Extended Markov Tracking with Ensemble Actions

Zinovi Rabinovich and **Nicholas R. Jennings**
Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ
{zr,nrj}@ecs.soton.ac.uk

Abstract

In this paper we extend the control methodology based on Extended Markov Tracking (EMT) by providing the control algorithm with capabilities to calibrate and even partially reconstruct the environment's model. This enables us to resolve the problem of performance deterioration due to model incoherence, a negative problem in all model based control methods. The new algorithm, Ensemble Actions EMT (EA-EMT), utilises the initial environment model as a library of state transition functions and applies a variation of prediction with experts to assemble and calibrate a revised model. By so doing, this is the first control algorithm that enables on-line adaptation within the Dynamics Based Control (DBC) framework. In our experiments, we performed a range of tests with increasing model incoherence induced by three types of exogenous environment perturbations: catastrophic, periodic and deviating. The results show that EA-EMT resolved model incoherence and significantly outperformed the best currently available DBC solution by up to 95%.

1 Introduction

Model based control methodologies have found their expression in a wide range of AI techniques. From basic planning methods like STRIPs to complex PID controllers, the main principle remains the same: the decision on what action to take is based on a mathematical model of the environment response to an action application. However, in spite of being mathematically sound with provable properties, model based control methods suffer from one common pitfall. If the model is incoherent, that is a discrepancy exists between the actual reaction of the environment to an action application and the reaction described by the model, the decision made by a model based controller will be suboptimal. Now, a common approach to resolve this problem is model calibration: either through off-line or on-line interaction with the environment, the model is adjusted (or even entirely reconstructed) to reduce incoherence and facilitate better decision making.

Although model calibration has received increasing attention in recent years, the existing approaches make significant behavioural assumptions on the environment, ranging from frame assumptions [Pasula *et al.*, 2004] to structured motion of physical robots [Eliazar and Parr, 2004; Stronger and Stone, 2005]. Against this background, in this paper we relax these environment behavioural limitations by concentrating on a control framework with discrete abstract state models. More specifically, we concentrate on solving on-line model calibration for the Dynamics Based Control (DBC) framework [Rabinovich *et al.*, 2007], which allows application of the model based control methodology to an even wider range of domains (e.g. environments with abstract discrete state spaces and generic behaviour) than hitherto was possible.

In more detail, the DBC framework is almost unique in its ability to capture dynamic control tasks from the subjective point of view, i.e. in terms of agent's beliefs and observations of the changes that occur in the environment, and is based on two key principles. First, a part of the perceptual control paradigm [Powers, 1973], it states that changes in the environment are a means to altering and controlling perceptions. For instance, if we feel cold, we adjust temperature in a room to feel warmer, thus changing the environment to produce the required perception. Second, is that the dynamics of the system, rather than a momentary system state, are a means of describing the control task and modulation of environment dynamics are a means of solving the task. Notice, for example, that in our cold room example it was necessary to produce a change – increase the temperature – rather than bring it to a certain value. Combined together, these principles were implemented in a model based control algorithm termed *Extended Markov Tracking (EMT) control*. The algorithm has been shown to be an effective polynomial time solution to the DBC framework in discrete Markovian environments, where the next system state depends only on the current system state and the control action taken [Rabinovich *et al.*, 2007].

However, as with any model based control algorithm, EMT control is subject to deteriorating effects of model incoherence. In particular, our experiments further reveal that the standard EMT controller can not recover from persistent or catastrophic incoherence, where the environment behaves in a way not captured by the model. Nevertheless, EMT Control remains the sole solution within the DBC framework. Therefore, it is the only algorithm capable of operating in environments with a control task description that is both subjective and dynamic. The algorithm's polynomial running time underlines its importance even further, making its extension imperative. We thus modify the EMT control algorithm to include model calibration, which resolves the performance deterioration induced by a model incoherence.

