Anticipatory and Improvisational Robot via Recollection and Exploitation of Episodic Memories

Yoichiro Endo

Georgia Tech Mobile Robot Laboratory
College of Computing,
Georgia Institute of Technology
Atlanta, GA 30332, USA
endo@cc.gatech.edu

Abstract

Previously, we have introduced an anticipatory robot that could generate a cognitive map while simultaneously localizing itself relative to it. Inspired by recent hippocampal research, there, we demonstrated that the robot could exhibit anticipatory behavior if episodic memories that encode both spatial/nonspatial stimuli and behavioral actions are adequately utilized. In this paper, we will propose various ways to improve the previous method. For example, to address the computational complexity problem observed in the previous method, we will incorporate a new internal state variable inspired by the *somatic marker* hypothesis (a biological premise that speculates the role of emotional responses in our brain in terms of memory and decision making). Furthermore, we conjecture that this framework for anticipation can be even extended farther for a robot to deal with a novel situation (i.e., improvisation). We will discuss the concept of our improvisational robot in terms of anticipatory failures and their possible solutions.

Introduction

Recently in robotics, substantial efforts have been invested on critical applications such as military [1-3], nursing [4, 5], and search-and-rescue [6, 7]. These applications are critical in a sense that these robots may directly deal with human lives in life-or-death situations, and they are therefore required to rapidly make highly intelligent decisions. The intelligence we are looking for in this type of situations is the ability to *anticipate* and *improvise*. Anticipation here means that the robot can assess the current situation, predict the future consequence of the situation, and execute an action to have desired outcome based on the assessment and the prediction. On the other hand, improvisation is performed when the consequence of the situation is not fully known [8]. In other words, it is the ability to deal with a novel situation based on knowledge or skill being acquired before (i.e., Piaget's *intelligence* [9]).

How can we make a robot anticipate and/or improvise? Here, we seek clues from how our own brains work. Like a human infant, a brand-new robot, unwrapped from a shipping box, may not be ready yet to perform anticipation or improvisation. However, after having interactions with the real world for a certain period of time, we conjecture that the robot should eventually be able to figure out how to anticipate and/or improvise by reasoning the current situation based on relevant episodes that the robot has experienced in the past. Naturally, in order for the robot to recall relevant episodes, they have to be stored in some form of memory. In particular, we are interested in an episodic memory, a form of memory that contains information associated with a particular episode of experience, and it is stored in a way that the episode can be traced back and recalled in later time [10]. Given a sufficient framework to process a current episode of experience, store it as an episodic memory, and recall and utilize relevant past episodes for an ongoing situation, the primary hypothesis here is that extended exposure to the real world and interactions with it should help a robot improve its ability to anticipate (i.e., provide better assessments of the current situation, formulate better predictions of the future consequence of the situation, and execute better actions based on the assessment and prediction). Furthermore, our supposition here is that, even if the anticipation fails, the robot should still be able to take an appropriate action to reach its goal state because the episodic memory should be also utilized for improvisation.

In this paper, we will first highlight the computational steps involved in our anticipatory robot, which was originally introduced in [11]. We will discuss about the limitations of the previous approach and suggest possible solutions to them. We will then describe a newly proposed improvisational robot in terms of anticipatory failures. Finally, the conclusions and future work are discussed at the end.

Anticipatory Robot

As mentioned above, an anticipatory robot should assess the current situation, predict the future consequence of the situation, and execute an action to have desired outcome based on the assessment and the prediction. The concept of an anticipatory robot may be best represented by Rosen's diagram (Figure 1). Rosen [12] proposed the notion of anticipatory systems in order to analyze how adaptive living organisms work. The labels S, M, and E in the figure stand for *object system*, *model*, and *effector*, respectively. More specifically, S represents some dynamical system that interacts with the environment. For example, the system could be a microorganism, animal, or even an economy of some country. M is a model of S. Given a current state of S and an environment, M foretells what state S is likely to reach in the future. E is the effector that can interact with S or the environment in order to influence the future state of S. According to Rosen [12], the function of the anticipatory system is to: (a) Do nothing if M expects that S is likely to stay in a "desirable" state; or (b) activate E to correct the "trajectory" of S if M forecasts that an unwanted outcome is imminent. One of the important properties of the anticipatory system is that, unlike a reactive system that executes actions simply as a response to a current state (stimuli), the system reacts to a state that is expected to happen in the future.

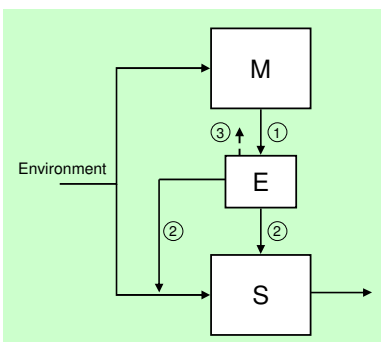

Figure 1: Anticipatory system. M = Model, E = Effector, and S = Object System. (Diagram reproduced from [12].)

