Appendix A

Motor Schema Formulations and Gain Values

A motor schema-based approach to robot behavioral design is used in this work [Ark89]. Individual motor schemas, or primitive behaviors, express separate goals or constraints for a task. As an example, important schemas for a navigational task would include **avoid_static_obstacles** and **move_to_goal**. Since schemas are independent, they can run concurrently, providing parallelism and speed. Sensor input is processed by perceptual schemas embedded in the motor behaviors. Perceptual processing is minimal and provides just the information pertinent to the motor schema. For instance, a **find_obstacles** perceptual schema which provides a list of sensed obstacles is embedded in the **avoid_static_obstacles** motor schema.

The concurrently running motor schemas are integrated as follows. First, each produces a vector indicating the direction the robot should move to satisfy that schema's goal or constraint. The magnitude of the vector indicates the importance of achieving it. It is not so critical, for instance, to avoid an obstacle if it is distant, but crucial if close by. The magnitude of the **avoid_static_obstacles** vector is correspondingly small for distant obstacles and large for close ones. The importance of motor schemas relative to each other is indicated by a gain value for each one.

This appendix reports the methods by which each of the individual primitive motor schemas used in this research compute their component vectors. The gain values and parameters for schemas used in each behavioral assemblage are also listed.

A.1 Foraging behaviors

Details regarding the high-level sequencing of behavioral assemblages for foraging are covered in Chapter 6. The motor schemas used to build the behavioral assemblages for foraging are described below. For each behavioral assemblage, several schemas are instantiated at once with appropriate parameter and gain values. The parameter and gain values used experimentally are listed in Table A.1.

- **noise**

  description: generates movement in a pseudo-random direction.
  parameter: P, persistence, the time in seconds between each change in direction.
  mathematical formulation: the vector is computed as follows:

$$\begin{aligned} V_{\text{direction}} &= \text{pseudo-random direction} \\ &\quad\ \text{between 0 and } 2\pi \\ V_{\text{magnitude}} &= 1 \end{aligned}$$

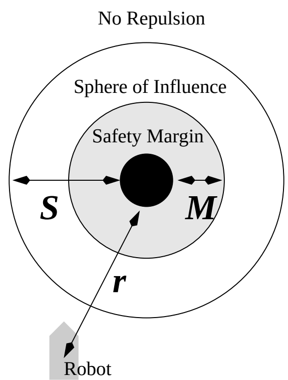

Figure A.1: Parameters used in the calculation of **avoid** motor schema vectors. The object to be avoided is represented as a black circle at the center of the illustration.

- **avoid_static_obstacles**

  description: repulsion from detected obstacles. The magnitude of repulsion varies with distance from each obstacle (Figure A.1). When beyond the sphere of influence (S), no repulsion is generated. Within the sphere of influence, repulsion increases linearly until the robot reaches the safety margin. When the robot is within the safety margin, the magnitude of repulsion is ∞.
  parameters: S, the sphere of influence beyond which detected obstacles have no effect; M, safety margin.

mathematical formulation: a separate vector is computed for each detected obstacle as follows, where r is the distance from the center of the robot to closest point on the obstacle:

$$V_{\text{direction}} = \text{along a line from the center of the obstacle to the robot, moving away from obstacle}$$

$$V_{\text{magnitude}} = \begin{cases} 0 & \text{for } r > S \\ \frac{S-r}{S-M} & \text{for } M < r \leq S \\ \infty & \text{for } r \leq M \end{cases}$$

The overall **avoid_static_obstacles** vector is computed by summing the individual vectors calculated for each obstacle.

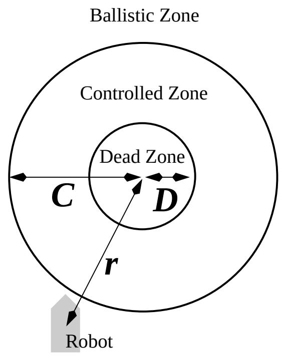

Figure A.2: Parameters used in the calculation of **move_to** motor schema vectors.