The adaptive algorithm we have developed, the Ensemble Action EMT (EA-EMT), enables model calibration through the use of expert ensembles [Cesa-Bianchi and Lugosi, 2006]. For our purposes, each expert in the ensemble represents an alternative way to capture and model effects that an action has on the environment. EA-EMT dynamically merges the expert alternatives together, thus building a new environment model. Over time EA-EMT changes the properties of that merger, reflecting the performance of each expert in capturing environment behaviour, thus calibrating the environment model it uses.

The rest of the paper is organised as follows. In Section 2 we detail the operation of the standard EMT Control algorithm. Section 3 follows with the description of our new EA-EMT algorithm, detailing how it reconstructs and calibrates the environment model through the use of expert ensembles. Experimental support for the effectiveness of our approach is given in Section 4. The experiments take special focus on the on-line property of the EA-EMT model calibration, underlining the algorithm's ability to work in environments with changing behavioural trends. Section 5 summarises the results and gives future directions of this research.

2 EMT Control

The standard EMT algorithm continually maintains an estimate of system dynamics. To do so, the algorithm assumes that the system is an autonomous discrete Markov chain. That is, the system state stochastically develops over time without external influence, and the next system state depends on the current state only. This allows EMT to describe the estimate of the system dynamics by a single stochastic matrix. To maintain the estimate, the EMT algorithm performs a conservative update of the system dynamics matrix, minimising the Kullback-Leibler divergence between the new and the old estimate, with the limitation that the new estimate has to match the observed system transition that triggered the update.

To put it formally, assume that two probability distributions, p_t and p_{t+1}, are given that describe two consecutive states of knowledge about the system, and τ_t^{EMT} is the old estimate of the system dynamics. Then the EMT update τ_{t+1}^{EMT} is the solution of the following optimisation problem, where D_{KL} is the Kullback-Leibler divergence:

$$\begin{gathered}\tau_{t+1}^{EMT} = \arg\min_{\tau} D_{KL}(\tau \times p_t \| \tau_t^{EMT} \times p_t) \\ \text{s.t. } p_{t+1}(x') = \sum_{x} (\tau \times p_t)(x', x) \\ \text{and } p_t(x) = \sum_{x'} (\tau \times p_t)(x', x)\end{gathered}$$

The update is abbreviated: $\tau_{t+1}^{EMT} = H\left[p_t \to p_{t+1}, \tau_t^{EMT}\right]$.

Although EMT can work with more general environmental descriptions (see e.g. [Adam *et al.*, 2008]), it has been more commonly used with a discrete Markovian environment with partial observability, described by a tuple $MEnv =< S, s_0, A, T, O, \Omega >$, where:

- S is the set of all possible environment states;
- s_0 is the initial state of the environment (which can also be viewed as a distribution over S);
- A is the set of all actions applicable in the environment;
- T is the environment's probabilistic transition function: a mapping $T : S \times A \to \Delta(S)$. That is, $T(s'|a, s)$ is the probability that the environment will move from state s to state s' under action a;
- O is the set of all possible observations. This is what the sensor input would look like for an outside observer;
- Ω is the observation probability function: a mapping $\Omega : S \times A \times S \to \Delta(O)$. That is, $\Omega(o|s', a, s)$ is the probability that o will be observed given that the environment moved from state s to state s' under action a.

This naturally connects with the EMT algorithm, as knowledge about the system is summarised by a distribution vector over the system states $p_t \in \Delta(S)$, in which case the system dynamics estimator created by EMT has the form of a conditional probability $\tau : S \to \Delta(S)$.

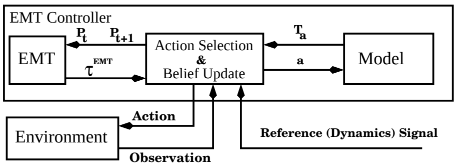


Figure 1: Closed loop of EMT Control

The overall control algorithm, termed *EMT Control*, forms a closed loop control with a reference signal [Stengel, 1994] (Figure 1 depicts the scheme). The reference signal encodes the task to be performed and takes the form of the conditional probability $\tau^* : S \to \Delta(S)$. The standard EMT Control (see Figure 2) can be summarised as a greedy one-step look ahead correction action selection. At every point in time, the algorithm attempts to predict the reaction of an estimation algorithm (EMT in this case) to the changes induced by an action (lines 2-7 of the algorithm), and then chooses the action that shifts the EMT estimator closest (line 8) to the reference dynamics τ^*. Once the action has been applied, the response of the EMT estimator to the changes in the environment is registered (line 11), and the control loops to make its next decision.

