**Proceedings of the 2001 IEEE/RSJ
International Conference on Intelligent Robots and Systems
Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001**

Detecting Regime Changes with a Mobile Robot using Multiple Models

Dani Goldberg & Maja J Matarić
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
{dani|mataric}@usc.edu

Abstract

We present an approach to the detection of global environmental regime changes by a mobile robot performing a task. The approach is based on the use of augmented Markov models *(AMMs), a variation of semi-Markov process. We have developed an algorithm that constructs AMMs on-line and in real-time with little computational and space overhead. AMMs are a general tool for capturing the interaction dynamics between a robot and its environment using the history of behavior executed by the robot. We have demonstrated AMMs to be effective in applications such as fault detected, dynamic leader selection, and reward maximization in non-stationary environments. In this paper, we extend AMMs to regime detection, using multiple models to monitor events at different time scales and provide statistics to detect regime changes at those time scales. This approach has been successfully implemented using a physical mobile robot performing a land mine collection task. In the context of this task, we present experimental results, first validating our approach, then demonstrating a more complex proportion-maintaining scenario of the land mine collection task. Finally, we present results using an alternative reward maximization decision criterion in the same task.*

1 Introduction

In certain classes of mobile robot tasks, it may be necessary for a robot to detect significant global changes in the environment and modify its behavior or the task structure accordingly. The environment can be in a particular regime (i.e., a period of steady state) and then switch to a different regime requiring the robot to modify its behavior. Detecting such environmental regime changes may be difficult for a number of reasons:

- The robot may have no *a priori* knowledge of the environment and thus also lack a baseline for gauging environmental shifts. In a system where the environment is evolving (i.e., a non-stationary system), determining a basis for comparison may be difficult.
- Given only local sensing capabilities, the robot may require a significant amount of time to estimate the state of the environment. Any estimate of state, however, may be outdated in a non-stationary system.
- The nature of the task may be stochastic, with uncertainties large enough to preclude an effective predictive model of environmental state, or dynamics too complex to make the development of such a model feasible or tractable. Alternatively, however potentially simple the system, there may be no *a priori* data with which to instantiate a model.
- Depending on the task or environment, the time scale of the environmental change that must be detected may differ. For example, in one task, the environmental change may be almost instantaneous, detectable between one moment and the next. In another task, the change may be slow and incremental, requiring the examination of a large time interval for detection. Hard-coding the robot with a specific time scale to use for regime detection can be problematic. A time scale that is too small makes the robot incapable of detecting the change. Conversely, a time scale that is unnecessarily large increases the time required to detect the change and may be undesirable in time-critical situations.

As a concrete example, consider the task of collecting undetonated land mines in a field. Assume that there are two types of mines, large and small, with destructive power proportional to their size. A robot is given the following instructions: "Go out to the field and first collect as many large mines as you can, since they are the more destructive. But don't spend all of your time searching for every last large mine if you discover that there aren't many of them. Instead, start collecting the small mines. After all, we want to clear the field as well as possible." In order for the robot to accomplish this task, it must have enough data about its environment (the mine field) to intelligently switch from collecting large mines to small ones. In this scenario, the robot is only able to carry one mine at a time, producing a large cost (in time) for each mine collected. It is important that the more critical large mines be collected first, but that the robot be able to decide when to switch to the smaller mines. (Here we assume that the task requires the robot to collect one type of mine at a time. Alternatively, the robot might switch between types as necessary. We explore this alternative when we consider a reward maximization scenario later in the paper.)

The difficulty of this task is compounded when the issues mentioned above apply. The robot may have no *a priori* information about the numbers of large and small mines in the field, their distributions, or relative proportions. The robot may also lack global sensing of the mines in the field and may not know the time scale appropriate to its decision for switching between mine types. This decision is dependent on factors including the size of the field and the relative densities of the two types of mines.