- **stay_near_homebase**
  move_to_red_attractor
  move_to_blue_attractor
  move_to_red_bin
  move_to_blue_bin

description: attraction to a detected object or goal location. The magnitude of attraction varies with distance from the goal. Figure A.2 illustrates three zones, defined by distance from the goal, used for magnitude computation. The radii of these zones are parameters of the schema. Outside the *controlled zone* attraction is set at a fixed maximum (1.0). Within the controlled zone attraction decreases linearly from 1.0 to 0.0 at the boundary of the *dead zone*. Inside the dead zone the magnitude is 0.0.

parameters: C, radius of the controlled zone; D, radius of the dead zone.

mathematical formulation: the vector is computed as follows, where r is the distance from the center of the robot to the goal location:

$$V_{\text{direction}} = \text{along a line from the robot to the goal, moving to the goal}$$

$$V_{\text{magnitude}} = \begin{cases} 1 & \text{for } r > C \\ \frac{r-D}{C-D} & \text{for } D < r \leq C \\ 0 & \text{for } r \leq D \end{cases}$$

- **avoid_robots**

description: repulsion from detected robots. The magnitude of repulsion varies with distance from each robot. When beyond the sphere of influence (S), no repulsion is generated. Within the sphere of influence, repulsion increases linearly until the robot reaches the safety margin. When the robot is within the safety margin, the magnitude of repulsion is ∞.

parameters: S, the sphere of influence beyond which detected robots have no effect; M, safety margin.

mathematical formulation: same as described for **avoid_static_obstacles**. A separate vector is computed for each detected robot and the vectors are summed to provide the overall value.

- **avoid_home_zone**

description: repulsion from the delivery area. Used to move agents away from this area during the search phase. The magnitude of repulsion varies with distance from the homebase. When far from the homebase, no repulsion is generated. As the robot approaches the area, repulsion increases linearly until the robot reaches the center of the zone.

parameters: S, the sphere of influence beyond which repulsion is zero.

mathematical formulation: the repulsion is computed as follows, where r is the distance from the center of the robot to the center of the homebase:

$$V_{\text{direction}} = \text{along a line from the homebase to the robot, moving away from the base}$$

$$V_{\text{magnitude}} = \begin{cases} 0 & \text{for } r > S \\ \frac{S-r}{S} & \text{for } r \leq S \end{cases}$$

- **avoid_delivered_attractors**

description: a localized repulsion centered on the delivery area with a small sphere of influence. Used to keep robots from pushing over previously delivered attractors.

parameters: S, the sphere of influence beyond which repulsion is zero.

mathematical formulation: the repulsion is computed as in the same manner as **avoid_home_zone**.

A.2 Soccer behaviors

Details regarding the high-level sequencing of behavioral assemblages for soccer are covered in Chapter 7. The motor schemas used to build the behavioral assemblages for soccer are described below. For each behavioral assemblage, several schemas are instantiated at once with appropriate parameter and gain values. Experiments were conducted in two slightly different domains (simplified and RoboCup soccer). Behaviors used in both domains are identical except the **avoid_teammates** schema was not used in the simplified domain. The parameter and gain values used experimentally are listed in Table A.2.

- **move_to_kickspot**

  description: attractive force to draw the robot to a point one-half of a robot radius behind the ball. If the robot bumps the ball from that location, the ball is propelled in the direction of the opponent's goal. Parameters used in this motor schema are illustrated in Figure A.2.

  parameters: C, radius of the controlled zone; D, radius of the dead zone.

  mathematical formulation: the vector is computed as described for the **move_to** schemas used in foraging.

- **move_to_halfway_point**

  description: attractive force to draw the robot to a point halfway between the ball and the defended goal. Parameters used in this motor schema are illustrated in Figure A.2.

  parameters: C, radius of the controlled zone; D, radius of the dead zone.

  mathematical formulation: the vector is computed as described for the **move_to** schemas used in foraging.

- **move_to_defended_goal**

  description: attraction to the defended goal. A large dead zone centered on the goal area permits the robot to roam freely if it is near the goal. Parameters used in this motor schema are illustrated in Figure A.2.

  parameters: C, radius of the controlled zone; D, radius of the dead zone.