In [11], we introduced an anticipatory robot that generates a cognitive map while simultaneously localizing itself relative to it. Here, the cognitive map is Rosen's M (model). More specifically, we regarded the cognitive map as a set of links (or relations) that connects discrete episodic memories; *episode* here means *a sequence of event representations, with each event characterized by a particular combination of spatial and nonspatial stimuli and behavioral actions*, according to the definition by Eichenbaum et al. [13] (neuroscientists) who investigated the formation of episodic memories within the hippocampus.

Expressing this formally in the context of robotics, the relationships between the episode (E) and a sequence of events (e) can be described by Equation 1:

$$E = (e_1, e_2, \ldots, e_n) \tag{1}$$

where n is the number of events in the episode. Furthermore, the event that encodes spatial/nonspatial stimuli and behavioral actions can be denoted as:

$$e_i = \{z_i, u_i\} \tag{2}$$

where z is the readings from all sensors onboard, u is a set of motor commands, and i is some instant. Treating e as a basic entity of a world representation (including both sensory and behavioral contents), in [11], we demonstrated that a robot could exhibit anticipatory behavior if this representation was adequately utilized. The subsequent subsections illustrate the four steps that involves in the computation of an anticipatory robot.

Step 1: Event Sampling

First, our assumption here is that any perception or interaction that a robot has with the environment is potentially useful for future anticipation. However, remembering all sensor readings and motor commands in its lifetime is not feasible as its size can easily exceed the capacity of the physical memory. In this work, we employed a simple reinforcement learning algorithm ($TD(\lambda)$ [14]) to accomplish temporal abstraction of incoming data.

At every time cycle, every current sensor reading is predicted by Equation 3:

$$r'_t = w_t \, \varepsilon_{t-1} \tag{3}$$

where r' is a single predicted sensor value (i.e., $r \in z$), w is a weight-vector, ε is a vector that includes readings from all onboard sensors and behavioral commands being executed at the instant (i.e., $\varepsilon = \{z, u\}$). At each measurement, the predicted value is compared against the actual value and the weight is updated via $TD(\lambda)$ update rule [14] (Equation 4):

$$\Delta w_t = \alpha (r_t - r'_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla r'_k \tag{4}$$

Here, α is a learning rate, λ^k is an exponential weighting factor, and the gradient $\nabla r'_k$ is a partial derivative of r'_k with respect to the weights; because of Equation 3, $\nabla r'_k$ is simply ε. Hence, Equation 4 can be rewritten as Equation 5:

$$\Delta w_t = \alpha (r_t - r'_t) \sum_{k=1}^{t} \lambda^{t-k} \varepsilon_{k-1} \tag{5}$$

In Figure 2, differences between predicted and actual sensor readings (after being root-mean-squared) are plotted against the time step when a simulated robot moved from one end of the hallway to the other (from left to right). As can be observed from the figure, the spikes of the errors appear to capture salient (or distinctive) features for the

robot, such as the presence of doors and the corridor junction. The peak of each spike is, here, considered as an occurrence of a new event (e), and the sensor readings and behavior commands of the instant are stored in the event.

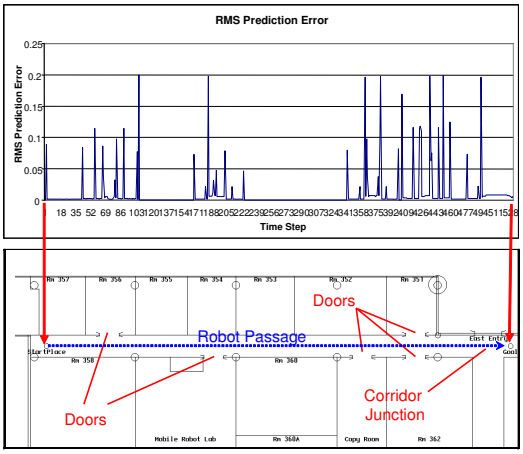

Figure 2: Error of the predicted sensor readings (simulation) and the environment.

Recall that an episode is a sequence of events (Equation 1). The question is what event starts a new sequence, and what event ends it (i.e., "when does an episode start and end?"). Our current proposition is to incorporate the robot's internal state in this context. However, in this (previous) study, the start and end points of an episode were arbitrary chosen by a human operator.

