Coordination and Multi-Tasking Using EMT

Zinovi Rabinovich
Electronics and Computer Science
University of Southampton
zr@ecs.soton.ac.uk

Nir Pochter and **Jeffrey S. Rosenschein**
School of Engineering and Computer Science
The Hebrew University of Jerusalem
{nirp, jeff}@cs.huji.ac.il

Abstract

We introduce a multi-model variant of the EMT-based control algorithm. The new algorithm, MM-EMT, is capable of balancing several control tasks expressed using separate dynamic models with a common action space. Such multiple models are common in both single-agent environments, when the agent has multiple tasks to achieve, and in team activities, when agent actions affect both the local agent's task as well as the overall team's coordination.

To demonstrate the behaviour that MM-EMT engenders, several experimental setups were devised. Simulation results support the effectiveness of the approach, which in the multi-agent scenario is expressed in the MM-EMT algorithm's ability to balance local and team-coordinated motion requirements.

Introduction

Many real-world tasks can be characterised as team exercises that require formation support. Some examples include air-to-air refuelling between a tanker aircraft and the receiver of the fuel, multiple search helicopters scanning the sea for survivors, and multiple manipulators cooperatively transporting heavy construction equipment. The roots of the basic problem, however, can be analysed at a deeper level, and are quite general. These team tasks are characterised by two or more independent or loosely correlated mission statements (e.g., cover a certain area, but maintain relative distance from other aircraft); the mission statements are independent in their description, but for one parameter, namely, the actions and effects they allow an agent to perform.

This separation into sub-systems, however, need not be problematic. In fact, an effective control methodology has been proposed (Arkin 1998), and continually refined (Buffet, Dutech, and Charpillet 2002; Kaminka and Frenkel 2005; Kaminka et al. 2007), based on just such a separation—Behaviour-Based Robotics (BBR). As for balancing and unification, this can be readily accomplished using certain expert-based approaches (Freund et al. 1997; Vovk and Watkins 1998; Littlestone and Warmuth 1994), or resolved directly via learning (Gabor, Kalmar, and Szepesvari 1998). These control and decision unification techniques, however, have their limitations. For instance, the expert-based approach provides a fluent merging of basic advice from sub-systems, but concerns itself only with this merging and does not dictate how the basic advice from the sources is produced.

In this paper, we consider the situation where the sub-system models are not explicitly designed together with the overall system controller, and yet the control algorithm needs to compute and provide selection of an action for each. In other words, the algorithm has to be model-independent, universal, and programmable. Thus, we modify the EMT-based controller (see, e.g., (Rabinovich and Rosenschein 2006)) to give it the capability to resolve, balance, and unify decision-making in multi-model domains. We then design a discrete time and space environment where a stochastic version of the formation support task is formulated. This environment allowed us to simulate multi-model task requirements, as well as to create model incoherence vis-à-vis the environment, along with exogenous noise effects.

Our algorithm has demonstrated good and reliable performance both in terms of controllable balance among the tasks, and in terms of scalability with respect to the number of agents in the formation.

The rest of the paper is organised as follows. We first describe in brief the EMT algorithm, its control mechanism, and our extension to the case of multiple models. Experimental settings are then described (the simulated environment and sub-system models), followed by our experimental data from the application of the algorithm. We conclude with a discussion of our results and future work.

Multi-Model EMT-based Control

The Extended Markov Tracking (EMT) algorithm was designed to identify and estimate the transition matrix of a single Markov chain, based on two consecutive estimates of the chain's state. EMT has been incorporated into a series of perceptual (Powers 1973) control algorithms (Rabinovich and Rosenschein 2004; 2005; 2006; Rabinovich, Rosenschein, and Kaminka 2007), culminating in the creation of a novel control framework (Rabinovich, Rosenschein, and Kaminka 2007). Until now, however, EMT-based algorithms have been incapable of generating a control signal that could influence multiple independent environments. In this paper, we present a modification of the EMT-based control algorithm that enables this capability.