Table A.1: Motor schema parameter values and gains used in the behavioral assemblages for foraging.

behavioral assemblage	**motor schemas**	**gain values**
wander	**noise**	1.0
search the environment for attractors	$P = 5.0$sec	
	avoid_static_obstacles	1.0
	$S = 1.5$m $M = 0.1$m	
	avoid_robots	1.0
	$S = 1.5$m $M = 0.1$m	
	avoid_home_zone	2.0
	$S = 4.0$m	
stay_near_home	**stay_near_homebase**	1.0
search the home zone for attractors	$C = 4.0$m, $D = 3.0$m	
	avoid_static_obstacles	1.0
	$S = 1.5$m $M = 0.1$m	
	avoid_robots	1.0
	$S = 1.5$m $M = 0.1$m	
	noise	1.0
	$P = 5.0$sec	
	avoid_delivered_attractors	1.0
	$S = 1.0$m	
acquire_blue	**move_to_{red|blue}_attractor**	1.0
acquire_red	$C = 0.4$m, $D = 0.0$m	
acquire the closest red or blue attractor	**avoid_static_obstacles**	1.0
	$S = 1.5$m $M = 0.1$m	
	avoid_robots	1.0
	$S = 1.5$m $M = 0.1$m	
	noise	0.2
	$P = 5.0$sec	
	avoid_delivered_attractors	1.0
	$S = 1.0$m	
deliver_red	**move_to_{red|blue}_bin**	1.0
deliver_blue	$C = 0.4$m, $D = 0.0$m	
go to the red or blue bin	**avoid_static_obstacles**	1.0
	$S = 1.5$m $M = 0.1$m	
	avoid_robots	1.0
	$S = 1.5$m $M = 0.1$m	
	noise	0.2
	$P = 5.0$sec	
	avoid_delivered_attractors	1.0
	$S = 1.0$m	

mathematical formulation: the vector is computed as described for the **move_to** schemas used in foraging.

- **avoid_teammates**

description: repulsion from robots on the same soccer team. The magnitude of repulsion varies with distance from each robot. When beyond the sphere of influence (S), no repulsion is generated. Within the sphere of influence, repulsion increases linearly until the robot reaches the safety margin. When the robot is within the safety margin, the magnitude of repulsion is ∞.

parameters: S, the sphere of influence beyond which detected robots have no effect; M, safety margin.

mathematical formulation: same as described for **avoid_static_obstacles** in the foraging section. A separate vector is computed for each detected robot and the vectors are summed to provide the overall value.

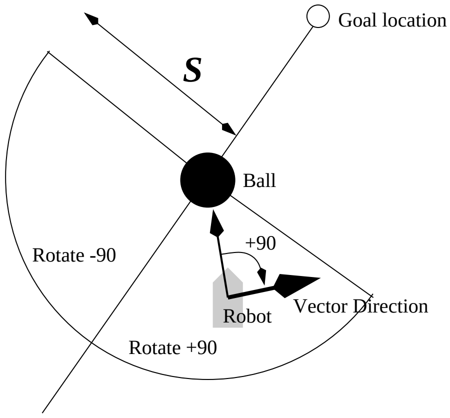

Figure A.3: Parameters used in the calculation of the **swirl_ball** motor schema vector.

- **swirl_ball**

description: a dodging behavior used to keep robots from colliding with the ball as they attempt to move behind it. The direction of the vector is perpendicular to a line drawn from the robot through the ball. The magnitude of the vector depends on distance to the ball. When beyond the sphere of influence (S), the magnitude is zero. Within the sphere of influence, magnitude increases linearly until the robot reaches the ball. If the robot is past the ball, the magnitude is zero.

parameter: S, the sphere of influence beyond which vector magnitude is zero.

mathematical formulation: the vector is computed as follows, where r is the distance from the center of the robot to the center of the ball:

$$V_{\text{direction}} = \text{a line from the robot to the ball}$$

rotated either +90 or −90
degrees such that the rotation
sweeps through the goal

$$V_{\text{magnitude}} = \begin{cases} 0 & \text{for } r > S \text{ or robot is past the ball} \\ \frac{S-r}{S} & \text{for } r \leq S \end{cases}$$

Table A.2: The **avoid_teammates** schema was not used in the simplified soccer domain experiments.

