2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

A Bio-Inspired Analog Scheme for Navigational Control of Lightweight Autonomous Agents

Takis Zourntos, Nebu John Mathai, Sebastian Magierowski and Deepa Kundur

Abstract— **We present an approach for the development of lightweight analog cognition for autonomous navigation in unknown environments. We consider a hypothetical differential drive robot, equipped with a limited sensor suite that includes one front and one rear monocular obstacle sensor and a target (bearing and distance) sensor. A robotic motion control scheme is developed for obstacle avoidance and target-seeking using an innovative dynamical systems architecture incorporating a nonlinear controller derived using Lyapunov techniques. We claim that the avoidance-steering behavioral property of the agent is a weakly emergent characteristic. Simulations are provided and discussed.**

I. Introduction

For a wide range of applications in mobile robotics, a coordinated group of lightweight agents can outperform a more heavily-equipped processing-intensive single robot. A team of economical agents can provide superior coverage and a degree of redundancy that increases the likelihood of mission success. Indeed, for surveillance, disaster-relief or remote exploration, the multi-robot team is envisioned to supplant the paradigm of the single powerful automaton [1], [2], [3], [4].

In such distributed robotic systems, individual agents must possess ample capabilities despite stringent hardware constraints that limit available processing power and memory capacity. A number of lightweight behavioral control architectures exist, referred to generally as *reactive* robotic control schemes [5], ranging from Braitenberg-type designs used in swarm robotics, BEAM (Biology, Electronics, Aesthetics, Mechanics) control methods based on neural-like processing, to behavior-based approaches. As described in [5], each of these techniques is inspired by biology, and attempts to emulate various aspects of biological cognition. A primary reason for developing bio-mimetic robot control schemes is to achieve the high "intellect-to-mass" ratios exhibited by simple animals.

Consider that a honeybee, equipped with a brain that consists of roughly 10^6 neurons with a power "rating" of less than 1 mW, exhibits a wide range of social interactions, the capacity for learning as well as three-dimensional navigation, whereas a modern autonomous ground-based robot, equipped with several consumer-grade microprocessors, each of which consisting of approximately 10^7 transistors and rated at more than 10 W, lacks the behavioral sophistication of the insect [6] [7] [8] [9] [10] [11]. The divide between the natural and the artificial is so vast, that technological improvements alone (as dictated by Moore's Law) may be insufficient to close the gap, and there seems to be a clear need for the continued development of novel paradigms of cognition for robotics, especially those inspired by nature [12] [13].

Most forms of robotic control have drawn inspiration from features of the brain common to many species. For example, there is substantial evidence that computation in the brain is fundamentally analog, i.e., occurring on continuous scales of amplitude and time, and nonlinear [14], [15], [16], [17]. This has motivated the use of artificial neural network (ANN) control in robotics, and, while practical examples of such systems exist ([18], [19]), there is still no systematic approach that dictates the topology and parameter settings of the ANN from behavioral requirements.

But perhaps the most compelling aspect of biological cognition is the mysterious way by which behavior emerges from the nonlinear dynamics of the brain [20]. That is, while in software-programmed robots many behaviors can be predicted from scripted action "units" and are, therefore, not emergent in the sense of Bedau [20][1], behaviors generated by biological cognition are not predictable, or, at least, not straightforwardly so. For even if the full complement of deterministic differential equations describing the (relevant portion of the) brain is available, the nonlinearity and high order of the system tend to thwart efforts to determine an exact solution analytically. Thus we argue that, in general, biological cognition produces behavior in a manner consistent with Bedau's definition of weak emergence.

Even if one rejects the idea that biological cognition is unscripted, and that it is simply beyond our current capabilities to script behavior "dynamically," then the following question still remains: How does a nonlinear dynamical system encode behavior?

II. Contributions

We present a robotic control scheme for simultaneous target-seeking and obstacle avoidance in unknown environments. To the best of our knowledge, our approach is the first to demonstrate a systematically synthesized cognition described in terms of nonlinear dynamics that exhibits useful, weakly emergent behavioral properties. Specifically, we show that subsumed obstacle avoidance steering occurs without an explicit attempt on the part of the designers; no where in the behavioral design of the agent is the behavior "steer to avoid obstacles" explicit, yet, such steering behavior

The authors are with Texas A&M University, College Station, Texas, USA 77843-3128, except for S. Magierowski, who is with University of Calgary, Calgary, Alberta, Canada. Contact email: takis@neo.tamu.edu.