EMT Control of Markovian Environments

Following previous work on EMT-based control, we focus on discrete Markovian environments with partial observability, described by a tuple $< S, s_0, A, T, O, \Omega >$, 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 possible actions applicable in the environment;
- T is the environment's probabilistic transition function: a mapping $T : S \times A \to \Pi(S)$. 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 $\Omega : S \times A \times S \to \Pi(O)$. That is, $\Omega(o|s', a, s)$ is the probability that one will observe o given that the environment has moved from state s to state s' under action a.

This, however, only describes the environment in which the control agent operates, and must be extended by a description of the task to be performed. For an EMT-based controller, the task is described by a *reference system dynamics*[1] $\tau^* : S \to \Pi(S)$, a conditional distribution that describes an idealised trajectory of the system. In a sense, $\tau^*(s'|s) \in [0, 1]$ can be interpreted as a preference for the system to move to state $s' \in S$ if it currently resides in state $s \in S$. According to the perceptual control principle (Powers 1973), given the reference dynamics τ^*, the target of the controller is to enforce a perceptual equivalent of τ^* on the system by means of selecting an appropriate sequence of actions. In other words, given an algorithm capable of estimating the system's autonomous dynamics of the form $\tau : S \to \Pi(S)$, the controller needs to produce a sequence of actions so that τ will be as close as possible to the reference τ^*.

The estimation algorithm used by an EMT-based controller is *Extended Markov Tracking (EMT)*, which also gives rise to the controller's name. The estimator takes advantage of the fact that at any point in time knowledge about the system state can be summarised by a distribution vector over the states $p_t \in \Pi(S)$, and this knowledge summary can be updated using a simple Bayesian rule. In turn, the estimate, $\tau_t^{EMT} : S \to \Pi(S)$, of the system's autonomous dynamics that it produces, has to describe the change in that knowledge from p_{t-1} at time $t-1$ to p_t at time t. Thus, the dynamics estimate has to satisfy $p_t = \tau_t^{EMT} p_{t-1}$. Since there exists more than one dynamics satisfying the equation, EMT selects its estimate *conservatively*, and produces τ_t^{EMT} as close as possible to τ_{t-1}^{EMT} with respect to the Kullback-Leibler divergence. Formally, the estimate is computed by the following optimisation problem:

$$
\begin{aligned}
\tau_t^{EMT} &= H[p_{t-1} \to p_t, \tau_{t-1}^{EMT}] \\
&= \arg\min_{\tau} D_{KL}(\tau \times p_{t-1} \| \tau_{t-1}^{EMT} \times p_{t-1}) \\
&s.t. \\
&\quad p_t(x') = \sum_x (\tau \times p_{t-1})(x', x) \\
&\quad p_{t-1}(x) = \sum_{x'} (\tau \times p_{t-1})(x', x)
\end{aligned}
$$

Note the update abbreviation:
$\tau_t^{EMT} = H[p_{t-1} \to p_t, \tau_{t-1}^{EMT}]$.

The EMT-based controller selects actions in a greedy, on-line manner, utilising the EMT estimator in two ways. First, it uses it as a predictor of a perceptual effect an action may have. Second, in combination with a Bayesian update of the system state knowledge $p_t \in \Pi(S)$, EMT is used to track the perception of the exhibited system dynamics under the applied control, given that we would like to enforce the perception of a reference system dynamics $\tau^* : S \to \Pi(S)$. EMT-based control is summarised in Algorithm 1.

Algorithm 1 EMT-based control

1: $p_0(s) = s_0 \in \Pi(S)$
2: $\tau_0^{EMT}(\bar{s}|s) = prior(\bar{s}|s)$
3: $t = 0$.
4: **for all** action $a \in A$ **do** ▷ select which action to apply:
5: $\quad \bar{p}_{t+1}^a = T_a * p_t$ ▷ predict the future state distribution
6: $\quad D_a = H[p_t \to \bar{p}_{t+1}^a, \tau_t^{EMT}]$
7: Select $a^* = \arg\min_a \langle D_{KL}(D_a \| \tau^*) \rangle_{p_t}$
8: Apply a^* and receive an observation $o \in O$
9: $p_{t+1}(s) \propto \Omega(o|s, a) \sum_{s'} T(s|a, s') p_t(s')$ ▷ Bayesian update
10: Compute $\tau_{t+1}^{EMT} = H[p_t \to p_{t+1}, \tau_t^{EMT}]$
11: Set $t := t + 1$, goto 4

Before we proceed, one needs to notice two important features of the EMT-based controller. First, it is a universal and *programmable* on-line controller in the following sense. The EMT-based controller relies on a task model and reference dynamics to make on-line decisions. The algorithm itself remains unchanged, which means that the model and the reference dynamics essentially operate as a *program*. Although for some specific environments it may be possible to design a hand-written controller that will perform better than the EMT-based controller, any hand-written controller will quickly lose its edge in an open or time-variant system.