behavioral assemblage	**motor schemas**	**gain values**
move_to_ball	**move_to_kickspot**	1.0
	$C = 0.0$m $D = 0.0$	
	avoid_teammates	0.8 (RoboCup)
	$S = 1.0$m $M = 0.1$m	0.0 (simplified)
get_behind_ball	**move_to_halfway_point**	1.0
search the home zone for attractors	$C = 0.0$m $D = 0.0$	
	swirl_ball	1.2
	$S = 0.5$m	
	avoid_teammates	0.8 (RoboCup)
	$S = 1.0$m $M = 0.1$m	0.0 (simplified)
move_to_backfield	**move_to_defended_goal**	1.2
	$C = 0.3$m, $D = 0.2$m	
	move_to_kickspot	1.2
	$S = 0.0$m $M = 0.0$m	

A.3 Formation behaviors

The overall navigational strategy is similar to the approach developed in earlier research [BA95b]: several motor schemas, **move_to_goal**, **avoid_static_obstacles**, **avoid_robots** and **maintain_formation** implement the overall behavior for a robot to move to a goal location while avoiding obstacles, collisions with other robots and remaining in formation. An additional low-gain attractive force, **move_to_unit_center**, draws all of the robots together.

Formation maintenance is accomplished in two steps: first, a perceptual process, **detect_formation_position**, determines the robot's proper position in formation based on current sensor data; second, the motor process **maintain_formation**, generates motor commands to direct the robot toward the correct location. The perceptual process used by robots for determining their position in a formation is discussed in Chapter 8.

The motor schemas used to build the navigational behavioral assemblages for cooperative movement are described below. The *diamond*, *line* and *column* assemblages are identical except for the perceptual process used to determine the robot's desired position in formation. The parameter and gain values used experimentally are listed in Table A.3.

- **move_to_goal**

  description: attractive force to draw the robot to a goal location. The goal is positioned 1000 meters beyond the finish line. The robots never actually reach the goal in experimental trials because each trial terminates when the robots cross the finish line. Parameters used in this motor schema are illustrated in Figure A.2.

  parameters: C, radius of the controlled zone; D, radius of the dead zone.

  mathematical formulation: the vector is computed as described for the **move_to** schemas used in foraging.

- **maintain_formation**

  description: attractive force to draw the robot into the proper formation position. Parameters used in this motor schema are illustrated in Figure A.2.

  parameters: C, radius of the controlled zone; D, radius of the dead zone.

  mathematical formulation: the vector is computed as described for the **move_to** schemas used in foraging.

- **move_to_unit_center**

  description: a low-gain attractive force, added to draw all of the robots together. Parameters used in this motor schema are illustrated in Figure A.2.

  parameters: C, radius of the controlled zone; D, radius of the dead zone.

  mathematical formulation: the vector is computed as described for the **move_to** schemas used in foraging.

- **avoid_static_obstacles** same as described for foraging.
- **avoid_robots** same as described for foraging.
- **noise** same as described for foraging.

Table A.3:

behavioral assemblage	motor schemas	gain values
diamond	**avoid_static_obstacles**	1.1
line	$S = 2.0$m $M = 0.1$	
column	**avoid_robots**	1.1
	$S = 2.0$m $M = 0.1$m	
	move_to_goal	0.7
	$C = 0.0$m, $D = 0.0$m	
	maintain_formation	1.3
	$C = 1.0$m $D = 0.0$m	
	move_to_unit_center	0.6
	$C = 3.0$m, $D = 2.0$m	
	noise	0.1
	$P = 5.0$sec	
no_formation	**avoid_static_obstacles**	1.1
	$S = 2.0$m $M = 0.1$	
	avoid_robots	1.1
	$S = 2.0$m $M = 0.1$m	
	move_to_goal	0.7
	$C = 0.0$m, $D = 0.0$m	
	move_to_unit_center	0.6
	$C = 3.0$m, $D = 2.0$m	
	noise	0.1
	$P = 5.0$sec	

Appendix B

Spearman's Rank-Order Correlation Test

Spearman's rank-order correlation test enables us to determine whether rank-ordered data is correlated. The key concept of non-parametric (rank-ordered) correlation is this: if we replace the value of each x_i by the value of its *rank* among all the other x_i's in the sample, that is $1, 2, 3, ..., N$, then the resulting list of numbers will be drawn from a perfectly known uniform distribution. Of course we do the same for the y_is, replacing each value by its rank among the others in the sample. Let R_i be the rank of x_i among the other xs, S_i be the rank of y_i among the other ys, ties being assigned a mid-rank. The rank-order correlation is defined to be the linear correlation coefficient of the ranks:

$$r_s = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2}\sqrt{\sum_i (S_i - \bar{S})^2}} \tag{B.1}$$

A value of $r_s = 1$ indicates complete positive correlation, $r_s = -1$ indicates complete negative correlation and $r_s = 0$ indicates no correlation of the data.

