Social Entropy: a New Metric for Learning Multi-robot Teams *

Tucker Balch
Mobile Robotics Laboratory
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332
email tucker@cc.gatech.edu

Abstract

As robotics research expands into multiagent tasks and learning, investigators need new tools for evaluating the artificial robot societies they study. Is it enough, for example, just to say a team is "heterogeneous?" Perhaps heterogeneity is more properly viewed on a sliding scale. To address these issues this paper presents new metrics for learning robot teams. The metrics evaluate diversity in societies of mechanically similar but behaviorally heterogeneous agents. Behavior is an especially important dimension of diversity in learning teams since, as they learn, agents *choose* between hetero- or homogeneity based solely on their behavior. This paper introduces metrics of *behavioral difference* and *behavioral diversity*. Behavioral difference refers to disparity between two specific agents, while diversity is a measure of an entire society. Social Entropy, inspired by Shannon's Information Entropy [5], is proposed as a metric of behavioral diversity. It captures important components of diversity including the number and size of castes in a society. The new metrics are illustrated in the evaluation of an example learning robot soccer team.

1 Introduction

At present there are no metrics of diversity in robot teams; societies are simply classified as "heterogeneous" if one or more agents are different from the others and "homogeneous" otherwise. This either/or labeling doesn't tell us much about the *extent* of diversity in heterogeneous teams. How can we determine, for instance, if one system is more or less diverse than another? The answer is crucial for investigations regarding the origins and benefits of heterogeneity.

As an example of the kind of issue this work addresses, consider two teams of robots: $\mathcal{R}_a$, and $\mathcal{R}_b$. Suppose $\mathcal{R}_a$ is composed of 99 identical robots and one unique robot; while $\mathcal{R}_b$ is composed of two groups of 50 identical robots each. Both teams have the same number of robots (100) and the same number of robot types (2), but intuitively it seems $\mathcal{R}_b$ is "more diverse" than $\mathcal{R}_a$. How can the difference be quantified? This paper suggests *Social Entropy*, inspired by Shannon's Information Entropy [5], as an appropriate measure of diversity in robot teams.

Investigation of diversity at the societal level forces several related issues to the surface. First, since diversity is based on differences between individuals in a group, a measure of robot difference is necessary. One can see how mechanical differences are quantifiable, but what about physically identical agents which differ only in their behavior? One approach is to look for the differences in the agents' behavioral coding. In robots using identical reinforcement learning strategies for instance, the robots' policies can be compared (specific examples are given later). Second, assuming differences between individuals can be measured, how should they be used to evaluate diversity in the *society*? In the approach advocated here, the society is partitioned into castes of behaviorally equivalent agents based on the difference metric. Diversity is evaluated based on the number of castes and the number of robots in each caste.

To provide an example for concrete discussion, robot soccer, a multi-robot learning task is presented. After that, terminology regarding robot behavior and difference are provided for the formal discussion that follows. Later sections introduce definitions and mathematical formulations of behavioral difference, societal hetero- and homogeneity, and Social Entropy for robot teams. Finally, the metrics are applied to example soccer robot teams.

*Proc. 10th International FLAIRS Conference (FLAIRS-97)