Notice that the controller action selection in lines 2-8 is heavily dependent on the environment model, as it uses the mapping T_a to predict action effects. If the model is incoherent the reaction of EMT can not be estimated correctly, which in turn will lead to selection of a suboptimal action. In what follows, we modify the action selection process to vary the environment model it uses.

3 Ensemble Action EMT

As already stated, the performance of the standard EMT Control algorithm deteriorates if the environment model is incoherent. However, by providing the algorithm with an additional method to correct model incoherences, it is possible to rectify the deterioration.

```
Require:
    Set the system state estimator $p_0(s) = s_0 \in \Delta(S)$
    Set the system dynamics estimator
        $\tau_0^{EMT}(\bar{s}|s) = prior(\bar{s}|s)$
    Set time to $t = 0$.
 1: loop
 2:   for all $a \in \mathcal{A}$ do
 3:     Set $\bar{T}_a = T_a$ {use transition model $T$ directly}
 4:     Set $\bar{p}^a_{t+1} = \bar{T}_a * p_t$
 5:     Set $D_a = H\left[p_t \rightarrow \bar{p}^a_{t+1}, \tau_t^{EMT}\right]$
 6:     Set $V(a) = \langle D_{KL}\left(D_a \| \tau^*\right)\rangle_{p_t}$
 7:   end for
 8:   Select $a^* = \arg\min_a V(a)$
 9:   Apply $a^*$, receive observation $o \in \mathcal{O}$
10:   Compute $p_{t+1}$ due to the Bayesian update:
          $p_{t+1}(s) \propto \Omega(o|s,a) \sum_{s'} \bar{T}(s|a,s') p_t(s')$
11:   Compute $\tau_{t+1}^{EMT} = H\left[p_t \rightarrow p_{t+1}, \tau_t^{EMT}\right]$
12:   Set $t := t+1$
13: end loop
```

Figure 2: EMT control algorithm.

Now, there are many incoherences a Markovian model, $MEnv =< S, s_0, A, T, O, \Omega >$, may have. While the choice of the state, action and observation spaces, as well as the observability function, may be dictated by subjective considerations (e.g. to make it more readable for the human domain designers), the transition function T is always dictated by the environment. Thus, in this work we choose to concentrate on the quality of the transition function T. This function maps actions into stochastic matrices, so that for each action $a \in A$ the matrix $T_a = T(\cdot|\cdot, a)$ models the effects of that action on the system state. The difference between the matrix T_a and the true effects of the action $a \in A$ is the incoherence type we have resolved in the EA-EMT algorithm (Figure 3). Thus, while the standard EMT Control views the transition mapping, $a \mapsto T_a$, to be constant, the EA-EMT algorithm modifies its transition mapping over time, reducing the mapping's incoherence. However, before we go into the details of how it was implemented, we would like to explain the principles of the approach taken by EA-EMT.

EA-EMT assumes that, although the mapping $T : A \rightarrow \Delta(S)^S$ is incoherent, the set of matrices $T_A = \{T_a = T(\cdot|\cdot, a)\}_{a \in A}$ represents feasible effects that the actions may have. The algorithm then attempts to assemble a better mapping, $\bar{T} : A \rightarrow \Delta(S)^S$, based on the set T_A. More specifically, for each action $a \in A$ the transition matrix $\bar{T}_a$ is a weighted linear combination of matrices in the set T_A, that is $\bar{T}_a = \sum_{b \in A} T_b * w_a(b)$. Intuitively, the weight $w_a(b)$ represents the similarity between the matrix $T_b \in T_A$ and the effects that the action $a \in A$ has on on the environment state. As the interaction between the EA-EMT algorithm and the environment progresses, the weights $w_a(\cdot)$ are updated, modifying the mapping $\bar{T} : A \rightarrow \Delta(S)^S$ to reduce its incoherence with the environment.

