inaccuracies that affect the variance of observations rather than the mean.
3. Unfortunately, capital letter $\boldsymbol{\Sigma}$ is the standard symbol for both summations and covariance matrices. Although we have retained this notation, we try to avoid confusion by always placing summation indices under $\boldsymbol{\Sigma}$, whereas covariance matrices are indexed by a subscript to the right of $\boldsymbol{\Sigma}$.
4. When clear from context, we omit $\boldsymbol{Z}^x$ or $\boldsymbol{\theta}^0$ from $\text{anom}(R, \boldsymbol{Z}^x, \boldsymbol{\theta}^0)$.
5. For this paper, we have chosen to focus on ground passes, thus the third dimension of the world is ignored; also missing is the ball spin and the robots' internal states.

References

Antonelli G (2003) *A Survey of Fault Detection/Tolerance Strategies for AUVs and ROVs*. Berlin: Springer, pp. 109–127.

Bellman R (1957) A Markovian decision process. *Journal of Mathematics and Mechanics* 6: 679–684.

Biswas J, Mendoza JP, Zhu D, Choi B, Klee S and Veloso M (2014) Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team. In: *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Biswas J and Veloso M (2013) Localization and navigation of the CoBots over long-term deployments. *International Journal of Robotics Research* 32(14): 1679–1694.

Brafman RI and Tennenholtz M (2003) R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3: 213–231.

Browning B, Bruce J, Bowling M and Veloso M (2005) STP: Skills, tactics, and plays for multi-robot control in adversarial environments. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 219(1): 33–52.

Bu Y, Leung O, Fu A and Keogh E (2007) WAT: Finding top-*K* discords in time series database. In: *Proceedings of the 2007 SIAM International Conference on Data Mining*, Minneapolis, MN, pp. 449–454.

Chandola V, Banerjee A and Kumar V (2009) Anomaly detection: A survey. *ACM Computing Surveys* 41(3): 15:1–15:58.

Choi SPM, Yeung DY and Zhang NL (2001) Hidden-mode Markov decision processes for nonstationary sequential decision making. In: *Sequence Learning - Paradigms, Algorithms, and Applications*. London: Springer-Verlag, pp. 264–287.

Cordier MO, Dague P, Lévy F, Montmain J, Staroswiecki M and Travé-Massuyès L (2004) Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 34(5): 2163–2177.

da Silva BC, Basso EW, Bazzan ALC and Engel PM (2006) Dealing with non-stationary environments using context detection. In: *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*. New York: ACM Press, pp. 217–224.

Doya K and Samejima K (2002) Multiple model-based reinforcement learning. *Neural Computation* 14: 1347–1369.

Duczmal L and Assuncao R (2004) A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. *Computational Statistics and Data Analysis* 45(2): 269–286.

Felzenszwalb PF and Huttenlocher DP (2004) Efficient graph-based image segmentation. *International Journal of Computer Vision* 59(2): 167–181.

Fox M, Ghallab M, Infantes G and Long D (2006) Robot introspection through learned hidden Markov models. *Artificial Intelligence* 170(2): 59–113.

Hester T and Stone P (2013) TEXPLORE: Real-time sample-efficient reinforcement learning for robots. *Machine Learning* 90(3): 385–429.

Hwang I, Kim S, Kim Y and Seah CE (2010) A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology* 18(3): 636–653.

Infantes G, Ingrand F and Ghallab M (2006) Learning behaviors models for robot execution control. In: *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, Cumbria, pp. 394–397.

Jong NK and Stone P (2007) Model-based function approximation for reinforcement learning. In: *The Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kearns M and Singh S (2002) Near-optimal reinforcement learning in polynomial time. *Machine Learning* 49(2–3): 209–232.

Keogh E, Lin J and Fu A (2005) HOT SAX: Efficiently finding the most unusual time series subsequence. In: *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM '05)*, Washington, DC, pp. 226–233.

Keogh E, Lonardi S and Chiu B (2002) Finding surprising patterns in a time series database in linear time and space. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, New York.

Kulldorff M (1997) A spatial scan statistic. *Communications in Statistics - Theory and Methods* 26(6): 1481–1496.

Kulldorff M, Huang L, Pickle L and Duczmal L (2006) An elliptic spatial scan statistic. *Statistics in Medicine* 25(22): 3929–3943.

Mendoza JP (2017) *Regions of Inaccurate Modeling for Robot Anomaly Detection and Model Correction*. PhD Thesis, Carnegie Mellon University.

Mendoza JP, Biswas J, Cooksey P, et al. (2016a) Selectively reactive coordination for a team of robot soccer champions. In: *Proceedings of the Association for the Advancement of Artificial Intelligence Conference (AAAI)*, Phoenix, AZ.

Mendoza JP, Biswas J, Zhu D, et al. (2016b) CMDragons 2015: Coordinated offense and defense of the SSL champions. In: *RoboCup 2015: Robot World Cup XVIX*, Hefei.

Mendoza JP, Simmons R and Veloso M (2016c) Detection of subtle context-dependent model inaccuracies in high-dimensional robot domains. *Big Data* 4(4): 269–285.

Mendoza JP, Veloso M and Simmons R (2014) Focused optimization for online detection of anomalous regions. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Hong Kong.

Mendoza JP, Veloso M and Simmons R (2015) Detecting and correcting model anomalies in subspaces of robot planning domains. In: *Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Istambul, Turkey.

Neill DB (2006) *Detection of Spatial and Spatio-temporal Clusters*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

Neill DB (2012) Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74(2): 337–360.

Neill DB, McFowland E and Zheng H (2013) Fast subset scan for multivariate event detection. *Statistics in Medicine* 32(13): 2185–208.

Neill DB, Moore AW, Pereira F and Mitchell TM (2004) Detecting significant multidimensional spatial clusters. In: *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, BC.

Nikiforov IV (1995) A generalized change detection problem. *IEEE Transactions on Information Theory* 41(1): 171–187.

Nouri A and Littman ML (2009) Multi-resolution exploration in continuous spaces. In: D Koller, D Schuurmans, Y Bengio and L Bottou (eds.) *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc., pp. 1209–1216.

Page E (1954) Continuous inspection schemes. *Biometrika* 41(1): 100–115.

Pettersson O (2005) Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems* 53(2): 73–88.

Pettersson O, Karlsson L and Saffiotti A (2003) Model-free execution monitoring in behavior-based mobile robotics. In: *Proceedings of the International Conference on Advanced Robotics (ICAR)*, pp. 864–869.

Pettersson O, Karlsson L and Saffiotti A (2005) Model-free execution monitoring by learning from simulation. In: *Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, pp. 505–511.

Rubinstein R (1999) The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability* 1(2): 127–190.

Shi J and Malik J (2000) Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8): 888–905.

Slivkins A (2014) Contextual bandits with similarity information. *The Journal of Machine Learning Research* 15(1): 2533–2568.

Stone P, Kuhlmann G, Taylor ME and Liu Y (2005a) Keepaway soccer: From machine learning testbed to benchmark. In: *RoboCup-2005: Robot Soccer World Cup IX*, Osaka, pp. 93–105.

Stone P, Sutton RS and Kuhlmann G (2005b) Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior* 13(3): 165–188.

Tango T and Takahashi K (2005) A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics* 4(1): 11.

Veloso M, Biswas J, Coltin B, Rosenthal S and Ventura R (2012) CoBots: Collaborative robots servicing multi-floor buildings. In: *International Conference on Intelligent Robots and Systems (IROS)*, Algarve.

Wald A (1945) Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics* 16(2): 117–186.

Willsky A and Jones H (1976) A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control* 21(1): 108–112.