The statistical significance of r_s is tested by computing the probability that x and y are uncorrelated and the the ranking occurred by chance (the null hypothesis):

$$p = \operatorname{erfc}\left(\frac{|r_s|\sqrt{N}}{\sqrt{2}}\right) \tag{B.2}$$

where erfc(x) is the complementary error function [PTVF88]. Values of p less than 0.05 indicate the correlation is statistically significant.

The *Numerical Recipes in C* software package was used to compute the values of Spearman's coefficient reported in this dissertation [PTVF88].

Bibliography

[AAMR88] C. Atkeson, E. Aboaf, J. McIntyre, and D. Reinkensmeyer. Model-based robot learning. Technical Report AIM-1024, MIT, 1988.

[AB97] R.C. Arkin and T.R. Balch. AuRA: principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2), 1997.

[ABN93] R.C. Arkin, T. Balch, and E. Nitz. Communication of behavioral state in multi-agent retrieval tasks. In *Proceedings 1993 IEEE Conference on Robotics and Automation*, Atlanta, GA, 1993.

[Ali97] K. Ali. Phd thesis proposal. College of Computing, Georgia Institute of Technology, 1997.

[AM94] R.C. Arkin and D. MacKenzie. Temporal coordination of perceptual algorithms for mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 10(3):276–286, June 1994.

[Ark89] R.C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, 8(4):92–112, 1989.

[Ark92] R.C. Arkin. Cooperation without communication: Multi-agent schema based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[BA93] T. Balch and R.C. Arkin. Avoiding the past: a simple but effective strategy for reactive navigation. In *IEEE Conference on Robotics and Automation*, pp 678–685, May 1993. Atlanta, Georgia.

[BA95a] T. Balch and R.C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 1995.

[BA95b] T. Balch and R.C. Arkin. Motor schema-based formation control for multiagent robot teams. In *First International Conference on Multiagent Systems*, June 1995. San Francisco.

[BA99] T. Balch and R.C. Arkin. Behavior-based formation control for multi-agent robot teams. *IEEE Transactions on Robots and Automation*, to appear.

[Bai90] K. Bailey. *Social Entropy Theory.* State University of New York Press, Albany, 1990.

[Bal] T. Balch. JavaBots. `www.cc.gatech.edu/~tucker/JavaBots`.

[Bal97a] T. Balch. Clay: Integrating motor schemas and reinforcement learning. College of Computing Technical Report GIT-CC-97-11, Georgia Institute of Technology, Atlanta, Georgia, March 1997.

[Bal97b] T. Balch. Social entropy: a new metric for learning multi-robot teams. In *Proc. 10th International FLAIRS Conference (FLAIRS-97)*, May 1997.

[Bal97c] T. Balch. Learning roles: Behavioral diversity in robot teams. In *AAAI-97 Workshop on Multiagent Learning*, Providence, R.I., 1997.

[Bal98] T. Balch. Integrating robotics research with JavaBots. In *AAAI-98 Spring Symposium*, March 1998.

[Bal98a] T. Balch. Personal communication. Georgia Institute of Technology, December 1998.

[BBC+95] T. Balch, G. Boone, T. Collins, H. Forbes, D. MacKenzie, and J. Santamaría. Io, Ganymede and Callisto - a multiagent robot trash-collecting team. *AI Magazine*, 16(2):39–51, 1995.

[Bel57] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, N.J., 1957.

[Bev90] H. Bevan. On the continuity between information theory entropy and statistical diversity. *Elektrontechnik und Informationstechnik*, 107(12):601–604, 1990.

[Boo97] G. Boone. Efficient reinforcement learning: Model-based acrobot control. In *IEEE International Conference on Robotics and Automation*, pp 229–234, Albuquerque, NM, 1997.

[Bro86] R. Brooks. A robust layered control system for a mobile robot. *IEEE Jour. of Robotics and Auto.*, RA-2(1):14, 1986.