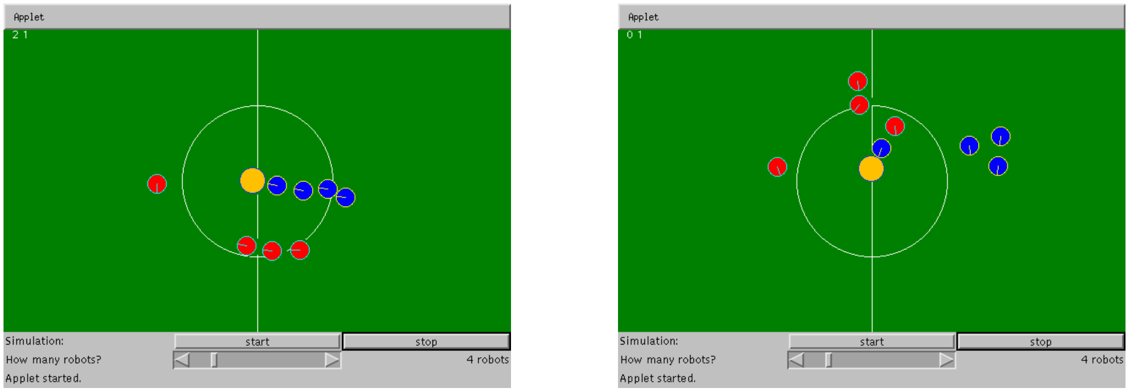

Figure 1: Examples of hetero- and homogeneous learning soccer teams. In both cases the learning team (dark) defends the goal on the right. The agents try to propel the ball across the opponent's goal by bumping it. A homogeneous team (left image) has converged to four identical behaviors which in this case cause them all to group together as they move towards the ball. A heterogeneous team (right) has settled on diverse policies which spread them apart into the forward middle and back of the field.

2 Robot Soccer

Robot soccer is an increasingly popular focus of robotics research [4]. It it is an attractive domain for multiagent investigations because a robot team's success against a strong opponent often requires some form of cooperation. Additionally, many people are familiar with the human version of soccer and can easily identify with and understand the problem. For this research, the game is simplified in a few respects:

- Teams are composed of four players.
- The sidelines are walls: the ball bounces back instead of going out-of-bounds.
- The goal spans the width of the field's boundary. This helps prevent situations where the ball might get stuck in a corner.
- The ball is propelled only by robot bumps.
- Play is continuous: After a scoring event, the ball is immediately replaced to the center of the field.

The examples discussed here are drawn from robot soccer simulations. The simulation proceeds in discrete steps. In each step the robots process their sensor data, then issue appropriate actuator commands. Ball position and defended goal sensors are used in the experiments examined here. Space precludes a more detailed description of the system.

The skills provided to the soccer agents are designed as motor schema-based behavioral assemblages. Motor schemas are the reactive component of Arkin's Autonomous Robot Architecture (AuRA) [1]. AuRA's design integrates deliberative planning at a top level with behavior-based motor control at the bottom. The lower levels, concerned with executing the reactive behaviors are incorporated in this research. Motor schemas may be grouped to form more complex, emergent behaviors. Groups of behaviors are referred to as *behavioral assemblages*. One way behavioral assemblages may be used in solving complex tasks is to develop an assemblage for each sub-task and to execute the assemblages in an appropriate sequence.

To implement an overall soccer strategy, each robot is provided a set of behavioral assemblages for soccer. Each assemblage can be viewed as a distinct "skill" which, when sequenced with other assemblages forms a complete strategy. The behavioral assemblages developed for these experiments are:

- *move_to_ball (mtb)*: The robot moves directly to the ball. A collision with the ball will propel it away from the robot.
- *get_behind_ball (gbb)*: The robot moves to a position between the ball and the defended goal while dodging the ball.
- *move_to_back_field (mtbf)*: The robot moves to the back third of the field while being simultaneously attracted to the ball.

The overall system is completed by sequencing the assemblages with a selector which activates an appropriate skill depending on the robot's situation. This is accomplished by combining a boolean perceptual feature, *behind_ball* (*bb*) with a selection operator. The selector picks one of the three assemblages for activation, depending on the current value of *bb*. Programming the selector is equivalent to specifying the agent's policy (e.g. Figure 2).

In experimental evaluations, learning robots are trained against a control team, using rewards based on the game's score. After the agents converge to stable behaviors, policies of the individual robots are examined for diversity in the resulting team. Since

perceptual feature	assemblage *mtb*	*gbb*	*mtb f*
not behind_ball	0	1	0
behind_ball	1	0	0

Control Team Forward

perceptual feature	assemblage *mtb*	*gbb*	*mtb f*
not behind_ball	0	1	0
behind_ball	0	0	1

Control Team Goalie

Figure 2: The control team's policy viewed as look-up tables. The 1 in each row indicates the behavioral assemblage selected by the robot for the perceived situation indicated on the left. The abbreviations for the assemblages are introduced in the text.

	mtb	*gbb*	*mtbf*	*mtb*	*gbb*	*mtbf*	*mtb*	*gbb*	*mtbf*
not behind_ball	0	0	1	0	0	1	0	0	1
behind_ball	0	0	1	0	1	0	1	0	0
not behind_ball	**0**	**1**	**0**	0	1	0	**0**	**1**	**0**
behind_ball	**0**	**0**	**1**	0	1	0	**1**	**0**	**0**
not behind_ball	1	0	0	1	0	0	1	0	0
behind_ball	0	0	1	0	1	0	1	0	0

Figure 3: The nine soccer robot policies possible for the learning agents discussed in the text. Each policy is composed of one row for each of the two possible perceptual states (not behind ball or behind ball - *bb*). The position of the 1 in a row indicates which assemblage is activated for that policy in that situation. The policies of the goalie and forward robots introduced earlier (Figure 2) are in bold.

this article focuses on evaluation metrics for robot teams, the learning system is presented in overview only. For more detail, the reader is referred to [3].

