Automated Robot Behavior Recognition Applied to Robotic Soccer

Kwun Han **Manuela Veloso**
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
{kwunh,veloso}+@cs.cmu.edu
http://www.cs.cmu.edu/{~kwunh,~mmv}

Abstract

Automated recognition of the behavior of robots is increasingly needed in a variety of tasks, as we develop more autonomous robots and general information processing agents. For example, in environments with multiple autonomous robots, a robot may need to make decisions based on the behavior of the other robots. As another interesting example, an intelligent narrator agent observing a robot will need to automatically identify the robot's behaviors. In this paper, we introduce a novel framework for using Hidden Markov Models (HMMs) to represent and recognize strategic behaviors of robotic agents. We first introduce and characterize the perceived signal in terms of behavioral-relevant state features. We then show how several HMMs capture different defined robot behaviors. Finally we present the HMM-based recognition algorithm which orchestrates and selects the appropriate HMMs in real time. We use the multi-robot robotic soccer domain as the substrate of our empirical validation, both in simulation and using real robots. Robots will then adapt their behaviors as a function of the autonomously recognized behavior of the other agents, either teammates or opponents.

1 Introduction

Intelligent autonomous robots perform tasks according to different behaviors [4; 9; 1; 12]. We can assume that in general, each robot will act according to a set of behaviors, either fully or partially pre-defined. Suppose now that an agent has the ability to observe a robot and to autonomously identify which behavior the robot is performing. This is the question that drives the work that we report in this paper.

We situate our work in the domain of robotic soccer, in which it is potentially advantageous to be able to recognize, model, and predict the other agents' behaviors.

Robotic soccer has been developed in simulation and with real robots [7]. Important aspects of opponent modeling have been developed in simulation [11]. The real robots are physical robotic agent computer controlled either onboard or offboard through radio communication. In both simulation and real robots, the agents must be fully autonomous, as human operators are not allowed to interfere once the game has started. In general, robotic or simulation agents observe the state of the environment, including the position of the other robots, and select actions based on the observed state. The decision making is mainly Markovian, where possibly limited memory processing of past states is inferred and included in the agent's internal state [3]. In general, these state observations and predictions are at the *low-level* of actions of the robots, namely coordinate current or predicted location. Our concern is at the level of extracting *high-level* strategy behaviors from the observed data, which clearly cannot be done in a Markovian way. Examples of such behaviors are "go-to-ball," "blocking robot," and "go-to-defend."

We see three main tasks for which the autonomous analysis of an agent's behavior is beneficial:

1. Adaptable Strategy

   Teams of robots and individual robots can change their game-playing strategy as a function of the recognized behavior of the other agents. Automatic recognition of the team strategy is a challenging problem. It represents an instance of the general plan recognition problem within uncertain and non-deterministic domains. In addition, we aim at representing and recognizing multi-agent behaviors.

2. Narrative Agents

   Several researchers are aiming at developing autonomous narrative agents, (eg.[2]). These agents need to "detect" what the soccer agents are doing at the high behavioral level to provide interesting narrations, e.g. *"The midfielder is dribbling the ball; The goalkeeper is aligning itself with the ball; The ball is passed to the right-wing attacker, which intercepts the ball and shoots. Goal!"*.

This paper contributes the use of Hidden Markov models to the representation and recognition of robots' behaviors. Observations are viewed as a perceived robot performance "signal." Our processing algorithm is inspired at a high conceptual level of abstraction level in other temporal nondeterministic signal understanding tasks, such as speech or music recognition. To our best knowledge, our work is novel in casting and developing representation and recognition of robot behavior (towards multi-agent robot behavior in particular) within a Hidden Markov Model based approach. However, Hidden Markov Models have been used in other recognition tasks, speech recognition[10], gesture recognition[8] and robot skill learning[6].