In this paper, we propose a mechanism for regime detection that resolves the above issues. The approach uses multiple *augmented Markov models* (AMMs). It builds upon previous work [1] successfully demonstrating the use of AMMs in multi-robot coordination applications such as fault detection, determination of group affiliation, and dynamic selection of the best leader in a group. This work demonstrates regime detection as another, potentially concurrent, application of the model. The AMMs are used to capture, in real time, the dynamics of a robot interacting with its environment in terms of the behaviors it performs. One AMM is created and maintained at each time scale that is monitored, and statistics about the environment at that time scale are derived from it. As task execution continues, AMMs are dynamically generated to accommodate the increasing time intervals. Sets of statistics from the models are used to determine whether the environmental regime has changed. This approach requires no *a priori* knowledge, uses only local sensing, and captures the notion of time scale. (With *a priori* knowledge, other techniques such as simple thresholding may also be applicable.) Additionally, it works naturally with stochastic task domains where variations between trials may change the most appropriate time scale for regime detection. The approach has been physically realized on a mobile robot performing the mine collection task. Experiments and results for this task are presented later in the paper.

It should be noted that it is difficult to define an absolute notion of regimes, especially since it relates to the dynamics of environmental changes. In a gradually shifting environment, the designation of a regime change can be fairly arbitrary. In this paper, we propose one principled method for regime detection based on statistical hypothesis tests, and empirically show it to be effective. In the next section, we present AMMs, then follow with a discussion of their use in regime detection.

2 Augmented Markov Models

An augmented Markov model (AMM) is a variation of a semi-Markov process (SMP) [2] in which the probability distribution associated with the time spent in a particular state is identical regardless of the subsequent state. This is similar to a Markov chain [3], except that Markov chains constrain the distribution to be geometric, while a semi-Markov process allows arbitrary distributions.

AMMs provide a compromise between the generality of SMPs and the computational simplicity of Markov chains. They allow standard expectation calculations from Markov chain theory to be easily combined with statistical hypothesis tests such as the (parametric) t and F tests, or their nonparametric counterparts. Though not suited to every problem, AMMs are a general modeling tool that is more widely applicable than modeling the interaction dynamics between a mobile robot and its environment, as has been our focus.

An AMM is defined as a five-tuple $\langle S, A, B, L, T\rangle$:

1. S, a set of symbols $\{s_1, s_2, \ldots, s_M\}$.
2. A, a set of states (or nodes) $\{a_1, a_2, \ldots, a_N\}$, each of which recognizes one of the symbols in S.
3. B, an $N \times M$ transition matrix.
4. L, a set of directed links $\{l_1, l_2, \ldots, l_P\}$, connecting the states.
5. T, a set of elements $\{t_1, t_2, \ldots, t_Q\}$, each of which stores information on a particular two-link traversal sequence entering and leaving a state.

The data used for constructing an AMM consist of a stream of symbols belonging to S. In our implemented robot behavior model, the symbols would be the names of behaviors that the robot performs, sampled at some appropriately high frequency to avoid aliasing. At each time step, the data used for generating and updating an AMM consist simply of the symbol indicating which behavior in the *behavior space* is currently active. The *behavior space* is designated as a set of mutually exclusive behaviors, which continuously describe the robot's activity, and are uniquely labeled.

[1] and [4] provide further details of AMM representation, AMM usage, and model generation using a computationally efficient algorithm. One point of interest regarding the AMM generation algorithm is its ability to represent non-first-order Markovian systems in first-order form by dynamically adjusting the model using higher-order statistics on the behavior of the system. This simplifies many of the calculations on AMMs.

In the next section we describe how AMMs may be used as part of a mechanism for regime detection.

3 AMMs for Regime Detection

In the context of mobile robotics, the system we consider is the robot and its environment. Our focus is on the difficult but realistic situation in which the robot lacks *a priori* information about its environment, an environmental model, and global sensing. In such a situation, the robot may require a relatively large amount of time to detect a trend that signals a global environmental regime change. This is especially the case if the system is noisy and stochastic, as is generally true for mobile robotics. Unless a sufficiently large time scale is employed, the regime change may be lost in the variation of the data. Determining the appropriate time scale, however, may not be possible ahead of time. It may be dependent on the exact nature of the task, the structure of the environment (including the presence of other robots), and the nature of the system's stochasticity. The time scale may also dependent on the specific attributes of the system being monitored for regime changes.