The control team includes three agents that move to the ball when behind it and another that remains in the backfield. For convenience, we refer to them as "forward" and "goalie" policies. The forwards and goalie are distinguished by the assemblage they activate when they find themselves behind the ball: the forwards move to the ball (*mtb*) while the goalie remains in the backfield (*mtbf*). Both types of player will try to get behind the ball (*gbb*) when they find themselves in front of the ball.

The learning teams are developed using the same behavioral assemblages and perceptual features as the control team. Clay (the system used for configuring the robots) includes both fixed (non-learning) and learning coordination operators [2]. The control team's configuration uses a fixed selector for coordination. Learning is introduced by replacing the fixed mechanism with a learning selector. A Q-learning [6] module is embedded in the learning selector. At each step, the learning module is provided the current reward and perceptual state, it returns an integer indicating which assemblage the selector should activate. The Q-learner automatically tracks previous perceptions and rewards to refine its policy. Altogether there are 9 possible policies for the learning agents since for each of the two perceptual states, they may select one of three assemblages. Figure 3 summarizes the possible policies. Based on these nine policies there are a total of 6561 possible 4 robot teams. Two example teams, one homogeneous, the other heterogeneous are illustrated in Figure 1.

3 Terminology

To facilitate the discussion, the following symbols and terms are defined:

- **Robots:**
  - R_j is an individual robot.
  - $\mathcal{R}$ is a society of N robots with $\mathcal{R} = \{R_1, R_2, R_3 ... R_N\}$
- **Castes:**
  - $\mathcal{C}$ is a classification of $\mathcal{R}$ into c possibly overlapping sub-sets. Example: a four robot team $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ is divided into three sub-sets, $\mathcal{C} = \{\{R_1, R_2\}, \{R_3\}, \{R_4\}\}$.
  - C_i is an individual sub-set of $\mathcal{C}$. Example: if $\mathcal{C} = \{\{R_1, R_2\}, \{R_3\}, \{R_4\}\}$, then $C_1 = \{R_1, R_2\}$.
  - The sub-sets of $\mathcal{C}$ are **castes**.
- **Sensing and Action**
  - $\bar{i_j}$ is the sensor input provided to R_j's control system. $\bar{i_j}$ is referred to as the **perceptual state** of the robot.
  - P_i^j is the proportion of time R_j experiences input $\bar{i}$.
  - $\bar{a_j}$ is the vector of outputs generated by R_j's control system based on the input $\bar{i_j}$.

Now consider how the behavioral difference between two robots might be evaluated. Imagine the ultimate robotics laboratory where we could enclose an agent in an evaluation chamber and expose it to all possible sensory situations while carefully recording its response to each. This corresponds to varying the sensory input $\bar{i}$ while tracking the actuator output $\bar{a}$. Having collected the data for one robot, the next agent could be placed in the chamber and exposed to the same situations. Finally, the traces of the robots' responses over all sensory inputs would be compared to draw up a measure of their behavioral difference. Since this evaluation chamber will likely never exist, other kinds of behavioral comparison must be considered.

When otherwise identical robots diversify by learning, their behavioral difference can be evaluated by comparing their policies. For example, the goalie and forward soccer robots introduced earlier exhibit behavioral differences that are reflected in and caused by their differing policies (Figure 2). In comparing robot policies, $\bar{i}$ represents the perceptual features an agent uses to selectively activate behavioral assemblages. In the case of the soccer robots $\bar{i} = 1$ if the robot is behind the ball and 0 otherwise. $\bar{a}$ is the selected behavioral assemblage. For the soccer robots $\bar{a}$ can be viewed as unit vector with one non-zero element indicating which behavioral assemblage is activated. For instance, $\bar{a} = (0, 0, 1)$, would indicate that the third assemblage is active.