Step 2: Episode Recollection

In order to predict the future consequence of the current event, past episodes that are relevant to the current situation are gathered in this step. Let us use set C to denote the collection of all episodes that the robot has experienced before (Equation 6):

$$C = \{E_1, E_2, \ldots, E_N\} \tag{6}$$

N is the number of accumulated episodes. It should be noted that C was referred to as *cognitive map* in [11]. The collection of relevant episodes (M_{rel}) is a subset of C that is compiled by the following rule:

$$M_{\text{rel}} = \{\forall E \mid E \in C \wedge f_{\text{rel}}(E) = true\} \tag{7}$$

where f_{rel} is a filtering function that returns *true* if the input episode is relevant to the current situation. While our current proposal is to utilize the robot's internal state to determine what episode is relevant, in this study, f_{rel} was always set to return *true* (i.e., $M_{\text{rel}} = C$).

Step 3: Event Matching

In this step, exactly what event in the past episode corresponds to the current event is determined by a generic Bayes filter. In other words, the probability of the current event being same as a past event given a history of sensor readings and motor commands is computed. By applying the Bayes rule, the Markov assumption, and the law of the total probability, the posterior probability is solved by Equation 8 (see [11] for its derivation):

$$p(e_k|z^i,u^i) = \eta\, p(z_i|e_k) \int_{e_{k-1}\in E} p(e_k|u_i,e_{k-1})\, p(e_{k-1}|z^{i-1},u^{i-1})\, de_{k-1} \tag{8}$$

Here, $p(z_i,|e_k)$ and $p(e_k|u_i,e_{k-1})$ are the sensor and motion models, respectively. Note that the Bayes filter is recursively computed using previous posterior probabilities. Figure 3 shows two possible outcomes of how the posterior probability could be distributed over an episode. The first case is when the posterior probability is distributed around the average value and never exceeds a predefined threshold (Θ). In this case, we consider that none of the events in the episode matches with the current event. The second case is when the posterior probability does exceed the threshold. We refer to the event that generates the greatest distinct peak in the graph as a matching event (or localized event). In other words, the localized event is determined by the following equation:

$$e_{localized} = \begin{cases} \underset{e\in E}{\operatorname{argmax}}\; p(e|z^i,u^i) & \text{if } \max\limits_{e\in E} p(e|z^i,u^i) > \Theta \\ \varnothing & \text{otherwise} \end{cases} \tag{9}$$

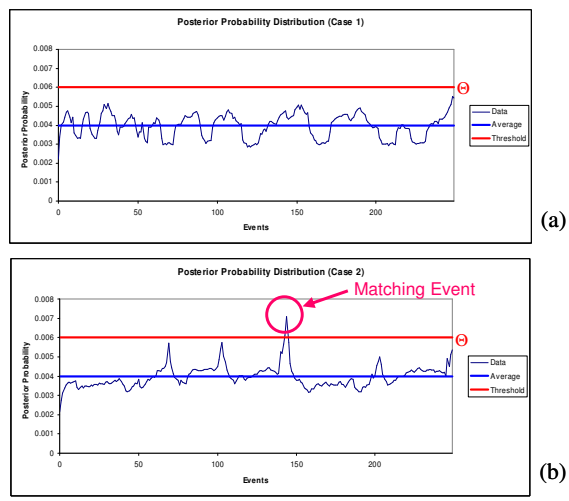

Figure 3: Posterior Probability Distribution over an Episode: (a) no matching event was found; (b) a matching event was found (e_{146}).

It should be noted that, at each computational cycle, the posterior probability is evaluated for all episodes collected in Step 2. In other words, if there are N episodes in M_{rel} (Equation 7), there would be at most N localized events at the end of each computational cycle.

Step 4: Behavior Selection

At this point, the robot knows what past event is relevant to the current situation. The next step is to decide what action to take in order to bring itself to a desired state (i.e., Rosen's E (Effector)). Imagine that, for example, there is a T-maze environment, and a red ball sits at its right arm of the maze. In [11], we proposed an anticipatory behavior that could guide the robot to the location of the ball by backtracking events in the episode. As shown in Figure 4, if

one of the events in the episode is known to have perceived the target object (the red ball), the robot attempts to follow a virtual path from the current event to the target object by executing the motor commands stored in the events between them. (Alternatively, the virtual path may be viewed as *intention* in the Belief-Desire-Intention (BDI) architecture [15].)

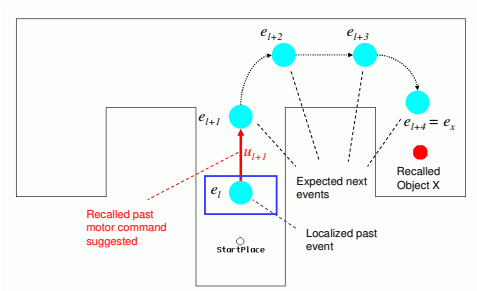

Figure 4: Following the virtual path from the current event to the target object.