The second important feature of the EMT-based controller is its optimality criterion. Although the controller is only a greedy representative of the more general DBC framework (Rabinovich, Rosenschein, and Kaminka 2007), it retains the framework's point of view on optimality of performance. As a consequence, a direct performance comparison with other control methods, e.g., controllers produced by Reinforcement Learning (RL) (Sutton and Barto 1998), would be extremely artificial. EMT has a qualitatively different optimality criterion than that used in the RL literature.

[1] EMT's term for this is *ideal system dynamics* or *tactical target*.

The action selection mechanism is different in implementation and objective: EMT tries to select the action that will keep a certain dynamic given the world model, while in RL, the agent chooses actions that tend to increase the long-run sum of reinforcement signal values. Comparisons would be artificial, and any performance advantage shown by MM-EMT might be claimed to be the result of the specific reinforcement function that was chosen. Since no other DBC framework representatives are known at the moment, only a direct performance evaluation of an EMT-based control algorithm can be validly performed.

Multi-Model EMT-based Control

At times, there may be several behavioural preferences. For example, in the case of multi-robot movement in formation, two preferences on motion direction exist—one dictated by keeping in formation, the other by robot-specific capabilities and circumstances. Furthermore, these motion preferences are expressed by separate models, with only one thing in common, namely the action space, dictated by the robot's capabilities alone.

We are thus faced with a set of environment models $< S^k, s_0^k, A, T^k, O^k, \Omega^k >_{k=1}^M$ with common action space A, and a set of respective reference dynamics $\tau^{*,k} : S^k \rightarrow \Pi(S^k)$. The control decision is to select a *single* action that would satisfy some balance in achieving *all* of the reference dynamics in *all* of the environments.

To satisfy this balanced action selection, we modify the EMT-based control algorithm, replacing the action selection loop (line 4 of Algorithm 1), by the multi-model version presented in Algorithm 2. For ease of presentation we denote $V(a,k) = \left\langle D_{KL}\left(D_a^k \| \tau^{*,k}\right)\right\rangle_{p_t^k}$. Also, $Z^k = \sum_{a \in A} V(a,k)$ is a normalisation factor.

Algorithm 2 MM_EMT action selection

Require: A set of environment models, corresponding reference dynamics and their balancing weights $w(k)$

1: **for all** actions $a \in A$ **do**
2: **for all** model indexes $k = 1 \ldots M$ **do**
3: $\bar{p}_{t+1}^{k,a} = T_a^k * p_t^k$;
4: $D_a^k = H(\bar{p}_{t+1}^{k,a}, p_t^k, \tau_t^{EMT,k})$;
5: **for all** reference dynamics $\tau^{*,k}$ **do**
6: $V^k(a) = \frac{1}{Z^k} V(a,k)$
7: Select $a^* = \arg\min_a \sum_{k=1}^M w(k) V^k(a)$

The weight vector $\vec{w} = (w_1, \ldots, w_M)$ allows the additional "tuning of importance" among the environment models, without redesigning them or their reference dynamics.

There is, however, an additional modification that we deem necessary for the multi-model version of EMT. In the scenarios of interest the requirement for a multitude of models may occur due to a simplification or a structural separation attempt that simplifies the system reaction complexity by factoring it into several models. As a result, no model in the set will be exact, but only approximate. This may significantly influence the auxiliary Bayesian update used by the EMT algorithm and render its outcome inappropriate.