The update of the weights $w_a(\cdot)$ is based on the approach of predictions with expert ensembles [Cesa-Bianchi and Lugosi, 2006]. The intuition behind this approach is that, when making a prediction or a decision, a readily available set of feasible alternatives (the expert ensemble) can be merged together to form a prediction which is potentially better than any of the alternatives standing alone. In our case the *expert ensemble* is the set T_A, where each expert attempts to predict the effects an action would have on the environment state. From this point of view, the weight $w_a(b)$ expresses how much the expert $T_b \in T_A$ is trusted to capture the effects of the action $a \in A$ correctly. Once EA-EMT has applied an action, a^*, it measures the discrepancy between the effect a^* had and the effect predicted by expert T_b. The lower the discrepancy, the higher will be the weight $w_{a^*}(b)$ when the next control decision is made.

```
Require:
    Set the system state estimator $p_0(s) = s_0 \in \Delta(S)$
    Set the system dynamics estimator
        $\tau_0^{EMT}(\bar{s}|s) = prior(\bar{s}|s)$
    Set action weight vectors $w_a(a') \propto \delta_a(a') + \epsilon$
    Set time to $t = 0$.
 1: loop
 2:   for all $a \in \mathcal{A}$ do
 3:     Set $\bar{T}_a = \sum_{a'} T_{a'} * w_a(a')$
 4:     Set $\bar{p}^a_{t+1} = \bar{T}_a * p_t$
 5:     Set $D_a = H\left[p_t \rightarrow \bar{p}^a_{t+1}, \tau_t^{EMT}\right]$
 6:     Set $V(a) = \langle D_{KL}\left(D_a \| \tau^*\right)\rangle_{p_t}$
 7:   end for
 8:   Select $a^* = \arg\min_a V(a)$
 9:   Apply $a^*$, receive observation $o \in \mathcal{O}$
10:   Compute $p_{t+1}$ due to the Bayesian update:
          $p_{t+1}(s) \propto \Omega(o|s,a) \sum_{s'} \bar{T}_a(s|s') p_t(s')$
11:   Compute $\tau_{t+1}^{EMT} = H\left[p_t \rightarrow p_{t+1}, \tau_t^{EMT}\right]$
12:   for all $a \in \mathcal{A}$ do
13:     Set $\bar{p}^a_{t+1} = T_a * p_t$
14:     Set $D_a = H\left[p_t \rightarrow \bar{p}^a_{t+1}, \tau_t^{EMT}\right]$
15:     Set $V(a) = \left\langle D_{KL}\left(D_a \| \tau_{t+1}^{EMT}\right)\right\rangle_{p_t}$
16:     Set $w_{a^*}(a) \propto w_{a^*}(a)\beta^{V(a)}$
17:   end for
18:   Set $t := t+1$
19: end loop
```

Figure 3: The EA-EMT control algorithm.

Given the above principles, we have modified the standard controller algorithm. Specifically, line 3, previously directly substituted into the calculations the transition function from the provided model. Whereas now it uses a weighted combination of the matrices in T_A. The rest of the computations proceed as before until the EMT estimate, τ_{t+1}^{EMT}, of the action outcome is computed in line 11: the algorithm predicts the effects of each action on the EMT estimate, chooses the action that would bring τ_{t+1}^{EMT} closest to the reference signal τ^*, applies the action and receives an observation. At that point, the algorithm has to measure the performance of each expert, and update the weights. Now, recall that the algorithm operates in terms of subjective beliefs, the relevant effects of the action are thus those expressed in the EMT estimate τ_{t+1}^{EMT}. This means that the performance of each expert can be expressed by the distance between the estimate τ_{t+1}^{EMT} and the estimate that would have been obtained based on the expert prediction. This distance is computed in lines 13-15, and the weight of the expert is updated accordingly. Specifically, the old weight of the expert is multiplied by β^d, where $\beta \in (0, 1)$ is the parameter of the update and d is the distance above. Once all weights are updated, they are normalised to sum to 1, so that $\bar{T}_a$ at the next step will be a stochastic matrix. Notice that all these operations take time polynomial in the model parameters, such as the size of state, action and observation spaces.