Expressing this formally, let us denote the sensor readings and motor commands stored in event e as $z_{[e]}$ and $u_{[e]}$, respectively (i.e., $\{z_{[e]}, u_{[e]}\} \subset e$). Let us also define e_l and e_x as the event to which the robot localized to (found in Step 3) and the event at which the target object (*object X*) was perceived, respectively. The output of the anticipatory behavior can be then computed by Equation 10:

$$u_{out} = \begin{cases} u_{[e_{l+1}]} & \text{if } r_x \in z_{[e_x]} \wedge \{e_x, e_l\} \subset E \wedge x \geq l \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where r_x is the sensor reading that corresponds to detection of *object X*.

Preliminary Experiment and Results

In [11], the above anticipatory behavior was referred to as *Traject-Path-To X*, and its feasibility was empirically evaluated in simulation. Figure 5 shows the behavioral assemblage (*Search X*) that combines *Traject-Path-To X* with other essential behaviors for the maze navigation (*Explore* and *Avoid-Static-Obstacle*). Two conditions were prepared for this experiment. In the first condition, the robot was trained to visit both arms in the T-maze by following predefined waypoints (the target object was placed at only one of the arms), and subsequently *Search X* was executed to see if the robot would actually reach to the target object. The second condition was similar to the first one except that *Traject-Path-To X* was disabled within *Search X*.

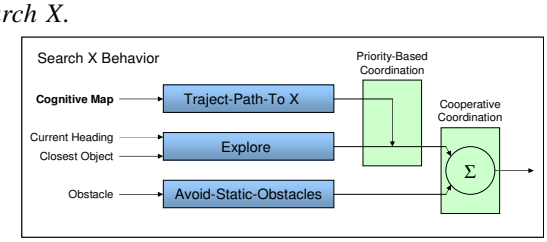

Figure 5: Behavioral Assemblage of *Search X*

Figure 6 shows the results of the experiment (after 64 tests). Having a statistically significant difference ($F_{1,126}$ = 18.986, p < 0.00003), this experiment proved that the anticipatory behavior was indeed effective. However, limitations of this approach were also identified. In the next section, we discuss those and possible solutions.

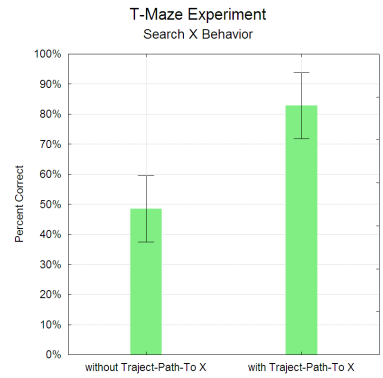

Figure 6: T-Maze Experiment Results from [11]

Improving Anticipation

The previous approach possessed some limitations. For example, it was observed that the computation time of each step increased drastically as the events were being accumulated. This predisposition is definitely undesirable for our anticipatory robot because it has to be able to make prompt decisions even after working in the environment for an extended period of time. We speculate that this problem is caused by our computational scheme in which full posteriors for the entire episodes in its memory are computed at every step (i.e., f_{rel} in Equation 7 always returned *true*.). Our proposition here is to improve f_{rel} by introducing new internal state variables: namely, desire (δ) and introspection (π). Furthermore, the question of how an episode should be segmented (i.e., "when does an episode start and end?") is also expected to be solved by these variables. The details of these internal state variables are explained below.

Desire. Recall that, in the previous approach, the robot could find its way to the goal autonomously, but the goal itself (*object X*) was set by an operator. Here, we propose a mechanism for the robot to automatically acquire its own goal at any given time based on its internal state: *desire*. The concept of desire is similar to a *goal state* in AI. It is the state that a robot desires to be at. However, as noted by Rao and Georgeff [15], while the goal state has to be believed by the robot that it is attainable, the desired state is not restricted by such a constraint. Furthermore, while a goal state in AI often is some symbolically describable world (e.g., *robot-has-bananas*), our desire is represented in the form of sensor readings (we also call it "sensation"). For example, the robot may have a gripper, and a tactile sensor on the gripper may be able to sense if a ball is grabbed or not. When the robot wants to grab a ball at some point, the exact perceptual state of the tactile sensor

for grabbing the ball ("ball grabbing" sensation) becomes a desire. A desire (denoted with symbol δ) can be formally described by the following set:

$$\delta = \{z, \alpha\} \tag{11}$$

where z is sensor readings, and α is a scalar indicating the magnitude of the desire. It should be noted that the α value can be negative (that is when δ is the state that the robot does not want to be at). Furthermore, depending on the circumstances, the robot may seek multiple desires at the same time. Let D be a set that contains all possible desires of the robot. Let us also suppose that there is a function (f_{des}) that frequently updates the contents of D based on the current sensor readings (Equation 12):

$$D_i = f_{\text{des}}(D_{i-1}, z_i) \tag{12}$$

For example, if a sensor indicates that the battery voltage is running low, f_{des} could increase the α value for the "battery charging" desire ($\delta_{battery\text{-}charging}$) in D. If the battery-charging is attained, the α value may be reset to zero. All desires that have non-zero α are considered to be active, and they are compiled as a new set (d) by the following rule (Equation 13):

$$d_i = \{\forall \delta_i \mid \delta_i \in D_i \wedge \alpha \in \delta_i \wedge \alpha \neq 0\} \tag{13}$$