We thus replace the Bayesian update used in the standard EMT algorithm by a belief update with explicit noise as was proposed in (Even-Dar, Kakade, and Mansour 2007):

Algorithm 3 MM_EMT belief update

Require: p_t^k: the auxiliary state estimate of the k'th model at time t. $Uni(\cdot)$: a uniform distribution over S^k, the state space of the k'th model. ϵ_U: a small mixing factor.

1: $\bar{p}_{t+1}^k(s) \propto \Omega^k(o|s,a) \sum_{s'} T^k(s|a,s') p_t^k(s')$
2: $p_{t+1}^k = (1-\epsilon_U)\bar{p}_{t+1}^k(s) + \epsilon_U Uni(s)$

Formation Support Experiment

To test the performance of this version of the algorithm, we designed a discrete space version of a multi-robot formation support domain. A set of M agents is placed on parallel discrete tracks, starting from position zero and with zero motion velocity, as depicted in Figure 1. In this domain, the number of steps taken by each agent at any given time is modelled by a Poisson distribution, $Pois(\lambda)$, and the velocity is then interpreted as a setting of that distribution's parameter λ. In turn, they also follow some stochastic process of motion, parametrised by acceleration, which becomes a hyperparameter of the position change along the tracks. The agents' personal task is to modulate their acceleration so as to maintain some predetermined motion velocity. However, in addition to their personal task, the agents are also given a group activity. Specifically, the agents are arranged in a ring formation, and are tasked to maintain a constant relative distance to the agent next in the ring order.

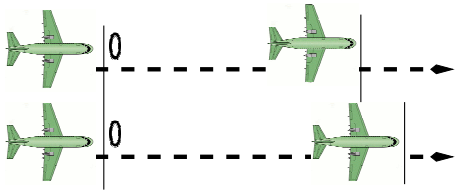

Figure 1: Discrete formation support

In our simulation, the range of possible velocities was $\lambda \in [0,3]$; however, we chose to discretise this range by a mapping $S^{vel} = [1 : vel_{max}] \leftrightarrow [0,3]$. This allowed us to model and implement the velocity development over time as a random discrete walk. The model naturally took the Markovian environment form $M^{vel} =< S^{vel}, s_0^{vel}, A, T^{vel}, O^{vel}, \Omega^{vel} >$, where A became a discrete set of possible accelerations, and the overall setting was reminiscent of the Drunk-Man model of (Rabinovich and Rosenschein 2004). The shift in agent's position along the tracks was simulated by sampling from a Poisson distribution parametrised by the velocity value mapped into the range $[0,3]$. As a result, the EMT-based control algorithm could be directly applied to the modulation of the local agent's speed. The relative distance control, however, needed an environment model different from the actual simulation process.

The relative distance of the agents, theoretically ranging from $-\infty$ to $+\infty$, was mapped using a Moebius transformation and discretisation onto a set $[1 : rpos_{max}]$. This was completed by the set of possible speeds, forming the discrete state space, $S^{rel} = [1 : rpos_{max}] \times [1 : rvel_{max}]$, of the Markovian model of the relative distance used by an EMT-based controller. The transition function, T^{rel}, of the model was formed under two assumptions. First, relative speed behaves as a random walk parametrised by the acceleration, and the relative position as a random walk parametrised by the relative speed at the previous time step; that is, $T^{rel}((r', v')|(r, v), a) = P_{DM}(r'|r, v) * P_{DM}(v'|v, a)$, where P_{DM} expresses a Drunk-Man type probability of transition. Secondly, the observation space of the relative model, $M^{rel} =< S^{rel}, s_0^{rel}, A, T^{rel}, O^{rel}, \Omega^{rel} >$, casts light only on the relative position, $O^{rel} = [1 : rel_{max}]$, with Ω^{rel} completely omitting the relative speed portion of the state, and adding blurring noise to the relative position.

Task Balancing Results

We ran a set of experiments with various settings of the relative model weights $w = (w^{vel}, w^{rel})$, with w^{vel} denoting the weight of the local velocity model, and w^{rel} denoting the weight of the relative distance and speed model.