4 Experimental Evaluation

To test the effectiveness of the EA-EMT algorithm, we have devised a set of comparative tests with the standard EMT Controller. To support comparability with previous work in this area, all tests were based on modifications of the Drunk Man (D-Man) domain: a controlled random walk over a linear graph (see Figure 4 for the principle structure) with actions weakly modulating the probability (only a small discrete set of probabilities in the range $(\epsilon, 1-\epsilon)$ with $\epsilon \gg 0$ is attainable) of the left and the right steps. A task within the domain is represented by a conditional probability $\tau^*(s'|s)$, the reference signal for the controller, specifying what sort of motion through the state space has to be induced. During an experiment run, the control algorithm was provided with a Markovian environment model, $MEnv =< S, s_0, A, T, O, \Omega >$, incoherent with the true behaviour of domain. The incoherences were created by introducing exogenous perturbations to the behaviour of the D-Man domain. In particular, three perturbations, making the model of the standard D-Man domain increasingly incoherent with the actual environment behaviour, were used:

- **Deviating**. An additional deterministic step (to the right) was done.
- **Periodic**. An additional deterministic step was done, but its direction changed with time.
- **Catastrophic**. A random permutation of actions was selected $\sigma : A \to A$. When the controller applied action $a \in A$, the environment responded instead to $\sigma(a)$.

Three baselines where obtained in various combinations: standard EMT Control algorithm operating in a perturbed environment, standard EMT Control operating within an unperturbed environment, and standard EMT Control operating in a perturbed environment with its model correctly encoding the environment perturbation. At least two baselines are present in each experimental setting to provide comparative performance bounds. Unless specified otherwise, the confidence envelope of 99.5% is depicted in all data graphs.

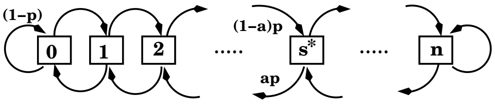


Figure 4: Principle structure of the Drunk Man domain.

In all our experiments the reference dynamics for the controller is given by $\tau^*(s'|s) \propto \delta_{s^*}(s') + \epsilon$, where $\epsilon > 0$ is small. In other words, the target prescribes that the environment should almost surely move to the ideal state s^* from any other state. In our experiments the state space was $S = \{0, ..., 12\}$, and the ideal state $s^* = 6$.

Notice that, due to the probabilistic nature of the domain, any reasonable[1] control scheme set to accomplish the task would result in a bell shaped empirical distribution of the system state. Success of the control scheme can then be readily appreciated visually by the difference of the expected value and the ideal system state, as well as the standard deviation of the empirical state distribution. To present an overall evaluation of a control scheme's performance, rather than a comparison of multiple parameters, we also measured the distance between the empirical distribution and δ_{s^*} using l_1 norm.

4.1 Deviating Perturbation

In this experiment we introduce a deviating perturbation. That is, beyond the usual probabilistic step, the environment has also deterministically shifted in one direction along the linear graph. For example (referring to Figure 4) if the system reached state $k \in \{0, ..., n-1\}$, the additional step will shift it to state $k + 1$.