Later in this paper, we show how d can be used to enhance f_{rel}, the filtering function for selecting relevant episodes.

Introspection. Above, we introduced the concept of *desire* to make the robot acquire its own goal. On the other hand, the concept of *introspection* is introduced here to provide the robot means to self-examine whether the current state is actually desirable or not. Such judgment should certainly be reflected by the status of the currently sought desires (d). However, other factors, such as survival of the robot or safety of humans, should be also taken account. For example, even if the desire is being satisfied (because the robot is getting closer to the goal), if the robot violently hits a human pedestrian, the robot should regard such a state as appalling.

In our proposed approach, the status of introspection is quantified by a single scalar variable (π), which can be either positive or negative. The value of π is adjusted by a function (f_{int}) that takes the current sensor readings and desire as well as the previous π value as its inputs (Equation 14):

$$\pi_i = f_{\text{int}}(\pi_{i-1}, z_i, d_i) \tag{14}$$

As mentioned above, there are two factors that could affect the adjustment of the π value: namely, desire and innate wiring (Equation 15):

$$\pi_i = \pi_{i-1} + \Delta\pi_{\text{desire}} + \Delta\pi_{\text{innate}} \tag{15}$$

The first factor ($\Delta\pi_{\text{desire}}$) relates to how much the current desires are being satisfied. The second factor ($\Delta\pi_{\text{innate}}$) relates innate wiring that specifies what types of perceptions are considered to be positive or negative for the robot. For example, a reading from a voltage meter can be considered as positive if it indicates that the battery is fully charged. On the other hand, if the tactile sensor reports that the robot is violently hitting some object, the perception may be considered as negative. We refer to this mapping as innate wiring since these rules are preprogrammed. Suppose we describe the innate wiring with a set (ϕ):

$$\phi = \{z, \alpha\} \tag{16}$$

As in δ (Equation 11), z is the sensor readings, and α is a scalar indicating the magnitude of the mapping (which can be positive or negative). Let Φ be the set that contains all ϕ being preprogrammed in the robot. Equation 14 can be then implemented by the following equation (Equation 17):

$$\pi_i = \pi_{i-1} + \sum_{\delta \in d_i} f_{\text{L}}(z_{[\delta]}, z_i)\, \alpha_{[\delta]} + \sum_{\phi \in \Phi} f_{\text{L}}(z_{[\phi]}, z_i)\, \alpha_{[\phi]} \tag{17}$$

where $z_{[\delta]}$, $\alpha_{[\delta]}$, $z_{[\phi]}$, and $\alpha_{[\phi]}$ are $z \in \delta$, $\alpha \in \delta$, $z \in \phi$ and $\alpha \in \phi$, respectively; and f_{L} is a function that returns the likelihood of a sample (specified in the first input parameter) given a measurement (specified in the second input parameter).

Recall that, in Equation 2, an event was a set of sensor readings and motor commands. In our newly proposed approach, the introspection value at the instant of sampling is also included in the event (Equation 18):

$$e_i = \{z_i, u_i, \pi_{\text{i}}\} \tag{18}$$

As discussed below, by being embedded in events (and hence in episodes), the introspection value should provide additional contextual information that would guide the robot to make better decisions (i.e., used as part of a heuristic function).

It is probably worth mentioning here that the notion of including the introspection value inside the event was inspired by Damasio's somatic marker hypothesis [16, 17]. In human brains, emotional responses are known to be generated by an element called amygdala. The emotional responses could be triggered by, for example, loud noise or fearful facial expression. Damasio [16] (neuroscientist) conjectures that some of the emotional responses are converted into new somatic signals (e.g., pain/pleasure) via the hypothalamus (hormonal) or brainstem/spine (neural). The somatic signals then arrive at the somatosensory cortex in the parietal lobe. Before entering the hippocampus, different sensory signals in the cortex are assembled at the transitional cortex, forming a uniform representation, which Damasio refers to as "dispositional representation" (which corresponds to our event e). Hence, the emotionally induced somatic signals (which corresponds to our π) from the somatosensory cortex are also embedded in the dispositional representation. The main point of Damasio's hypothesis is that the embedded somatic signals or *somatic markers* in the dispositional representation are very crucial upon when we make decision based on the past experience; the somatic markers prioritize the memory, so that irrelevant memories are filtered out upon recollection. To

test the hypothesis, Damasio [17] and recently Bar-On et al. [18] have conducted experiments using human subjects, and showed that patients with lesions in the neural circuit involving the somatic marker are prone to make poor judgments.