The reference dynamics for the relative distance and speed were set to converge and maintain zero relative distance; the system is to be forced to remain in the subset of states $\{s = (r, v) \in S^{rel} | r = rpos_{max}/2\}$. Combined with the local speed model, this should have resulted in the agents synchronously increasing their speed to reach the optimum, and then modulating it slightly to maintain zero relative distance. An ideal relative distance, as a function of time, is zero. However, neither speed nor the relative position remain ideal. The speed varies stochastically under the applied acceleration, and since the position of the agents along the tracks is developing stochastically as well, different experiment runs can and will deviate temporarily from the ideal speed and relative distance.

Application of a control signal will lead to an empirical distribution of the speed and relative distance, with the parameters of that distribution expressing the effectiveness of the control signal applied. Figure 2 shows a set of such empirical distributions for the relative distance in a two-agent scenario. Each distribution corresponds to a specific relative weight setting between the tasks, and was obtained under multi-model EMT control.

The algorithm successfully combined the two tasks. Under control of the multi-model EMT algorithm, both agent-specific speed and the relative distance and speed were maintained, as can be seen from Figure 3. Although the potential contradiction between the two models has affected the means, the response to the varying weight of a model is best seen when the variance of the empirical distributions is inspected (Figure 4). Notably, asymmetry in the response to weights has also been observed in the multi-target version of the algorithm in (Rabinovich and Rosenschein 2006), and was also attributed to the dynamic properties of the reference dynamics used.

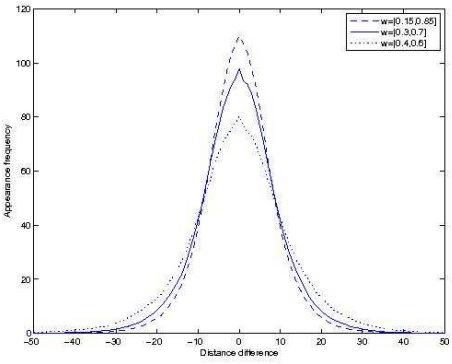

Figure 2: Empirical distribution of relative distance

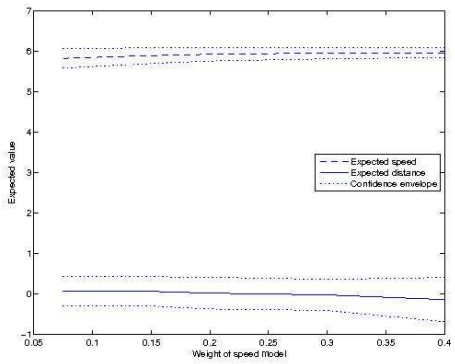

Figure 3: Expected value of the speed and relative distance as a function of the speed model weight

Scaling

The success of the two-agent version of our scenario, however, is hardly indicative of the real power of the algorithm in larger formations. We thus fixed the relative weight of the models, but increased the number of agents participating in the ring formation and once again measured the properties of the empirical distributions of the speed and relative position of the agents. The experiments illustrated the persistence of the expected value of the speed (Figure 5). What is more interesting is that long-range influences propagating through the ring had only sub-linear effect, as can be seen from the speed variance (Figure 6). We also performed two types of measurements on the relative position of the agents. First, we recorded a *local* measure—the empirical distribution of the maximum absolute difference in position of any two *consecutive* agents in the ring formation, denoted $z(t)$, where t is the time step in the system development. Second, we recorded a *global* measure—the maximum absolute difference between any two agents' positions, without regard to their order in the ring, denoted $n(t)$.

The *local* measure, although it deteriorated both in the expected value and variability, deteriorated at a sublinear rate with the increase in the number of agents (Figure 7). In other

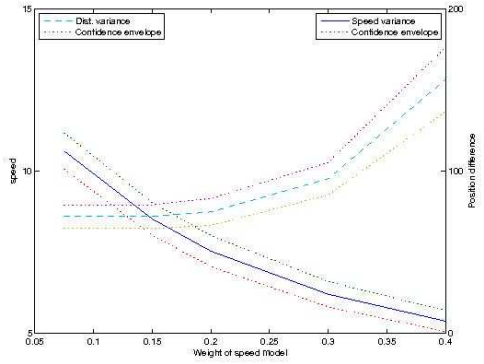

Figure 4: Variance of the speed and relative distance as a function of model weight