The paper is organized as follows. Section 2 discusses the behavioral model that we follow. It also then briefly presents the robotic soccer framework and in particular it describes the robot perceived signal that is used for behavior recognition. Section 3 introduces the behavior representation using Hidden Markov Models, which it illustrates with several examples. It formally introduces a general *Behavioral Hidden Markov Model* which is independent of the particular domain used. Section 4 presents in detail the recognition algorithm. Section 5 shows experimental illustrations of our approach with multiple behaviors. The experiments are performed both in simulation and with real robots. Section 6 concludes the paper.

2 Substrate: Behavioral Approach and Input Robot Signal

Let R be the robot being observed and O being the observing robot. The agent R is situated in its operating environment, or the world and it acts in reaction to the environment. O observes R and does not know its internal states. O's goal is to infer R's "strategic behavior" from what it perceives of R's physical actions and the environmental states.

This recognition goal may appear at first as a futile process since, without knowledge of R's internal state, there are endless possibilities of what behavior R is really performing. Therefore we need to state the problem as a *behavior membership decision*, under the following assumption:

> R acts according to a known set of behaviors $h(i)$. While acting in its environment, it reactively chooses a behavior that matches the current state. O has a model of the set of possible behaviors. O needs to "decide," i.e., to *recognize*, which $h(i)$ R is performing.

We believe that HMMs are suitable for this recognition process, as the internal state of the observed agent R is naturally "hidden." Environmental state features provide observations that can be used in the HMMs. We now introduce the perceived state in the robotic soccer.

In a robotic soccer game, there are $2n$ mobile robots (where n is the number of robots in each team) and one moving ball. A global vision system is used to capture the location of the objects in the soccer field [5]. We characterize this as a time dependent function: $\overline{f}(t)$. For two teams of robots in the game, each robot's state is encoded into x, y location and orientation. The ball's state is encoded into x, y location and velocity. Therefore, the dimensionality of $\overline{f}(t)$ is $2 \times 3 \times n + 4$.

This provides a discretization of $\overline{f}(t)$. We now have a $6n + 4$ dimensions time series $\overline{f}[t]$. The time granularity of $\overline{f}[t]$ depends on the speed of the vision algorithm, in our case, $1/30$ sec lapses between successive readings [5].

The goal of the recognition algorithm is to take this time series $\overline{f}[t]$ as input, and generate a scene description. Since one of the major application of such an algorithm is to generate real-time commentating, the algorithm must be an online algorithm.

3 Representing Behaviors as Hidden Markov Models

In this section, we overview Hidden Markov Model and then we describe how we adapted the model to behavior recognition.

In the Hidden Markov Model framework, the system is represented as a set of discrete states. At each time step, the system is in one of N states ($\{s_i\}$). A state transition takes place according to a certain probability distribution. The transition probability $Pr(S_{t+1} = s_j | S_t = s_i)$ (ie. for the state transition $s_i \rightarrow s_j$) is denoted by a_{ij}.

However, the state of the system at time t, S_t, is not directly observable. Instead, a set of state dependent observation variables, o_i, are available. The variables o_i are not necessarily discrete, it can either be from a set ($\{o_i\}$) or from a multi-dimensional continuous distribution (R^m). For each state s_i, an observation probability $b_i(o)$ is defined over o, be it discrete or continuous.

The robot soccer domain is naturally multi-agent. However, the complexity of the agent behavior can increase exponentially with the number of agents. For this work founding the robot behavior of robot recognition, we limit the number of agents to be one and the ball.

We will now proceed to describe applying HMM to behavior recognition in the robotic soccer domain.

3.1 Unobservables in an observable environment

As we have seen, a Hidden Markov Model consists of a discrete set of states and a discrete set of observations. In our application of HMM to behavior recognition, the set of Markov states corresponds to a model of the *mental state* of the agents and the set of observations correspond to the physical state of the agents.

This can be more clearly explained if we consider the behavior generation process. Each agent situated in its environment must make decision according to its sensor inputs. Most complex agents' behavior can be broken down into stages. Eg. If a robot is trying to execute the behavior "go behind the ball," the stages corresponding to this behavior can be {beginning of behavior execution, rotating towards ball, in front of ball, beside the ball, behind the ball}. It is worth noting that these stages of a behavior may not be explicitly described in the implementation of the behavior. Nonetheless, it provides the modeled abstraction of the potentially complex behavior.