Improving f_{rel}. In this section, we show how the internal state variables (desire and introspection) can be used to improve f_{rel}, the filtering function that selects relevant episodes in Equation 7. First, we revise the representation of an episode (Equation 1) by incorporating a desire:

$$E_\delta = \{\delta, (e_1, e_2, ..., e_n)\} \tag{19}$$

This implies that episodes are indexed in the context of desires. In other words, a new episode starts recording events when a new desire becomes active (detected by monitoring d in Equation 13); the episode ends when the desire becomes inactive. In should be noted that, since a robot can seek multiple desires simultaneously, multiple episodes (sharing same events) may be compiled at the same time. For example, if a "ball grabbing" sensation and a "battery charging" sensation are simultaneously sought by a robot, two separate sets of episodes would be produced (e.g., $E_{ball\text{-}grabbing}$ and $E_{battery\text{-}charging}$).

Given the new representations of an event (Equation 18) and episode (Equation 19), improved f_{rel} can be described by the following equation (Equation 20):

$$f_{\text{rel}}(E) = \begin{cases} \textit{true} & \text{if } \max\limits_{\delta_i \in d_i} f_{\text{L}}(z_{[\delta_i]}, z_{[E]}, c) \geq \theta_\delta \wedge \Delta\pi_{[E]} \geq \theta_\pi \\ \textit{false} & \text{otherwise} \end{cases} \tag{20}$$

where f_{L} is the same likelihood function as in Equation 17 except that the parameters for the likelihood function (e.g., standard deviation) is explicitly specified (c); $z_{[\delta_i]}$ is the sensor readings of δ_i in the current desire set (Equation 13); $z_{[E]}$ is the sensor readings of δ that is a member of E_δ (Equation 19); θ_δ and θ_π are predefined thresholds; and $\Delta\pi_{[E]}$ is the difference of the π values between the first and last events in E (i.e., the progress of the introspection value through out the episode). In other words, if the desire associated with this past episode is not related to any of the currently sought desires, it is rejected; if there is no indication that this episode can contribute to boost the π value, it is also rejected by the filter.

Improvisational Robot

Improvisation is *to make, invent, or arrange offhand* [19]. As Calvin [20] points out, improvisation relates to what Piaget [9] called *intelligence*. Piaget suggested that intelligence is the ability to deal with a novel situation based on knowledge/skill being acquired before. Such intelligence would allow animals to deal with progressively more complex problems as base knowledge/skill accumulates. There are at least two noteworthy studies conducted on improvisation in the context of AI: one by Agre [8] and the other by Anderson [21] (both for their dissertations). Anderson considers improvisation as part of a control problem (referring to it as "intelligent control"). Conceptually, improvisation is similar to classical AI planning as both suggest a plan of action that should achieve a predefined goal [22]. However, it has a clear distinction. Agre [8], for example, points out that improvisation is performed when the consequences of the actions are not fully known. In other words, in classical AI planning, the information regarding the state space is given to the agent (its main concern is to find a correct sequence of actions). On the other hand, the idea behind improvisation in AI is that the agent would need to constantly consult with its memory in order to find a suitable action for the time being because computing every contingency in the world is not feasible (i.e., the agent has to ignore some part of the state space). Anderson [23] for instance utilizes domain knowledge in the form of constraints to help the agent narrow down its options or limit the search space. The constraints may also provide additional alternative options that the agent otherwise would have ignored. In other words, the constraints influence how the agent retrieves relevant actions from its memory. If the situation is within a familiar domain, some "routine response" can be applied quickly; Anderson [21] refers to this case as *weak* improvisation. On the other hand, if the situation is totally novel, it requires *strong* improvisation (i.e., "deeper reasoning" is necessary upon choosing the action). Anderson farther suggests that delaying execution of the action (or "deliberation") may help the agent explorer better alternatives, but it could also compromise the end result if any underlying assumption being made is time dependent.

In our view, one of the common denominators that the processes of anticipation and improvisation for a robot share is that they are both goal-oriented. In other words, whether the consequences of the actions are fully known or not, the mission of the robot is to reach its desired state. Hence, our supposition here is that the same infrastructure for anticipation that allows a robot to recollect and exploit relevant episodes to seek its desire can be straightforwardly extended for improvisation by adding a mechanism that could: 1) recognize failures of anticipation, and 2) recover from such failures. In the following sections, we explain the concept of our improvisational robot in terms of the anticipatory failures and their possible solutions.