Even though this paper focuses on comparing behavior at the level of an agent's behavioral sequencing strategy, similar evaluations could be conducted at other levels. The agent could be measured at a lower level, for instance, with $\bar{i}$ representing the full set of real-valued sensor inputs, and $\bar{a}$ being the full set of actuator outputs (e.g. motor currents, etc.).

Returning to the soccer example, recall that at each movement step, one of three behavioral assemblages (*move_to_ball, get_behind_ball* or *move_to_backfield*) is selected, based on the *behind_ball* perceptual feature. If, in every perceptual state, two robots select the same output, they are considered **behaviorally equivalent**.

In more complex systems, with perhaps thousands of perceptual states, it makes more sense to provide a sliding scale of equivalence. This would allow substantially similar agents to be considered "equivalent" even though they differ by a small amount. Recall the ideal laboratory where robots are evaluated by sweeping them through all possible sensory situations. When robots are compared at the policy level, the same effect can be achieved by checking the robots' policies for their response at every perceptual state.

The general idea is to compare two robots, R_a and R_b, by summing the differences between their responses, $\mid \bar{a}_a - \bar{a}_b \mid$, over all perceptual states, $\bar{i}$. It is also important to emphasize the response differences in perceptual states where the agents spend most of their time and to de-emphasize those that are infrequently experienced. This notion is encapsulated by multiplying the response difference in each situation by the probabilities of each agent $(P_i^a + P_i^b)$ being in that situation. Formally, **behavioral difference** between two robots R_a and R_b is defined as:

$$D(R_a, R_b) \quad = \quad \sum_{\text{for all i}} \frac{1}{2}(P_i^a + P_i^b) \mid \bar{a_a} - \bar{a_b} \mid \qquad (1)$$

If R_a and R_b select identical outputs $(\bar{a})$ in all perceptual states $(\bar{i})$, then $D(R_a, R_b) = 0$. When R_a and R_b select differing outputs in a given situation, the difference is normalized by the joint proportion of time they spend in that situation.

Now that a measure of behavioral difference is available, it is possible to define a type of equivalence using it. Two robots are ϵ-equivalent when their difference is less than ϵ:

Definition 1: R_a and R_b, are **ϵ-equivalent** iff $D(R_a, R_b) < \epsilon$.

Definition 2: $\equiv_\epsilon$ indicates ϵ-equivalence, $R_a \equiv_\epsilon R_b$ means R_a and R_b are ϵ-equivalent.

This in turn provides for definitions of societal homo- and heterogeneity:

Definition 3: A robot society, $\mathcal{R}$, is **ϵ-homogeneous** iff for all $R_a, R_b \in \mathcal{R}$, $R_a \equiv_\epsilon R_b$

Definition 4: A robot society, $\mathcal{R}$, is **ϵ-heterogeneous** iff there exists an R_a and $R_b \in \mathcal{R}$) such that $R_a \not\equiv_\epsilon R_b$

4 Social Entropy

The goal is to devise a metric which captures the following expectations regarding behavioral diversity:

- The least diverse society is one in which all agents are equivalent.
- The greatest diversity is achieved when no agent is equivalent to any other agent.
- A society in which one agent is different and all the rest are equivalent is slightly more diverse than the society in which all agents are the same.
- If two societies have uniformly-sized groups, the one with more groups is more diverse.

Before a proposed metric addressing these requirements is presented, the manner in which a robot society is partitioned into behavioral castes must be explained. The idea is to group agents into castes, according to their behavioral similarity, using ϵ-equivalence:

$$\mathcal{C} = \{C_1, C_2, C_3 ... C_c\}$$
$$\text{for all } R_a, R_b \in C_i, R_a \equiv_\epsilon R_b$$

$\mathcal{R}$ is broken into c castes, and each caste is ϵ-homogeneous. Observe that if $\epsilon > 0$, the sub-classes are not necessarily disjoint; one robot may belong to more than one class.