The HMM for recognizing the behavior will consist of these stages, represented as states. Probabilistic transitions between any two states models the noise in the system and any discrepancies between the model and the actual behavior.

Even though some of the Markov states appear to map to physical location/state of the agent, it is worth noting that this is not necessarily the case. Physical location and Markov states are distinct.

It is possible to model all behaviors together as one hidden Markov model, however, the resulting model is unnecessarily complex. Our algorithm uses one HMM for each behavior, with them executing concurrently during recognition.

3.2 HMM Structure for a behavior

A Behavior HMM consists of four different types of states, namely:

- **Initial States** – States in which a robot will be at the start of the execution of the behavior. For complex behaviors, it is possible to have more than one initial states.
- **Accept States** – Upon successful completion of a behavior, the robot will be at the accept state. For the recognizer, accept states serve as the signal for a successful detection.
- **Intermediate states** – A complex behavior may have any number of intermediate stages. These stages are captured in a Behavior HMM by the intermediate states. Normally, to reach an accept state from the initial states, one must transverse through intermediate states.
- **Reject States** – A naive behavior recognizer will give ambiguous output when given a observation not generated by its behavior. We propose a semi-combined HMM that *partially* includes other behaviors.

  A reject state serves two different purposes: 1) It labels states that the behavior definitely should not reach. For example, for the behavior Go-Behind-Ball and the recognizer detected that the robot is in front of the ball, then the recognizer should give a reject state a high probability, signifying that what the robot is doing is highly unlikely to be Go-Behind-Ball; 2) All other states not relevant to this behavior can be collapsed into one single reject state. Falling into these states during recognition means that the sequence do not conform to the definition of this behavior according to this HMM.

As an example, Figure 1 shows the Hidden Markov model representation of the behavior "Go-To-Ball". The Go-To-Ball Behavior HMM has 4 states, S1 - S4. 4 types of state are present: S1 is an initial state, S2 is an intermediate state, S3 is an accept states, S4 is a reject state.

The initial state, S1, captures the state in which the robot is far away from the ball. As the robot moves closer to the ball, it will be captured by the transition to S2. When the robot is at state S2, it can either continue move closer towards the ball, in which case, it will transit to S3. Alternatively, if the robot moves away from the ball at this point, this will be captured in S4. S4 is a reject state because when a robot moves close to a ball and then moves away from it, it is not a Go-To-Ball behavior. S3 is the accepting state. This represents the completion of the behavior.

The robot can transit from state to state, the possible state transitions are denoted by the arrows. The probabilities for these transitions are predefined.

The observations o1 – o4 are identified according to Figure 2.

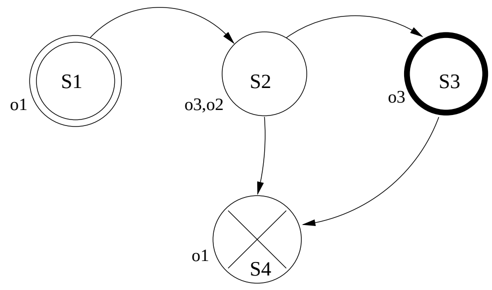

Figure 1: The "Go-To-Ball" behavior represented as a Behavior HMM

3.3 Observation feature extraction

The observations o_i are generated from the location of the objects in the field. The nature of the generation method is dependent on the behavior in which we want to recognize. However, it generally falls into these categories:

- **Absolute position** – The (x, y, θ) location of an object is used to directly compute o_i. Since o_i are discrete, the continuous (x, y, θ) space is segmented into regions, each corresponds to one of o_i.
- **Object relative** – The relative location of an object relative to another object is used to compute o_i. This is often used in a behavior which is ball-centric, eg. Go-To-Ball and Go-Behind-Ball. In other words, we use $(x_{robot} - x_{ball}, y_{robot} - y_{ball}, \theta)$ as our base continuous value, and then segment this space into finite regions, as described above.
- **Dynamic** – The dynamic properties of an object are used to compute o_i. Often, as in our case, robot behaviors are guided by some predictive measures such as an Extended Kalman Filter [5], to recognize such behaviors, the recognizer need to employ similar techniques. An example is the Intercept-Ball behavior, in which the ball's velocity is used to segment the robot's relative location into regions.