In order to negotiate these challenges and endow a robot with the ability to detect global environmental regime changes, we maintain models (AMMs) of the robot's interaction with its environment at multiple time scales. As the robot performs its task, we extract and store particular statistics from these models, which are used to detect a specific regime change based on a sound criterion of significance. In our first experimental validation, the regime switch is detected as a significant change in the density of mines, while in the proportion-maintaining scenario it is detected as a change in the proportion of mines. Before presenting the algorithm for regime detection, we first introduce some notation used in the algorithm.

3.1 Notation

- Let $\tau > 0$ be the minimum time scale, or number of input symbols, used to construct an AMM.
- Let f_i be a positive valued function of τ returning the size of (number of input symbols maintained by) the i-th time scale.
- Let $k > 0$ specify the number of AMM-extracted values used in detecting a regime.
- Let the AMM at time scale f_i be m_i.
- Let Q_i be a sequence of at most k statistics for model m_i.
- Let n be the total number of input symbols that have been used to construct the models.
- Let M be a special AMM that is constructed using all of the input symbols that have been seen.

This notion is now employed in the regime detection algorithm.

3.2 Algorithm for Regime Detection

1. Initialize M, m_0, and set $n \leftarrow 0$.
2. Get an input symbol and use it to update M and all m_i.
3. Set $n \leftarrow n + 1$.
4. For all i such that $(n \bmod f_i) = 0$
   (a) If no such m_i exists due to the fact that a new time scale has been reached, then create m_i and initialize it to equal M.
   (b) Call Stat(m_i) to get the statistic for the model and insert that value into Q_i.
   (c) If the length of Q_i equals k, then call DetectRegime(Q_i).
   (d) If DetectRegime(Q_i) returns true, then the regime has changed, else it has not.
   (e) Re-initialize m_i to be an empty model.

Stat() is a function on an AMM, returning application-dependent statistics extracted from the model (e.g., the mean time in a state/behavior). DetectRegime() performs a statistical hypothesis test (such as Student's t or ANOVA) on a list of values. DetectRegime() returns true if the result is significant, false otherwise. Essentially, DetectRegime() provides a *meta-threshold* based on a statistical hypothesis test. The threshold is not a set number, but rather a measure of the statistical significance of the shift in the environment.

The algorithm maintains multiple AMMs at different time scales. At each time step, each AMM (m_i and M) is updated with a new input symbol. If no m_i exists for a new, larger time scale (f_i), then that model is created and initialized to M. If a model m_i has received its maximum number of input symbols (as designated by f_i), then Stat(m_i) is called to extract the appropriate application-dependent data from it, and m_i is reinitialized to be empty. The data from m_i is inserted into a queue Q_i of maximum length k, and if $\mid Q_i \mid = k$ then DetectRegime(Q_i) is called to test for significant differences in the values of Q_i. It is just such a significant difference or shift in the data which is designated as a regime change.

In the next section we describe our experimental setup and example.

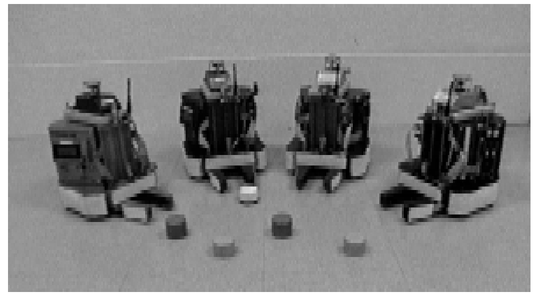

Figure 1: The Don Group: four IS Robotics R2e robots used individually in the experiments.

4 Land Mine Collection Task

To validate our algorithm for detecting global environmental regime changes, we used a task analogous to the land mine collection example of the Introduction.

4.1 The Robot

We use one IS Robotics R2e robot (Figure 1) in the experiments. The robot is a differentially-steered base equipped with two drive motors and a two-fingered gripper. The sensing capabilities include piezo-electric contact sensors around the base and in the gripper, five infrared (IR) sensors around the body and one on each finger, a color sensor in the gripper, a radio transmitter/receiver for communication and data gathering, and an ultrasound triangulation system for positioning. The robot is programmed in the Behavior Language [5]. Experiments are performed in a rectangular enclosure (the Corrall) that was adjusted to be either 11×14 feet or 11×8 feet, depending on the experiment. The Corrall had up to 36 small plastic cylinders (pucks) of two different colors: clear (representing large mines) and black (representing small ones). The pucks were evenly distributed throughout the Corrall, except in the drop-off area, called Home, a ninety degree sector of a circle with a radius of two feet, located in one corner (Figure 2).