[BSW90] A. Barto, R. Sutton, and C. Watkins. Learning and sequential decision making. In Michael Gabriel and John Moore, editors, *Learning and computational neuroscience : Foundations of adaptive networks*. M.I.T. Press, Cambridge, Mass, 1990.

[CAR92] R.J. Clark, R.C. Arkin, and A. Ram. Learning momentum: On-line performance enhancement for reactive systems. In *IEEE Conf. on Robotics and Automation*, pp 111–116, May 1992. Nice, France.

[CG93] C. Connolly and R. Grupen. On the applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–936, 1993.

[CGH96] D. J. Cook, P. Gmytrasiewicz, and L.B. Holder. Decision-theoretic cooperative sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1013–23, 1996.

[Com97] RoboCup Committee. Robocup real robot league regulations. `http://www.robocup.org`, 1997.

[CS96] P. Cheeseman and J. Stutz. Bayesian Classification (AutoClass): Theory and Results. Chapter in *Advances in Knowledge Discovery and Data Mining*, Kluwer, 1996.

[CSB65] J.M. Cullen, E. Shaw, and H.A. Baldwin. Methods for measuring the three-dimensional structure of fish schools. *Animal Behavior*, 13:534–543, 1965.

[Dem92] L. Demetrius. The thermodynamics of evolution. *Physica A*, 189(3-4):417–436, November 1992.

[DF94] A. Drogoul and J. Ferber. From tom thumb to the dockers: Some experiments with foraging robots. In *From Animals to Animats: Proc.*

2nd International Conference on the Simulation of Adaptive Behavior, pp 451–459, Honolulu, HI, 1994.

[DJW93] G. Dudek, E. Jenkin, and D. Wilkes. A taxonomy for swarm robots. In *Proc. 1993 IEEE International Conference on Intelligent Robots and Systems*, pp 441–447, 1993.

[FM97] M. Fontan and M. Mataric. A study of territoriality: The role of critical mass in adaptive task division. In *From Animals to Animats 4: Proceedings of the Fourth International Conference of Simulation of Adaptive Behavior*, pp 553–561, 1997.

[For92] U.S. Air Force. *Air Combat Command Manual 3-3.* Department of the Air Force, Washington, D.C., 1992.

[GM97] D. Goldberg and M. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings, AAAI-97*, pp 637–642, July 1997.

[HW90] B. Holldobler and E. Wilson. *The Ants.* Belknap Press, 1990.

[JS71] N. Jardine and R. Sibson. *Mathematical Taxonomy.* John Wiley & Sons, 1971.

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

[KLM96] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[Koz92] J. Koza. *Genetic Programming - On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, 1992.

[LCK95] M. Littman, A. Cassandra, and L. Kaelbling. Learning policies for partially observable environments: Scaling up. In *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[LDK95] M. Littman, T. Dean, and L. Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, May 1995.

[Lin91] Long-Ji Lin. Programming robots using reinforcement learning and teaching. In *Proceedings, AAAI-91*, 1991.

[Lin92] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, August 1992.

[Lin93] Long-Ji Lin. Hierachical learning of robot skills by reinforcement. In *International Conference on Neural Networks*, 1993.

[LVW83] D. Lurie, J. Valls, and J. Wagensberg. Thermodynamic approach to biomass distribution in ecological systems. *Bulletin of Mathematical Biology*, 45(5):869–872, 1983.

[LW80] D. Lurie and J. Wagensberg. Information theory and ecological diversity. In L. Garrido, editor, *Systems Far from Equilibrium*, pp 290–303, Springer-Verlag, Berlin, 1980.

[MAC97] D. MacKenzie, R. Arkin, and J. Cameron. Multiagent mission specification and execution. *Autonomous Robots*, 4(1):29–52, 1997.

[Mag88] A.E. Magurran. *Ecological Diversity and Its Measurement.* Princeton University Press, 1988.

[Mat92] M. Matarić. Designing emergent behaviors: From local interactions to collective intelligence. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 2*, pp 432–441, 1992.

[Mat94] M. Matarić. Learning to behave socially. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3*, 1994.

[Mat97] M. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, January 1997.