For each state, there is a probability in which we will see a certain observation. This is captured in the observation probability, $\{b_i(o_k)\}$.

Figure 2 shows the ball-relative observations for the Go-To-Ball Behavior HMM. The region relative to the ball is segmented into three regions by two concentric circles. The regions are labelled o_1, o_2 and o_3. The robot locating inside each of the region will trigger the respective observation.

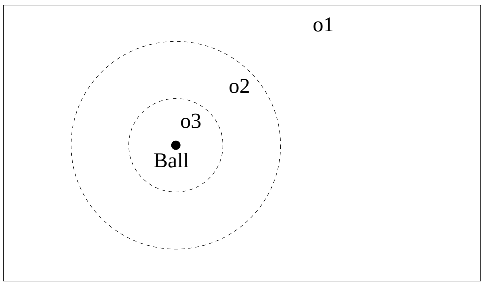

Figure 2: Observations for the "Go-To-ball" Behavior HMM.

3.4 Formalizing a Behavior HMM

We now formalize an augmented version of an HMM that we call a *Behavior Hidden Markov Model* (BHMM).

A Behavior Hidden Markov Model is the 5-tuple (λ):

- $N = \{s_{initial}\} \cup \{s_{intermediate}\} \cup \{s_{accept}\} \cup \{s_{reject}\}$ – the states in the model, union of initial, intermediate, accept and reject states
- $M = \{o_i\}$ – the observation space
- $A = \{a_{ij}\}$ – The state transition matrix, where: $$a_{ij} = Pr(S_{t+1} = s_j | S_t = s_i), 1 \le i,j \le N$$
- $B = \{b_i(o_k)\}$ – The observation probabilities $$b_i(o) = Pr(o | S_t = s_i), 1 \le i \le N$$
- $\pi = \{\pi_i\}$ – The initial state distribution $$\pi_i = Pr(S_1 = s_i), 1 \le i \le N$$

4 Behavior Recognition

The question is now, how to perform behavior recognition using above model? Since the state is not directly observable, we can only infer the probability being at some state being state s_i, $Pr(S_t = s_i)$, where $\sum_i Pr(S_t = s_i) = 1, \forall i$. This probability gives the likelihood that the correspond state to be the actual mental state of the robot.

As we have mentioned, a different BHMM is used for each type of behavior. It is possible that two BHMMs shows high accepting probabilities at the same time. This is a natural, however, since behaviors are not necessarily mutually exclusive. For example, in order to intercept a ball, a robot must also go to the ball. In this scenario, the Go-To-Ball BHMM and Intercept-Ball BHMM will both show a high accepting probability.

While the robot is executing, observation comes in at some interval, the question we want to ask is, given an observation sequence $O = o_1, o_2, \ldots, o_t$, what is the probability that we are in s_i? in other words, we are interested in the following probability:

$$Pr(S_t = s_i | O = o_1, o_2, \ldots, o_t, \lambda)$$

where λ is the Hidden Markov Model parameters.

we can decompose the above to:

$$Pr(S_t = s_i | O = o_1, o_2, \ldots, o_t, \lambda) = \frac{Pr(S_t = s_i \wedge O = o_1, o_2, \ldots, o_t, \lambda)}{Pr(O = o_1, o_2, \ldots, o_t, \lambda)}$$

let $\alpha_i(t) = Pr(S_t = s_i \wedge O = o_1, o_2, \ldots, o_t, \lambda)$, then we have:

$$Pr(S_t = s_i | O = o_1, o_2, \ldots, o_t, \lambda) = \frac{\alpha_i(t)}{\sum_i \alpha_i(t)}$$

$\alpha_i(t)$ is computed recursively by the following:

$$\alpha_i(t+1) = \sum_j \alpha_j(t) a_{ji} b_i(o_{t+1})$$

4.1 Behavior Segmentation and Restart

Our assumption requires each behavior to be a sequence of state traversals. We assume that each behavior starts from the initial state, and completes at the accept state. After executing one behavior, the robot will either idle or start executing another one. To complicate things further, the robot may terminate the execution of an behavior and start executing a new, possibly different, one.