4.2 Behaviors

The Behavior Language, used for programming the robots, provides a natural structure for implementing behavior-based controllers. Behaviors are defined as sets of concurrent, real-time, asynchronous rules receiving inputs from sensors and other behaviors, and sending outputs to actuators and other behaviors. A set of these behaviors can often trivially map to the behavior space for an AMM, as is the case with our experimental mine collection task. The following 9 behaviors constitute the behavior space for our task:

- **avoiding**: avoid any object (detected by IR and contact sensors) deemed to be in the path of the robot.
- **wandering**: move forward and, at random intervals, turn left or right through some random arc.
- **puck detecting**: if avoiding is not active, and if an object is detected by the front IRs, lift up the gripper fingers

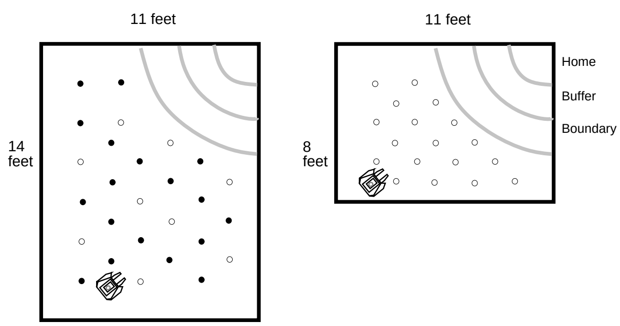

Figure 2: Two versions of the mine collection task environment: (Left) 11 × 14 foot Corrall with 9 clear and 18 black pucks; (Right) 11 × 8 foot Corrall with 18 clear pucks.

to determine whether the object is short enough to be a puck. If it is, approach the object and try to place it between the fingers and pick it up. If unsuccessful, perform **avoiding**.

- **color detecting**: if **puck detecting** is successful, detect the color of the puck. If it is the desired color, then perform **homing**, else perform **leave puck**.
- **leave puck**: drop the puck and continue searching for more, using **avoiding**, **wandering** and **puck detecting**.
- **homing**: if carrying a puck, move towards the designated goal, Home.
- **creeping**: when near Home, perform a slower, more accurate homing behavior.
- **exiting**: if in the Home region, drop puck and exit Home.
- **reverse homing**: move away from the Home region.

Control of the robot's drive motors was the basis for selecting the constituent members of this behavior space. When active, each of the behaviors has exclusive control of the motors, and together they account for all activity (or inactivity) of the motors for the duration of the task.

4.3 Validating the Approach

In order to validate our approach to regime detection, we show that: (1) regime changes do happen at different time scales, and (2) our algorithm using multiple AMMs can detect such changes, as brought about by shifts in large mine density. We compare results from two versions of the mine collection task that are identical except for the environmental setup. The hypothesis is that the decrease in environment size and the increase in clear puck (large mine) density in the second version pushes the regime change to a different time scale, most likely smaller.

The first version of the task uses an 11 × 14 foot (large) Corrall with 9 clear and 18 black pucks evenly distributed throughout (Figure 2: Left). With no *a priori* information about the environment, the robot must collect only the clear pucks (i.e., large mines), while executing the regime detection algorithm to determine when to switch to black pucks (i.e., small mines). (In reality, data were sent via a serial radio link to an off-board Power Macintosh G3/266 which performed the regime detection algorithm and notified the robot of any regime changes. This was done because programming limitations of the R2's and the Behavior Language made implementing the algorithm on-board an R2 extremely difficult. These limitations are platform-specific and do not exist on many, even low cost, mobile robots. Note that although the regime detection computations were performed off-board, they were done in on-line and in real-time.) In the second version, the Corrall is decreased in size to 11 × 8 feet (small) and only 18 clear pucks are used (Figure 2: Right). The key statistic of interest in these two versions is the time scale at which the robot detects a regime change and decides to begin collecting black pucks (small mines).