[May98] M.M. Mayoral. Renyi's entropy as an index of diversity in simple-stage cluster sampling. *Information Sciences*, 105(1-4):101–114, March 1998.

[MB90] P. Maes and R. Brooks. Learning to coordinate behaviors. In *Proceedings, AAAI-90*, pp 796–802, 1990.

[MC92] S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, pp 311–365, 1992.

[MCA95] D.C. MacKenzie, J.M. Cameron, and R.C. Arkin. Specification and execution of multiagent missions. In *IEEE International Workshop on Intelligent Robots and Systems (IROS '95)*, 1995.

[MGL92] I. Martinez, M.A. Gil, and M.T. Lopez. Analysis of mutual information measures in cluster sampling. *Applied Mathematics and Computation*, 52(2-3):389–402, December 1992.

[MM92] P. Meyer and S. McIntosh. The USA Today index of ethnic diversity. *International Journal of Public Opinion Research*, page 56, Spring 1992.

[MW89] Merriam-Webster. *Webster's ninth new collegiate dictionary.* Merriam-Webster, 1989.

[Nil84] N. Nilsson. Shakey the robot. SRI International Tech. Note 323, 1984.

[NMH96] I. Noda, H. Matsubara, and K. Hiraki. Learning cooperative behavior in multi-agent environment: a case study of choice of play-plans in soccer. In *Proc. of 4th Pacific Rim International Conference on Artificial Intelligence*, pp 570–579, 1996. Cairns, Autralia.

[PAR92] M. Pearce, R.C. Arkin, and A. Ram. The learning of reactive control parameters through genetic algorithms. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp 130–137, 1992. Raleigh, NC.

[Par93] L. Parker. Designing control laws for cooperative agent teams. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pp 582–587. IEEE, 1993.

[Par94] L. Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, M.I.T., 1994.

[PPM92] L. Pardo, J.A. Pardo, and M.L. Menendez. Unified (r,s)-entropy as an index of diversity. *Journal of the Franklin Institute*, 329(5):907–921, September 1992.

[PTVF88] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.

[RS93] A. Ram and J.C. Santamaría. Multistrategy learning in reactive control. *Informatica*, 17(4):347–369, 1993.

[Sah94] M.K. Sahota. Reactive deliberation: An architecture for real-time intelligent control in dynamic environments. In *Proceedings, AAAI-94*, pp 1303–1308. AAAI, 1994.

[Sha49] C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[SS73] P. Sneath and R. Sokal. *Numerical Taxonomy*. W. H. Freeman and Company, San Francisco, 1973.

[SSR98] J. Santamaria, R. Sutton, and A. Ram. Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive Behavior*, 6(2):163–217, 1998.

[Sto98] P. Stone. Personal communication. Carnegie Mellon University, April 1998.

[Sut91] R. Sutton. DYNA, an Integrated Architecture for Learning, Planning and Reacting. In *Working Notes of the AAAI Spring Symposium on Integrated Intelligent Architectures*, March 1991.

[SV98] P. Stone and M. Veloso. Layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, 12(2-3):165–188, 1998.

[SWS98] R. P. Sałustowicz, M. A. Wiering, and J. Schmidhuber. Learning team strategies: Soccer case studies. *Machine Learning*, 1998. To appear.

[Tes92] G. Tesauro. Practical issues in temporal difference learning. *Machine Learning*, 8:257–277, 1992.

[TVR96] J. Tsitsiklis and B Van Roy. An analysis of temporal-difference learning with function approximation. Technical Report LIDS-P-2322, M.I.T. Laboratory for Information and Decision Systems, 1996.

[Veh87] S. L. Veherencamp. Individual, kin, and group selection. In P. Marler and J.G. Vandenbergh, editors, *Handbook of Behavioral Neurobiology, Volume 3: Social Behavior and Communication*, pp 354–382, Plenum Press, New York, 1987.

[WD92] C. Watkins and P. Dayan. Technical note: Q learning. *Machine Learning*, 8:279–292, 1992.

[Wil92] E.O. Wilson. *The Diversity of Life*. Norton, 1992.

[YS93] H. Yanco and L. Stein. An adaptive communication protocol for cooperating mobile robots. In *From Animals to Animats: Proc. 2nd International Conference on the Simulation of Adaptive Behavior*, pp 478–485, Boston, 1993.