Failure of Episode Recollection

Recall the second step in anticipation where relevant episodes (M_{rel}) for the current situation are collected using the filtering function (f_{rel}). While f_{rel} could help reducing exhaustive computation of the Bayes filter in the next step (Event Matching), we conjecture that overly restrictive f_{rel} may leave M_{rel} as an empty set (i.e., no past episode is considered to be relevant). As shown in Equation 20, there are three parameters (c, θ_δ, and θ_π) that affect the outcome of f_{rel}; we consider these parameters as constraints for improvisation. c and θ_δ determine how close the currently

active desire is to the one stored in the episode in question. More specifically, c is the input to the likelihood function (f_L) that determines the shape of the probability distribution, and θ_δ is a threshold that determines how much the output of f_L has to be in order for the two desires to be considered same. If, for example, we want our improvisational robot to indiscriminately accept more episodes, lowering the θ_δ value may suffice. On the other hand, if we want the robot to selectively relax the constraint for a particular sensor, the contents of c may be adjusted (e.g., to accept balls with any color instead of just red balls). Furthermore, progress of the introspection value during the episode ($\Delta\pi_{[E]}$) also influences the outcome of f_{rel}. A high θ_π value means that only highly rewarding episodes are opted. However, in some instances, the robot may not be able to afford itself to be too fastidious (e.g., time critical situations); hence, the decision may have to be made based on less attractive episodes by lowering θ_π.

Failure of Event Matching

Anticipatory failures could be also caused by failures in the event matching step. In other words, anticipation can fail if the robot fails to localize itself to any of the events in the past episode. Recall that the shape of the posterior probability distribution of e determines the outcome of the localization (Equation 9); if the posterior probability never exceeds the threshold (Θ), localization cannot be attained. Here, we hypothesize that no localization implies: (1) Θ was too restrictive, or (2) the Bayes filter actually identified that none of the events in the episode was relevant to the current event. It should be noted that distinguishing these two cases is not necessary straightforward. Nevertheless, the first case should be solved by simply lowering the Θ value. On the other hand, the second case perhaps requires deeper reasoning (strong improvisation). One possible solution for this is to introduce an intermediate desire (goal). For example, suppose that the robot is in a living room, and the battery-charging desire suddenly becomes active. However, it fails to localize to any of the past episodes for battery-charging ($E_{battery\text{-}charging}$) perhaps because the battery charger is located in a dining room. In this case, what we wish our improvisational robot to recognize is that, if it goes to the dining room, the battery can be charged. Notice, however, that our robot does not have the high-level concept of *dining room* because the information about the world is represented only in the form of sensor readings (z). In order to generate a plan of action in this framework, intermediate desires (d') are compiled by the following rule (Equation 21):

$$d'_i = \{\forall \delta_i \mid \delta_i \in D_i \wedge e \in E_g \wedge f_L(z_{[\delta_i]}, z_{[e]}) \geq \theta_{d'}\} \qquad (21)$$

where E_g is the episode that contains a real goal (e.g., $E_{battery\text{-}charging}$ in the example above); f_L is the same likelihood function mentioned in Equations 17 and 20; $z_{[\delta_i]}$ and $z_{[e]}$ are $z \in \delta$ and $z \in e$, respectively; and $\theta_{d'}$ is a predefined threshold. In other words, the intermediate desires are collected from D (all possible desires) if their sensor readings are same as the ones in any of the events that belong to E_g (according to the likelihood function). The intermediate desires (d') are then sought by the robot with the exactly same way it seeks the normal desires (d).

Conclusion and Future Work

In this paper, we proposed various ways to improve our previous model of an anticipatory robot. For example, in order to solve the computational complexity problem, incorporation of new internal state variables was suggested. In particular, the concept of *desire* was introduced to make the robot automatically acquire its own goal and partition the episodic memories based on the goals. On the other hand, by embedding in the representation of an event, *introspection* was introduced to serve as part of the heuristic function, deciding what past episodes are worth paying attention to for the current situation.

Furthermore, we conjectured that our anticipatory robot can be straightforwardly extended for an improvisational robot. By adjusting the constraints for the heuristic function, our supposition here is that the robot should be able to recover from anticipatory failures and improvise its actions based on the new constraints. Moreover, the concept of intermediate desires was also formulated in order to overcome the situations when strong improvisation is necessary.

In order to test the above hypothesis, empirical experiments are ought to be conducted next. In particular, we must evaluate: (1) if the internal state variables actually help reducing the overall computational time; (2) how adjustment of the constraints for the heuristic function affects the robot's improvisation; and (3) how effectively the intermediate desires help the robot to deal with a totally novel situation.