We complete the description of the validation experiment by presenting the parameter values used in the regime detection algorithm: the minimum time scale $\tau = 5$; the number of statistics kept for each model was $k = 8$; function $f_i = 2^i\tau$; Stat(m_i) returned the number of pucks that had been collected during the lifetime of AMM m_i; and DetectRegime(Q_i) performed an analysis of variance (ANOVA) on two groups of data (namely, the first and second $\frac{k}{2}$ values in Q_i), to determine if the means were different at a significance level of 10%. Since in each trial the robot was initialized to collect clear pucks (large mines), DetectRegime(Q_i) essentially determined if the number of clear pucks changed significantly enough over k consecutive intervals of size f_i to indicate a regime change.

We conducted five experimental trials in each of the two environments and gathered data about the time scale at which regime detection occurred. In each of the 10 trials, the algorithm successfully detected a regime switch. In the large Corrall environment, the mean time scale of detection was 1024, while in the small Corrall it was 256. (Since data were collected at 2 Hz, this translates to approximately 512 seconds and 128 seconds, respectively.) A hypothesis test based on Student's t distribution [6] indicates that the two means are statistically different at a significance level of 1%. Thus, we have validated our approach by showing that regime changes do occur at different time scales (even in the same task but with different environments), and that our algorithm is able to detect such changes. Next, we describe a more sophisticated use of our approach.

4.4 Maintaining the Proportion of Mines

In a more complex version of the mine collection task, the robot is required to maintain the proportion of large to small mines in the environment at a specified value p. A significant switch in this value indicates a non-local regime switch, since p itself is a non-local measure. Once again, the robot begins by collecting large mines, but this time switches to small mines when the observed proportion p_{obs} is significantly different from p, and $p_{obs} < p$. Conversely, the robot switches back to large mines when $p_{obs} > p$ and this difference is significant. The goal of this experiment was to determine whether the robot could detect multiple consecutive regime changes in its environment due to shifts in the proportion of large to small mines.

For this experiment, the Corral was 11 × 8 feet and con-

Puck type	Trial #				
	1	2	3	4	5
Clear pucks	4	8	15	10	14
Black pucks	8	4	16	9	10

Table 1: Pucks remaining in the environment at the end of each trial of the proportion maintaining mine collection task.

tained 18 each of clear and black pucks. The parameter values used in the regime detection algorithm were: $\tau = 5$; $k = 4$; $f(i) = 2^i\tau$; Stat(m_i) returned the proportion of clear to black pucks encountered; and DetectRegime(Q_i) performed an analysis of variance (ANOVA) at a significance level of 10% on Q_i and a list of length k having all values equal to p. The proportion p was set to 1.0, indicating that the robot should try to maintain equal numbers of the two types of mines. Whenever a regime switch was detected, the regime detection algorithm was re-initialized so as to be able to detect the next regime change.

In this experiment, a trial was considered complete when the robot detected two consecutive regime changes. The robot successfully did so in each of the five trials that were conducted. Table 1 shows the numbers of clear and black pucks remaining in the environment at the end of each trial. The correlation between the numbers is quite large ($\rho = 0.70$) and indicates that their proportion tended to be close to 1.0. Thus, not only was the robot able to detect multiple consecutive regime changes, but was also effective in maintaining the desired proportion of pucks (mines).

4.5 Maximizing Reward

We now present results for a third experimental scenario, requiring the robot to maximize the expected reward garnered from collecting mines. Instead of designating *a priori* that the robot begin by collecting large mines (as in the previous experiments), here the robot is told the reward value associated with each type of mine and must decide which to collect in order to maximize its total reward. Reward values are set in proportion to a mine's explosive power, thus making reward maximization identical to minimizing the mine field's destructive potential.