The Behavior Hidden Markov Model described so far can recognize a single execution of a behavior, provided it is instantiated around the time when the real behavior starts executing. Otherwise, it will be "off phase" with the actually behavior and reliable recognition will no longer be guaranteed.

Akin to word segmentation in speech recognition, we need to perform behavior segmentation in order to apply HMM in the robot soccer domain. However, unlike non-continuous speech domain, the segmentation of robot behaviors is not as well defined. The execution of a sequence of robot behaviors is continuous and does not have easily perceivable "silent gaps" between behaviors. Therefore, it is very hard to look for segmentation points between behaviors.

We devised a scheme to go around the segmentation problem. We chose to instantiate a recognizer at regular intervals. By doing so, instead of spending effort searching for segmentation points between successive executions of two behaviors, and possible fail, we ignore such points. The hope is that one of the recognizers will instantiate at a point in time close to the behavior start time.

The granularity of the interval is important, too sparse an interval will result in the recognizer missing the beginning of the behavior.

With instantiation of BHMMs at regular intervals, the number of BHMM will go out of control as the recognition system may run continuously throughout soccer game. Although BHMM update is computationally cheap, A large amount of them will overload the processor. Therefore, a mechanism must be devised to remove BHMM from the system.

Two separate schemes are used to determine the removal of a BHMM. 1) a *timeout* is set for each type of behavior. It is assumed that a behavior such as Go-To-Ball will not take more than a certain amount of time to execute, say 30 seconds. Thus, 30 seconds after the instantiation, it will be discarded. 2) When a HMMs reach a high probability for a reject state. This tells us that very probably the robot is not doing what the BHMM is trying to recognize, therefore it is of no use continue updating this BHMM.

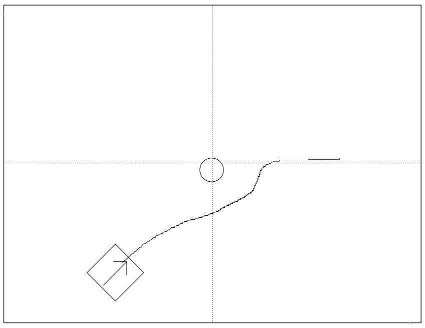

Figure 3: Robot trajectory

5 Algorithm Pseudo-code

- For all behavior recognizer type
  - ∗ Instantiate initial copy
  - ∗ Record start time and mark this instance as active
- Forever until done
  - ∗ Obtain object locations from vision system
  - ∗ For each active behavior recognizer
    - – Compute current observation from vision data
    - – Update current state probabilities using Most Like State update.
    - – Find most likely state (= `mls`)
    - – If (`mls` $\in$ {accept state})
      - – signal
      - – Continue to next recognizer instance
    - – If (`mls` $\in$ {reject state} AND $Pr($`mls`$)$ > reject threshold)
      - – signal and mark instance as inactive
      - – Continue to next recognizer instance
    - – Compute elapse time for this instance. If larger than timeout threshold mark instance as inactive
  - ∗ if ((current-time - last-instantiate-time) > instantiate threshold)
    - – For all behavior recognizer type
      - – Instantiate initial copy
      - – Record start time and mark this instance as active
    - – last-instantiate-time = current-time

6 Experimental results

The algorithm described about is implement. We implemented BHMM for the following behaviors: Go-To-Ball, Go-Behind-Ball, Intercept-Ball and Goalie-Align-To-Ball. An example robot trajectory of the Go-Behind-Ball behavior is shown in Figure 3. The robot is attempting to go around and behind the ball, but at the moment it went behind it, it aborts the behavior and heads away from the ball. Two Behavior HMMs are instantiated manually to recognize the behavior, Go-To-Ball and Go-Behind-Ball. In order to facilitate the illustration of a simple recognition, we do not use the restart mechanism.