Given that a robot society is partitioned into castes how should a measure of its diversity be based on the partitioning? Consider $\text{Het}(\mathcal{R})$, a candidate function evaluating the heterogeneity of $\mathcal{R}$. The requirements listed above are restated more formally:

- $\text{Het}(\mathcal{R}) = 0$ iff $\mathcal{R}$ is ϵ-homogeneous.
- $\text{Het}(\mathcal{R})$ should be at a maximum when every robot in $\mathcal{R}$ is different.
- $\text{Het}(\mathcal{R})$ should be at a minimum (but still > 0) when only one robot in $\mathcal{R}$ is different from the others.
- If $\mathcal{R}_a$ and $\mathcal{R}_b$ contain uniformly-sized sub-classes, we should have $\text{Het}(\mathcal{R}_a) < \text{Het}(\mathcal{R}_b)$ when $\mathcal{R}_b$ is composed of more sub-classes than $\mathcal{R}_a$.
- If two robot societies, $\mathcal{R}_a$ and $\mathcal{R}_b$ have c_a and c_b sub-classes respectively, with robots uniformly distributed among the sub-classes, we should have $\text{Het}(\mathcal{R}_a) = \text{Het}(\mathcal{R}_b)$ iff $c_a = c_b$.

$H(X)$, referred to as *Information Entropy* meets all these criteria [5]. The Information Entropy of a symbol system X is used in coding theory as a lower-bound on the average number of bits required per symbol to send multi-symbol messages. X assumes discrete values in the set $\{x_1, x_2, x_3 ... x_c\}$ (the alphabet to be encoded). p_i is the probability that $\{X = x_i\}$. $H(X)$ measures the "randomness" of X as follows: Each $p_i \geq 0$ and the sum of the p_i's is 1.0. $H(X)$ is greatest in the "most random" case, where each x_i is equally likely, e.g. $p_i = \frac{1}{c}$. $H(X)$ is smallest in the least random case where some $p_i = 1$ and all the others are zero. The Information Entropy of X is given in bits as:

$$H(X) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2}$$

The log is taken to the base 2 because entropy is in bits. If symbols were coded in a ternary versus binary system the log would be to the base 3.

In appropriating $H(X)$ for use as a social metric, the robot castes correspond to the symbols to be coded (one symbol per caste), while the proportion of robots in each caste correspond to the probabilities of each symbol occurring in a message. $H(X)$ for a robot society is therefore the average number of bits required to specify which caste a randomly selected robot belongs to; a more diverse society requires more bits.

It is easy to adopt $H(\mathcal{R})$ as $\text{Het}(\mathcal{R})$ by specifying each p_i as the proportion of robots in the corresponding sub-class C_i:

$$p_i = \frac{|C_i|}{\sum_{j=1}^{c} |C_j|} \tag{3}$$

$$\text{Het}(\mathcal{R}) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{4}$$

Since ϵ-equivalence does not ensure disjoint sub-classes, p_i is normalized so that $\sum p_i = 1$. The other restriction, $p_i \geq 0$, is also met.

Now, to help clarify the use of these new metrics, they are applied in the evaluation of several example robot teams.

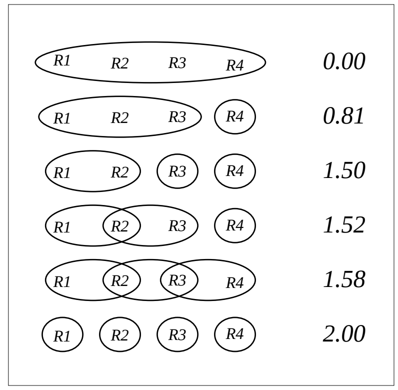

Figure 4: The Social Entropy of several example four robot societies. The ovals enclose ϵ-equivalent robots forming castes. The value of $\text{Het}(\mathcal{R})$ for each society is to the right.

5 Example Evaluations

We now return to the learning soccer robots introduced earlier. First, consider the behavioral difference between the forward and goalie policies of Figure 2. For comparisons between agents, behavior is considered at the policy level, so that robots with identical policies are considered equivalent. All the possible policies for these example agents are listed in Figure 3.

Following the terminology introduced in Section 3, there are two potential perceptual inputs for the soccer robots, $\bar{i} = 1$ and $\bar{i} = 0$, depending on whether the agent is behind the ball or not. Similarly there are three potential outputs: $\bar{a_1} = (0, 0, 1)$ (*move_to_backfield*), $\bar{a_2} = (0, 1, 0)$ (*get_behind_ball*) or $\bar{a_3} = (1, 0, 0)$ (*move_to_ball*). Manhattan distance is used here when evaluating vector differences, so that $\bar{a_1} - \bar{a_2} = (0, -1, 1)$ and $| \bar{a_1} - \bar{a_1} |= 2$.