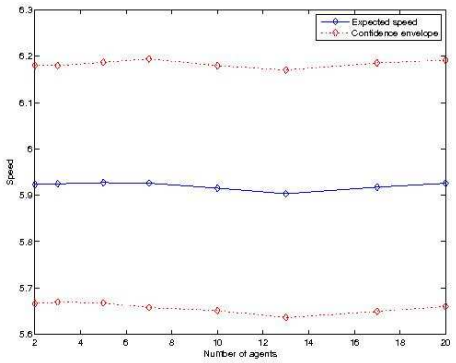

Figure 5: Speed expected value vs. formation size

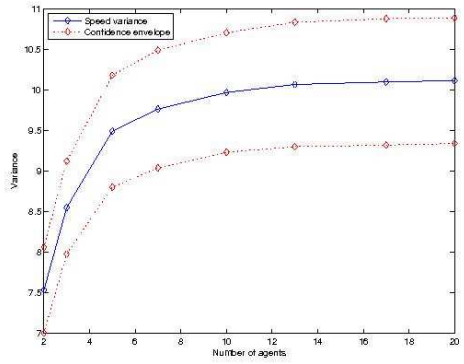

Figure 6: Speed variance vs. formation size

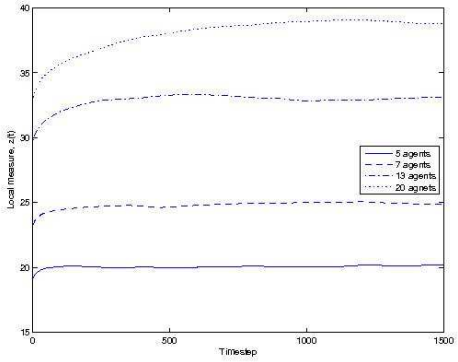

Figure 8: *Local* measure vs. time for various formations

words, although it became harder and harder to maintain relative position within the formation, the formation did not disintegrate, but rather actively counteracted the accumulating noise and disturbance within the team. This is further demonstrated by the stability of the expected value and variance parameters over time. For instance, Figure 8 shows the development of the expected value of the *local* measure for a variety of formation sizes over time. As if to underline the resistance to error accumulation by the formation as a group, the *global* measure repeated this pattern. Furthermore, the increase in the *global* measure parameters slows down as the formation becomes larger. To demonstrate this, we plotted the ratio between the expected value (variance) and the number of agents in the ring formation, as seen in Figure 9.

Discussion and Future Work

In this research, we modified the EMT-based control algorithm to support multi-model environments. Such environments are frequently observed in domains where a team of agents has to balance their action selection with respect to local and team goals. It is important to note that the resulting control algorithm is *programmable*: the EMT-based controller relies on a task model and reference dynamics to make on-line decisions. The algorithm remains unchanged, i.e., the model and reference dynamics essentially operate as a *program*. With respect to multi-tasking, this means that task model calibration, as well as the number of the tasks, can be done dynamically and also on-line. The algorithm should be able to accommodate any change in the number or nature of tasks in an on-line fashion, though experimental support of this assessment remains for future work.

We also constructed a discrete time and space domain, where agents are tasked with supporting motion in formation, under random variation of their speed and position in response to applied acceleration. We considered a simple ring formation that allowed each agent to maintain just two distinct models—the model of the local agent's speed, and the model of its relative position and speed with respect to the next agent in the ring structure. Multi-model EMT was successfully applied to the ring formation domain, and showed natural balancing capabilities between local and global tasks. Furthermore, the algorithm's performance survived the scaling of the domain to larger numbers of agents composing the ring. Obviously, our algorithm is not limited to this specific structure, and can accommodate general formations where the number of models can naturally vary and be heterogeneous across the formation.