We now more formally define the problem. Let $\mathcal{R}_s$ and $\mathcal{R}_l$ be the rewards for small and large mines, respectively. Let τ_s^f be the expected time required for the robot to find a small mine, and let τ_s^d be the expected time to deliver the mine to the goal location once it has been found. Similarly, τ_l^f and τ_l^d represent these times for large mines. The robot maximizes its reward by deciding for each mine found whether to deliver it or leave it in search of a higher valued mine. The action chosen is the one that maximizes the expected reward per unit time (and thus the overall expected reward). If the robot finds a small mine, and the inequality

$$\frac{\mathcal{R}_s}{\tau_s^d} > \frac{\mathcal{R}_l}{\tau_l^f + \tau_l^d}$$

holds, then delivering the small mine maximizes reward. Otherwise, the small mine should be left and a large mine sought. The complementary inequality is used when a large mine is found. The

The main issue in evaluating the inequality is calculating τ^d and τ^f for each mine type. One could maintain internal variables that record these values. Our approach, however, is to calculate these values from the robot's AMM, providing an additional concurrent use of the model. First, the stochastic transition matrix, P, of the Markov chain designated by the AMM is extracted. In order to continue with the calculation, P must be ergodic (i.e. any state must be reachable from any other state). The requirement of regularity is usually met in repetitive tasks such as mine collection.

Next, the fixed point probability vector w indicating the proportion of time spent in each state is calculated from $wP = w$. The fundamental matrix, Z, is now calculated as $Z = [I - (P - W)]^{-1}$, where each row of W is w. Finally, Z is used in the calculation of the mean first passage matrix $E = (I - Z + JZ_{\text{dg}})D$, where J is a matrix of ones, Z_{dg} is the main diagonal of Z, and D is the diagonal matrix with $d_{ii} = 1/w_i$. ([3] provides more detail about Markov chains and these computations.) Each element, e_{ij}, of E gives the mean time (number of steps) to get from state i to state j. E also contains the values for τ^f and τ^d. τ^f is simply the entry in E associated with the minimum mean time from the **wandering** state to the **color detecting** state, and τ^d is the minimum mean time from the **color detecting** state to the **wandering** state, for each color of puck.

Regime detection enters the scenario when the environment is non-stationary, i.e., pucks are not replenished. As the environment changes, average values of τ^f and τ^d become inaccurate and thus not effective for reward maximization. At some point, the robot must decide that the environmental regime has changed and that it is appropriate to use more current values for τ^f and τ^d. [7] describes the algorithm used to maintain current values for τ and more detail about the experiments below. We have tested the efficacy of our approach to reward maximization in both abruptly and gradually shifting non-stationary environments.

In our experiments with abruptly changing non-stationary environments, the Corrall initially contained 18 clear pucks (large mines), which after 20 minutes were replaced by black pucks (small mines). The goal was to determine whether the robot adapted more quickly to this regime change when it was able to compensate for the shift in the environment by maintaining up-to-date statistics. We ran two experiments using reward maximization with adaptation to regimes, and three experiments using reward maximization alone. In these experiments, the reward for a white puck was 7 points and the reward for a black puck was 1 point. For reward maximization alone, the mean time to adaptation was 47 minutes, while with adaptation it was 18.3 minutes. A t-test indicates that these means are different at a significance level of 0.01. This strongly supports the importance of being able to accommodate environmental regimes in non-stationary environments.

In other experiments, we tested the effectiveness of regime adaptation for reward maximization in a gradually shifting environment initialized to contain 18 clear pucks worth 10 points each, and 18 black pucks worth 1 point each. Instead of abruptly changing the color of the pucks, as in the previous experiment, here the environment shifted slowly as the robot collected pucks. Due to the high degree of variance in the mine collection task using physical robots, we anticipated that the number of experiments we would have to conduct in order to obtain statistical significance in the gradually shifting environment would pose a practical impossibility. The experiments were therefore conducted in a simulation of the mine collection task. Once again, we found that maintaining current statistics by accommodating environmental shifts allowed the robot to perform better. We calculate the number of reward points that the robot was expected to have accrued on average during the course of a trial: 149.3 with no adaptation, and 153.8 with adaptation. These data are significantly different at a level of 0.02 using a two-tail version of Student's t test.