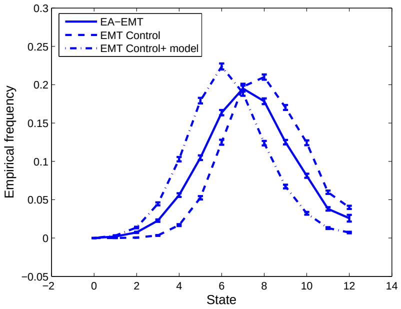
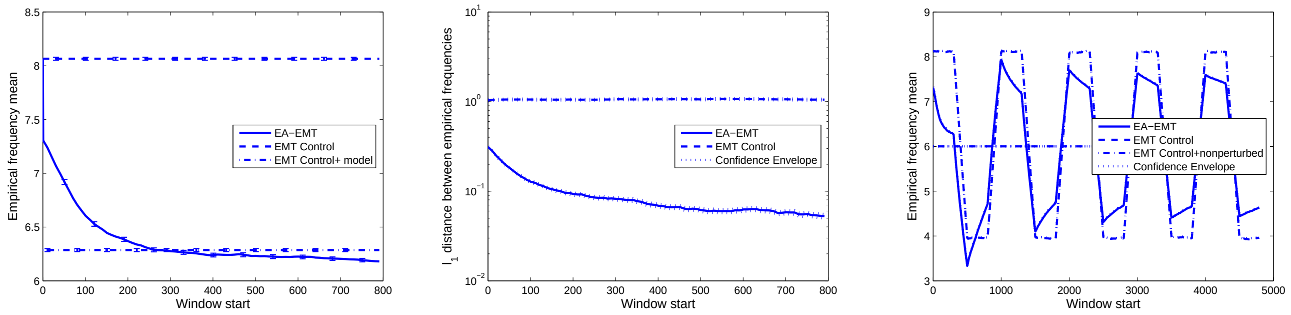


Figure 5: EA-EMT performance under persistent shift

In this context, Figure 5 shows the empirical distribution of system states under three control strategies: the EA-EMT controller and the standard EMT Controller equipped with the standard D-Man model (thus excluding the shift modelling), and the standard EMT Controller equipped with the environment model that explicitly captures the additional shift. The figure shows the complete empirical distribution of the EA-EMT obtained during the first 200 control choices made in this experiment, and marks a definitive improvement in performance. This can be seen from the fact that the standard EMT Control fails to enforce the reference dynamics τ^*, with the system spending the majority of its time away from the ideal state, $s^* = 6$, while EA-EMT manages to force the state distribution to concentrate closer to s^*. In fact, the distance between δ_{s^*} and the EA-EMT distribution induced in

[1] Unreasonable, for instance, would be choosing a constant action to equalise the left and the right step probabilities, as this would result in an almost uniform distribution, utterly defeating the controller purpose.



(a)

(b) Notice the log-scale of the Y axis.

(c)

Figure 6: EA-EMT adaptation to various perturbations: (a) Persistent Shift, (b) Random Permutation, (c) Switching Shift

the first 200 steps is 40% less than the comparable distance for the EMT controller. This, however, does not fully reflect the adaptability of EA-EMT. To this end, Figure 6(a) shows how the mean of the empirical distributions of the 200 step windows behave. The distributions induced by EMT Control do not change over time, resulting in straight horizontal lines depicting the constancy of the mean. On the other hand, the data shows that EA-EMT quickly adapts, the algorithm induces the empirical state distribution with the mean approaching the ideal state $s^* = 6$. In this respect, EA-EMT even slightly surpasses the performance of the standard EMT algorithm with the correct environment model. This is due to the adaptive portion of EA-EMT contributing to the tie breaking when considering similar actions – this tie breaking is rigid in EMT Control. Similar pictures occur with respect to the variance of the empirical distributions. This means that EA-EMT overcomes the model incoherence and increasingly concentrates the state empirical distribution around the ideal state, which is exactly what the reference dynamics, τ^*, requires.

4.2 Catastrophic Perturbation

The action space of the D-Man domain has a simple intuitive interpretation – the action sets how quickly the system state will shift left or right. The deviating perturbation did not exceed this interpretation, it simply meant that the system will naturally move in one direction faster than the other. In a way it also meant that the perturbation induced a very mild model incoherence – principally the model remained correct. However, EA-EMT can adapt to much more severe model incoherences. In fact, in the next set of experiments the environment model is completely incorrect. For each run in this experiment set a random permutation $\sigma : A \rightarrow A$ was selected. Then, when action $a \in A$ was applied, the environment reacted as if the action was $\sigma(a)$.