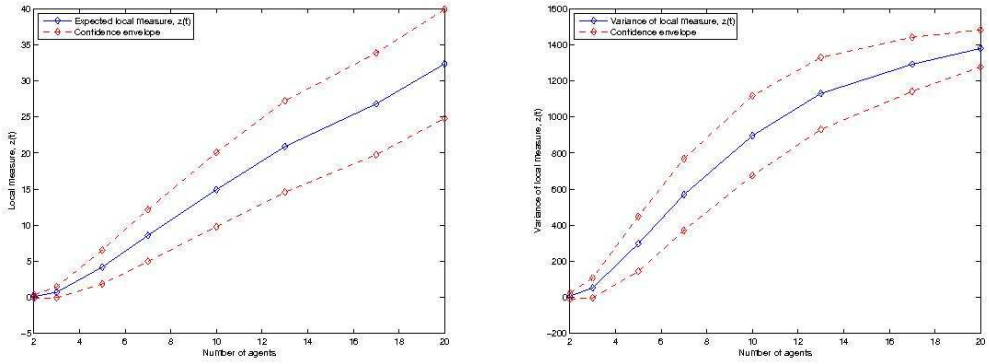

Figure 7: *Local* position measure, expected value vs. formation size and variance vs. formation size

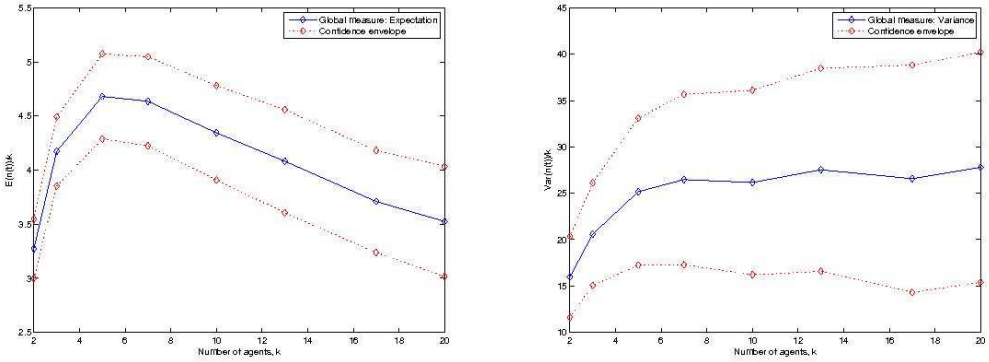

Figure 9: Expected value and variance of the *global* position measure in ratio to the number of agents

Acknowledgements

This work was partially supported by Israel Science Foundation grant #898/05, and by the EPSRC-funded project on Market-Based Control (GR/T10664/01).

References

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

Buffet, O.; Dutech, A.; and Charpillet, F. 2002. Learning to weigh basic behaviors in scalable agents. In *AAMAS'02*, volume 3, 1264–1265.

Even-Dar, E.; Kakade, S. M.; and Mansour, Y. 2007. The value of observation for monitoring dynamic systems. In *IJCAI 2007*, 2474–2479.

Freund, Y.; Schapire, R. E.; Singer, Y.; and Warmuth, M. K. 1997. Using and combining predictors that specialize. In *STOC'97*, 334–343.

Gabor, Z.; Kalmar, Z.; and Szepesvari, C. 1998. Multi-criteria reinforcement learning. In *ICML'98*, 197–205.

Kaminka, G. A., and Frenkel, I. 2005. Flexible teamwork in behavior-based robots. In *AAAI'05*, 108–113.

Kaminka, G. A.; Yakir, A.; Erusalimchik, D.; and Cohen-Nov, N. 2007. Towards collaborative task and team maintenance. In *AAMAS'07*, 464–471.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Information and Computation* 108:212–261.

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

Rabinovich, Z., and Rosenschein, J. S. 2004. Extended Markov Tracking with an application to control. In *The MOO Workshop, at the 3rd AAMAS Conference*, 95–100.

Rabinovich, Z., and Rosenschein, J. S. 2005. Multiagent coordination by Extended Markov Tracking. In *AAMAS'05*, 431–438.

Rabinovich, Z., and Rosenschein, J. S. 2006. On the response of EMT-based control to interacting targets and models. In *AAMAS'06*, 465–470.

Rabinovich, Z.; Rosenschein, J. S.; and Kaminka, G. A. 2007. Dynamics based control with an application to area-sweeping problems. In *AAMAS'07*, 785–792.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An introduction*. The MIT Press.

Vovk, V., and Watkins, C. 1998. Universal portfolio selection. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT'98)*, 12–23.