In the case where the robots are behind the ball, $\bar{i} = 1$, both the goalie and forward select the same assemblage. They choose different assemblages when they aren't behind the ball. Assuming the robots spend the same amount of time in each of the two perceptual states, Equation 1 gives the behavioral difference between a goalie robot, R_g, and a forward robot, R_f, as:

$$\begin{aligned} D(R_g, R_f) &= \sum_{\text{for all i}} \frac{1}{2}(P_i^g + P_i^f) \mid \bar{a}_g - \bar{a}_f \mid \\ &= \frac{1}{2}(\frac{1}{2} + \frac{1}{2}) \mid (0,1,0) - (0,1,0) \mid + \\ &\quad \frac{1}{2}(\frac{1}{2} + \frac{1}{2}) \mid (0,0,1) - (1,0,0) \mid \\ &= \frac{1}{2}(1) \mid (0,0,0) \mid + \frac{1}{2}(1) \mid (-1,0,1) \mid \end{aligned}$$

$$= \frac{1}{2}(1)(0) + \frac{1}{2}(1)(2)$$
$$= 1$$

The behavioral difference between the goalie and forward agents is 1. Following this approach, we find that the maximum difference between any of the agent policies in Figure 3 is 2.

Now consider the Social Entropy, $\mathrm{Het}(\mathcal{R})$, of a heterogeneous team composed of one goalie and three forward agents (following the policies in Figure 2). First, we choose $\epsilon = 0$ so that robots must have identical policies to be considered equivalent. The society consists of four robots, $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$. One robot, R_4 (the goalie) is not ϵ-equivalent to the others so there are two castes, $C = \{C_1, C_2\}$, with $C_1 = \{R_1, R_2, R_3\}$ (the forwards' caste) and $C_2 = \{R_4\}$ (the goalie caste). Then,

$$\begin{aligned} p_1 &= .75 \\ p_2 &= .25 \\ \mathrm{Het}(\mathcal{R}) &= -\sum_{i=1}^{2} p_i \log_2(p_i) \\ &= -((p_1 \log_2(p_1)) + (p_2 \log_2(p_2))) \\ &= -((.75 \log_2(.75)) + (.25 \log_2(.25))) \\ &= .811 \end{aligned}$$

The Social Entropy of the control team is .811.

Finally we evaluate the Social Entropy of the homogeneous team in Figure 1. All four learning robots have converged to the forward behavior given in Figure 2. The team consists of the robots $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$. Homogeneity implies there is only one caste, so $\mathcal{C} = \{C_1\}$, and $C_1 = \{R_1, R_2, R_3, R_4\}$. Then:

$$\begin{aligned} p_1 &= 1 \\ \mathrm{Het}(\mathcal{R}) &= -\sum_{i=1}^{1} p_i \log_2(p_i) \\ &= -(p_1 \log_2(p_1)) \\ &= -(1 \log_2(1)) \\ &= 0 \end{aligned}$$

As expected, the Social Entropy of a homogeneous society is $\mathrm{Het}(\mathcal{R}) = 0$. This result generalizes to all homogeneous cases.

The entropies for several other four robot society examples are illustrated in Figure 4.

6 Conclusion

New metrics and definitions for multi-robot learning teams have been introduced, including

- A mathematical expression for the **behavioral difference** between two robots.
- Definitions of behavioral **homogeneity** and **heterogeneity** for multi-robot teams.
- **Social Entropy**, a new measure of robot team **behavioral diversity**.

Use of the metrics is illustrated through evaluations of a learning multi-robot soccer team. It is hoped that these metrics will serve as tools for future work in the evaluation of learning multi-robot teams.

The author thanks Ron Arkin, Chris Atkeson, Maria Hybinette and Juan Santamaria for helpful discussions on these topics.

References

[1] R.C. Arkin. Motor schema based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

[2] T. Balch. Clay: Integrating motor schemas and reinforcement learning. Coll. of comp. tech. report, Ga. Inst. of Tech., 1997.

[3] T. Balch. Learning roles: Behavioral diversity in robot teams. Coll. of comp. tech. report, Georgia Inst. of Technology, 1997.

[4] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *Proc. Autonomous Agents 97*. ACM, 1997. Marina Del Rey, California.

[5] C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[6] Christopher J. C. H. Watkins and Peter Dayan. Technical note: Q learning. *Machine Learning*, 8:279–292, 1992.