Acknowledgments

The author would like to thank Prof. Ronald Arkin for his helpful comments and advises on this study.

References

[1] L. Chaimowicz, A. Cowley, D. Gomez-Ibanez, B. Grocholsky, M. A. Hsieh, H. Hsu, J. F. Keller, V. Kumar, R. Swaminathan, and C. J. Taylor, "Deploying Air-Ground Multirobot Teams in Urban Environments," presented at *Multirobot Workshop*, Washington, D.C., 2005.

[2] Y. Endo, D. C. MacKenzie, and R. C. Arkin, "Usability Evaluation of High-Level User Assistance for Robot Mission Specification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 34, 2004, pp. 168-180.

[3] T. R. Collins, R. C. Arkin, M. J. Cramer, and Y. Endo, "Field Results for Tactical Mobile Robot Missions," presented at *Unmanned Systems*, Orlando, Fla., Assoc. for Unmanned Vehicle Systems International, 2000.

[4] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma, "Experiences with a Mobile Robotic Elderly Guide for the Elderly,"

presented at *AAAI National Conference on Artificial Intelligence*, 2002.

[5] K. Wada, T. Shibata, T. Saito, and K. Tanie, "Robot Assisted Activity for Elderly People and Nurses at a Day Service Center," *Proc. IEEE Int'l Conf. Robotics and Automation*, 2002, pp. 1416-1421.

[6] M. Burke J.L., R.R., Coovert, M.D., Riddle, D.L., "Moonlight in Miami: A Field Study of Human-Robot Interaction in the Context of an Urban Serach and Rescue Disaster Response Training Exercise," *Human-Computer Interaction*, vol. 19, 2004, pp. 85-116.

[7] J. Casper and R. R. Murphy, "Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at the World Trade Center," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 33, 2003, pp. 367-385.

[8] P. Agre, "The Dynamic Structure of Everyday Life," Massachusetts Inst. Technology, Technical Report AITR-1085, 1988.

[9] J. Piaget, *The Psychology of Intelligence*. Routledge & Paul, London, 1951.

[10] E. Tulving, "Episodic and Semantic Memory," in *Organization of Memory*, E. Tulving and W. Donaldson, eds., Academic Press, New York, 1972.

[11] Y. Endo and R. C. Arkin, "Anticipatory Robot Navigation by Simultaneously Localizing and Building a Cognitive Map," *Proc. Int'l Conf. Intelligent Robots and Systems*, 2003, pp. 460-466.

[12] R. Rosen, *Anticipatory Systems: Philosophical, Mathematical, and Methodological Foundations*, 1st ed. Pergamon Press, Oxford, 1985.

[13] H. Eichenbaum, P. Dudchenko, E. Wood, M. Shapiro, and H. Tanila, "The Hippocampus, Memory, Review and Place Cells: Is It Spatial Memory or a Memory Space?" *Neuron*, vol. 23, 1999, pp. 209-226.

[14] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 3, 1988, pp. 9-44.

[15] A. S. Rao and M. P. Georgeff, "Modeling Rational Agents within a BDI-Architecture," *Proc. Knowledge Representation and Reasoning*, San Mateo, Calif., Morgan Kaufmann, 1991, pp. 473-484.

[16] A. R. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*. G.P. Putnam, New York, 1994.

[17] A. R. Damasio, "The Somatic Marker Hypothesis and the Possible Functions of the Prefrontal Cortex," *Philosophical Trans. the Royal Society of London. Series B, Biological Sciences*, vol. 351, 1996, pp. 1413-1420.

[18] R. Bar-On, D. Tranel, N. L. Denburg, and A. Bechara, "Exploring the Neurological Substrate of Emotional and Social Intelligence," *Brain*, vol. 126, 2003, pp. 1790-1800.

[19] *Merriam-Webster's Collegiate Dictionary*, 10th ed. Merriam-Webster, Springfield, Mass., 1993.

[20] W. H. Calvin, "Evolving Improvisational Intelligence," 1995; http://faculty.washington.edu/wcalvin.

[21] J. E. Anderson, *Constraint-Directed Improvisation for Everyday Activities*, PhD Dissertation, Dept. Computer Science, Univ. of Manitoba, 1995

[22] P. Agre and D. Chapman, "What Are Plans for?" in *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, P. Maes, ed., MIT Press, 1991, pp. 17-34.

[23] J. E. Anderson and M. Evans, "Constraint-Directed Improvisation for Complex Domains," *Proc. the 11th Biennial Conf. the Canadian Soc. for Computational Studies of Intelligence on Advances in Artificial Intelligence*, Toronto, Springer-Verlag, 1995, pp. 1-13.