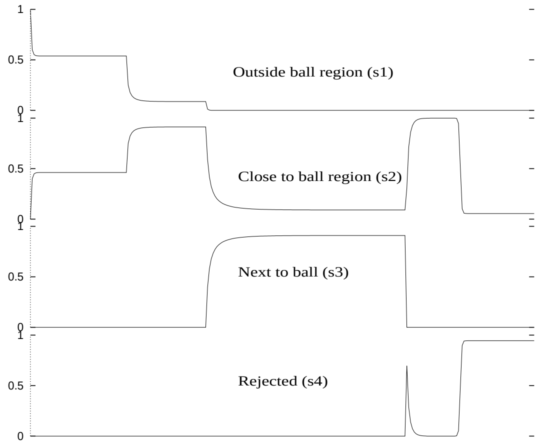

Figure 4: Go-To-Ball Behavior HMM probabilities

They are instantiate at the same time the robot starts executing the behavior. Result of the Go-To-Ball behavior HMM is in Figure 4. Result of a single run of the Go-Behind-Ball behavior HMM is in Figure 5.

The second set of results shows the running of the recognizers with restart. The robot trajectory used is shown in Figure 6. This trajectory is continuous execution of the same two behavior. The robot starts away from the ball, than it going behind the ball, leaves the ball vicinity, and then go behind the ball again. Figure 7 shows a trace of the output of all the Behaviors HMMs instantiated. In the diagram, each bar corresponds to the BHMMs' life span. The sections shows the change of the most likely state of the BHMM over time. White sections are intermediate/initial states, black sections are accepting states and a short section with a cross denotes rejected state. A rejected BHMM is automatically disposed of. The shift in position denotes the periodic instantiation of the BHMM for this behavior.

As can be seen in the diagram, the first set of BHMMs traversed through a set of intermediate states, as the robot arrives behind the ball, the state shows that it is accepted. Once the robot leaves it location, the behavior is no longer conform to the Go-Behind-Ball specification and is immediately rejected. At the same time, new BHMMs are created periodically, these captures the second execution of the Go-Behind-Ball behavior. Just like the first one, it successfully captures the completion of the behavior and then the rejection at the end.

7 Conclusion

Apart from recognition of behaviors, HMM are also useful in behavior prediction. Intermediate states probabilities is an indicator of a behavior in progress. They can be used in anticipating the future states of the robot. This can be very useful in a robotic soccer games.

In this work, we address the challenging problem of autonomous robot high-level behavior recognition from observed low-level state features. The paper contributes the formulation of robot behaviors as Hidden