5 Related Work

We confine our review to a small selection of the most relevant work. Various modeling approaches have been employed on mobile robot platforms to date. [8] uses a partially observable Markov decision process (POMDP) to model uncertainty of location in a robot navigation task for an office environment. [9] employs a POMDP to model sensor, actuator, and metric uncertainties in a similar office navigation task. [10] presents a learning technique that uses a tree-like behavioral model of a robot's interaction with the environment. [11] presents Discrete Event Systems (DES) with emphasis on applications to mobile robotics. Unlike much of the above work which requires multiple trials to achieve convergence, the focus in our AMM work in mobile robotics is real-time response, i.e., results within one trial.

Related to our approach in this paper is other work in regime detection, and information representation at different time scales. [12] uses a network of nonlinear gated experts to detect regime changes in a time series. [13] considers stationary regime detection in a finite Markov process as part of the larger problem of availability analysis in a computer system. [14] discusses the importance of representing knowledge at multiple time scales in planning and decision making, and uses an approach based on Markov decision processes and semi-Markov processes.

To the best of our knowledge, our work is the first demonstration of on-line robot regime detection using multiple models.

6 Conclusions

We have presented a novel approach that enables a robot to detect and respond to global environmental regime changes having no *a priori* knowledge or models of the environment, and limited to only local sensing. The approach is based on augmented Markov models, a general modeling tool we developed. In this work, multiple AMMs were constructed at different time scales and used to derive sets of statistics that were analyzed to detect a regime change. We have experimentally validated the approach on a physical mobile robot performing a land mine collection task and shown that regime changes do occur at different time scales. We then demonstrated the efficacy of the approach in a more complex mine proportion-maintaining scenario, and described experiments using an alternative reward maximization decision criterion.

Acknowledgments

The research reported here was conducted at the Interaction Lab, part of the Robotics Research Labs at the University of Southern California Computer Science Department. The work was supported in part by the Office of Naval Research Grant N00014-95-1-0759 and N00014-00-1-0140, by the Sandia National Labs Grant 3065, and by a grant from the DARPA TASK program.

References

[1] Dani Goldberg and Maja J Matarić, "Coordinating mobile robot group behavior using a model of interaction dynamics," in *Proceedings, The Third International Conference on Autonomous Agents (Agents '99)*, Seattle, Washington, May 1–5 1999, pp. 100–107, ACM Press.

[2] Sheldon M. Ross, *Applied Probability Models with Optimization Applications*, Dover Publications, Inc., New York, 1992.

[3] Fred S. Roberts, *Discrete Mathematical Models: With Applications to Social, Biological, and Environmental Problems*, Prentice-Hall, Inc., 1976.

[4] Dani Goldberg and Maja J Matarić, "Augmented markov models," Tech. Rep. IRIS-99-367, University of Southern California, 1999.

[5] Rodney A. Brooks, "The behavior language; user's guide," Tech. Rep. AIM-1227, MIT AI Lab, 1990.

[6] John E. Freund, *Mathematical Statistics*, Prentice Hall, fifth edition, 1992.

[7] Dani Goldberg and Maja J Matarić, "Learning multiple models for reward maximization," in *Proceedings, The Seventeenth International Conference on Machine Learning (ICML-2000)*. 2000, pp. 319–326, Morgan Kaufman Publishers.

[8] Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman, "Acting optimally in partially observable stochastic domains," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, 1994, pp. 1023–1028.

[9] Sven Koenig and Reid G. Simmons, "Unsupervised learning of probabilistic models for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996, vol. 3, pp. 2301–2308.

[10] François Michaud and Maja J Matarić, "Learning from history for behavior-based mobile robots in non-stationary conditions," *Autonomous Robots*, vol. 5, no. 3–4, pp. 335–354, July/August 1998.

[11] Jana Košecká and Ruzena Bajcsy, "Discrete event systems for autonomous mobile agents," in *Proceedings of the Symposium on Intelligent Robotic Systems*, 1993, pp. 21–31.

[12] Andreas S. Weigend, Morgan Mangeas, and Ashok N. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting," *International Journal of Neural Systems*, vol. 6, pp. 373–399, 1995.

[13] Bruno Sericola, "Availability analysis and stationary regime detection of markov processes," Tech. Rep. 2886, INRIA, France, May 1996.

[14] Richard S. Sutton, Doina Precup, and Satinder Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, pp. 181–211, August 1999.