In more depth, Figure 7 shows the empirical distributions obtained in the first 200 steps of decision making. Permuting the action breaks any connection between what EMT Control expects the action to do and what actually occurs in the environment, essentially the actions are scrambled and the EMT Control chooses a random action. This results in the failure of the algorithm – the empirical state distribution is equivalent to that of applying no control at all[2]. In contrast, EA-EMT easily adapts to scrambling and performs increasingly well, as can be seen in Figure 6(b). Following the development of the empirical distribution within a sliding 200 step window, the figure shows the l_1 norm distance from the distribution formed by the standard EMT algorithm in the non-perturbed environment. This data demonstrates that EA-EMT exponentially quickly discovers the true effects of actions and approaches the performance of the EMT control in a non-perturbed environment. Even though the empirical distribution of the first 200 steps includes the first decisions made based on the scrambled model, it already recovers 70% of the performance lost due to the model incoherence and, through further adaptation, it reaches 95% recovery.

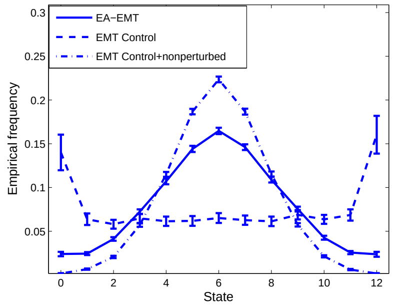


Figure 7: EA-EMT performance under random permutations.

4.3 Periodic Perturbation

Finally, it is important to evaluate whether the algorithm can also perform well in a dynamically changing environment. For example in robotics, even if everything else remains the same, the robot body will behave differently over time due to natural wear-and-tear. To test EA-EMT in such environments, we consider yet another perturbation: an additional deterministic step is made, and the direction of the step switches between left and right with constant period (500

[2] Since left and right steps fail in respective terminal states, the empirical probability there is higher.

control steps in our experiments). The shape of the distributions formed by the controllers are equivalent to those in the persistent shift experiment (see Figure 5), and we omit the respective graph. On the other hand, the development of the empirical distribution over time is quite different. In particular, Figure 6(c) shows the behaviour of the mean value for empirical distributions calculated within a 200 step sliding window. While the standard algorithm literally switches from one value to another, depending on the direction of the shift, the performance of EA-EMT always shows recovery after a direction switch occurs. Notice also, that the magnitude of the mean variation at the switch point becomes significantly (25%) less for EA-EMT than the standard EMT. This suggests that, beyond its ability to recover from irrelevant adaptations, the adaptive controller version learns to reduce the control inertia. In other words the algorithm reduces the impact of the sudden change in the environment behaviour, making the overall performance more stable.

5 Conclusions and Future Work

In this paper we present the Ensemble Action EMT algorithm – a polynomial time solution to the Dynamics Based Control framework with capabilities of on-line adaptation to environment model incoherences. The EA-EMT algorithm, uses the transition matrices contained within the model as an expert library for the feasible action effects in the environment, and treats any possible action as an ensemble predictor based on this experts set. Following the relevance of experts to the exhibited effects an action has on the EMT dynamics estimate, the weights within each ensemble action predictor are updated. We have experimentally verified the efficiency of the EA-EMT algorithm and shown that it quickly adapts to deviating, periodic and catastrophic exogenous perturbations of the environment. Furthermore, the data from the periodic perturbation experiment suggests reduced control inertia.

These adaptive capabilities of the EA-EMT algorithm allow a wider range of problems to be solved within the DBC framework than could hitherto be addressed. For example, previously a precise environment model was required to solve a type of search problem called area sweeping [Rabinovich *et al.*, 2007]. In contrast, the EA-EMT algorithm is capable of operating with environment behaviour described by a set of dynamic primitives that may occur in response to an action. This allows the algorithm to be applied in environments whose behaviour is only partially known, making precise modelling impossible.