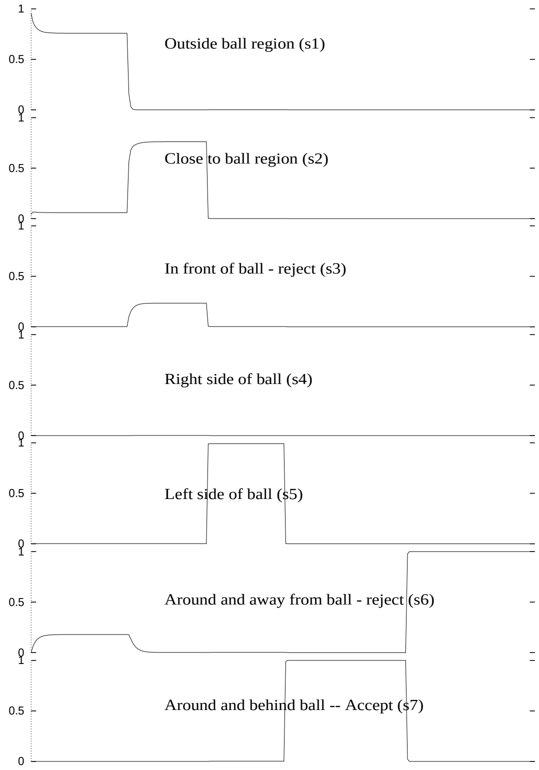

Figure 5: Go-Behind-Ball Behavior HMM probabilities

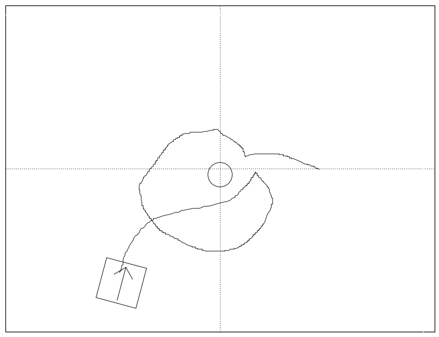

Figure 6: Robot Trajectory

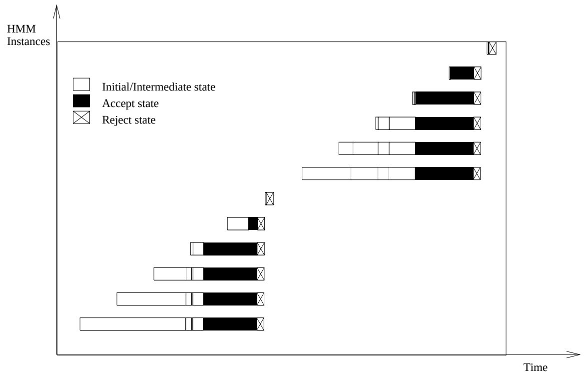

Figure 7: Go-Behind-Ball Behavior with Restart

Markov models. States in the HMMs correspond to an abstracted decomposition of the robot's behavior. Transitions between states are probabilistically achieved through observations of the world state. We introduce Behavior HMMs. Finally the paper presents the recognition algorithm which orchestrates multiple behaviors represented by different HMMs. The algorithm has been tested and used both in simulation and with real robots. Illustrative experiments are shown to demonstrate the real-time performance of our recognition algorithm.

References

[1] T. Balch and R.C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 1995.

[2] Kim Binsted. Character design for soccer commentary. In Minoru Asada and Hiroaki Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999.

[3] Mike Bowling, Peter Stone, and Manuela M. Veloso. Predictive memory for an inaccessible environment. In *Working Notes of the IROS-96 Workshop on RoboCup*, Osaka, Japan, November 1996.

[4] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2:14–23, 1986.

[5] Kwun Han and Manuela Veloso. Reactive visual control of multiple non-holonomic robotic agents. In *Proceedings of the International Conference on Robotics and Automation*, Leuven,Belgium, May 1998.

[6] Yang J and et al. Hidden markov model approach to skill learnign and its applications to telerobotics. *IEEE Transactions on Robotics and Automation*, 10(5):621–631, October 1994.

[7] Hiroaki Kitano, Milind Tambe, Peter Stone, Manuela Veloso, Silvia Coradeschi, Eiichi Osawa, Hitoshi Matsubara, Itsuki Noda, and Minoru Asada. The RoboCup synthetic agent challenge 97. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 24–29, San Francisco, CA, 1997. Morgan Kaufmann.

[8] Christopher Lee and Yangsheng Xu. Online, interactive learning of gestures for human/robot interfaces. In *IEEE International Conference on Robotics and Automation*, pages 2982–2987, 1996.

[9] Maja J. Mataric. Interaction and intelligent behavior. MIT EECS PhD Thesis AITR–1495, MIT AI Lab, August 1994.

[10] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–86, Feb. 1989.

[11] Milind Tambe. Agent architectures for flexible, practical teamwork. In *Proceedings of AAAI-97*, Menlo Park, California, 1997. AAAI Press.

[12] Manuela Veloso, Peter Stone, and Kwun Han. CMUnited-97: RoboCup-97 small-robot world champion team. *AI Magazine*, 19(3):61–69, 1998.