Speaking more generally, the use of the expert ensemble method has a close association with plan recognition techniques [Bui, 2003; Riley and Veloso, 2002; Pynadath and Wellman, 2000], where a library (ensemble) of potential plans is commonly used. Plan recognition algorithms, through observation and causal interpretation of events, select a most likely explanation from the library. Similarly, EA-EMT views expert alternatives as explanations to changes in the environment state (which are not unlike the behaviours of plans in a library). Specifically, EA-EMT evaluates the performance of each expert based on the observed effects of an action within the environment. This parallel opens up the possibility of using EA-EMT as an opponent recognition and classification method in multi-agent adversarial scenarios.

However, EA-EMT also fuses and arbitrates between the various expert predictions. Specifically, the action model is essentially a weighted combination of the expert alternatives, which links it to behaviour-based robotics (BBR). In BBR [Arkin, 1998] a collection of simple (usually reactive) control algorithms is fused by an arbitration mechanism that combines their control signal. Given that this arbitration can include mutual inhibition or activation of other control signals, the resulting system can exhibit complex, adaptive behaviour (see e.g. [Mataric, 1998; Buffet *et al.*, 2002] and references therein) through the modulation and adaptation of the arbitration process itself. Ideologically similar adaptation occurs in EA-EMT, where individual experts gain higher weight in assembling the model with respect to their performance in predicting the effects of an action. Thus we plan to exploit this connection to construct new hybrid control schemas that combine the subjective dynamics task specification of EMT control with the structural task decomposition of BBR.

Finally, we also would like to investigate the possibility of altering the weight adaptation to include *forgetting*. That is, over time the weights should have an inherent tendency to equalise. By so doing, the controller could possibly produce higher frequency adaptability, enabling even better response to periodic perturbations.

References

[Adam *et al.*, 2008] A. Adam, Z. Rabinovich, and J. S. Rosenschein. Dynamics based control with psrs. In *7th AAMAS*, pages 387–394, 2008.

[Arkin, 1998] R. C. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.

[Buffet *et al.*, 2002] O. Buffet, A. Dutech, and F. Charpillet. Learning to weigh basic behaviors in scalable agents. In *1st AAMAS*, volume 3, pages 1264–1265, 2002.

[Bui, 2003] H. Bui. A general model for online probabilistic plan recognition. In *18th IJCAI*, pages 1309–1315, 2003.

[Cesa-Bianchi and Lugosi, 2006] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[Eliazar and Parr, 2004] A. I. Eliazar and R. Parr. Learning probabilistic motion models for mobile robots. In *21st ICML*, pages 32–??, 2004.

[Mataric, 1998] M. J. Mataric. Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in Cognitive Sciences*, 2(3):82–86, 1998.

[Pasula *et al.*, 2004] H. M. Pasula, L. S. Zettlemoyer, and L. P. Kaelbling. Learning probabilistic relational planning rules. In *14th ICAPS*, pages 73–82, 2004.

[Powers, 1973] William T. Powers. *Behavior: The control of perception*. Aldine de Gruyter, Chicago, 1973.

[Pynadath and Wellman, 2000] D. V. Pynadath and M. P. Wellman. Probabilistic state-dependent grammars for plan recognition. In *16th UAI*, pages 507–514, 2000.

[Rabinovich *et al.*, 2007] Z. Rabinovich, J. S. Rosenschein, and G. A. Kaminka. Dynamics based control with an application to area-sweeping problems. In *6th AAMAS*, pages 785–792, 2007.

[Riley and Veloso, 2002] P. Riley and M. Veloso. Recognizing probabilistic opponent movement models. In *The 5th RoboCup Competitions and Conferences*. 2002.

[Stengel, 1994] R. F. Stengel. *Optimal Control and Estimation*. Dover Publications, 1994.

[Stronger and Stone, 2005] D. Stronger and P. Stone. Simultaneious calibration of action and sensor models in a mobile robot. In *Proceedings of the ICRA*, 2005.