[1]Bedau states that a property P of a system S with dynamics D is called *weakly emergent* if, given known initial conditions and external inputs, the existence of P can be deduced only by simulation of D.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

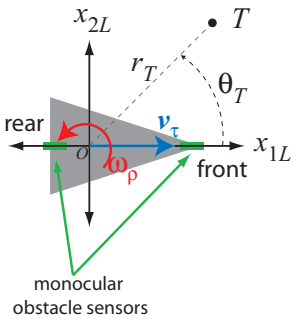

Fig. 1. Top view of agent. The chassis of the agent is indicated by the shaded triangle. A body frame of reference is indicated by the x_{1L}-x_{2L} coordinate system. Shown are the front- and rear-mounted obstacle sensors. The position of the target, denoted by T, is given by the polar coordinates (r_T, θ_T).

is evident from simulation. Implied steering information from sensors is also not available since the front and rear obstacle sensors are monocular.

III. Problem Formulation

We consider an environment represented in Euclidean space, $\mathcal{R}^m$, $m \in \mathcal{Z}$, $2 \leq m \leq \infty$. The target, obstacles and initial robot position are located within a compact subset of $\mathcal{R}^m$. Obstacles are assumed to be simply connected subsets of $\mathcal{R}^m$. A stationary target is positioned arbitrarily in the environment. Our derivations are valid for any m, however, for the purposes of simulation (presented in Section V), we shall impose the restriction $m = 2$ (i.e., navigation in the plane).

The agent or robot, also dimensionless (but with a specific orientation), has a local frame of reference, as shown in Figure 1. Not shown in the figure (but taken into account in simulations) is the underlying differential drive [21]. The robot may translate forward or reverse, in the direction of its x_{1L} axis, and rotate about its center (coincident with the origin of the local coordinate system, O). Translational speed is denoted by v_τ and rotational rate is denoted by ω_ρ.

The robot has no map of the environment, nor does it possess the ability to estimate its own global position or velocity. It is equipped with fore and aft monocular obstacle sensors for which readings use a class-L function (i.e., a bounded, continuous, positive and monotonically decreasing function) of the distance to the nearest obstacle point directly in front or behind the robot, respectively. The agent is also equipped with a target sensor which returns the distance and bearing to the target, denoted by r_T and θ_T, as shown in Figure 1. The robot has no other sensors.

The readings from the front and rear obstacle sensors are given by the polar-coordinate forms $\begin{bmatrix} -\sigma_{0,f} \\ 0 \end{bmatrix}$ and $\begin{bmatrix} \sigma_{0,r} \\ 0 \end{bmatrix}$, respectively, where $\sigma_{0,f}$ and $\sigma_{0,r}$ represent the class-L functions mentioned above. The overall obstacle sensor reading is formed as the virtual force

$$\boldsymbol{\sigma}_0 = \begin{bmatrix} -\sigma_{0,f} \\ 0 \end{bmatrix} + \begin{bmatrix} \sigma_{0,r} \\ 0 \end{bmatrix}. \quad (1)$$

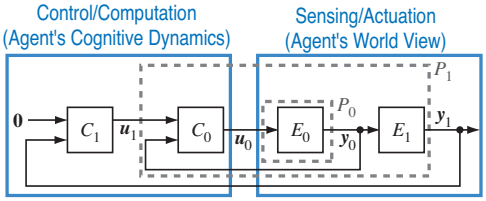

Fig. 2. Proposed analog-compatible robotic control scheme. The agent's dynamics are captured by "control/computation" and models of the world or the functioning of sensors (external to the robot) comprise the "sensing/actuation" portion of the scheme.

Note that there is no angular component in these obstacle sensor readings and so no explicit steering information is imparted to the agent.

Target sensing information is denoted by $\boldsymbol{\sigma}_1 = \begin{bmatrix} r_T \\ \theta_T \end{bmatrix}$, where r_T and θ_T denote distance and bearing to the target, respectively, as shown in Figure 1. To avoid singularities in the control law $\boldsymbol{u}_1$, described below, we impose the restrictions that $r_T \in [a, \infty)$, and $\theta_T \in [-b, b]$, where $a > 0$ and $0 \leq b < \frac{\pi}{2}$.

IV. Approach

Beer's formulation lays a foundation for us, describing an autonomous agent based on dynamical systems, in which an explicit internal state is associated with the agent [22]. This state suggests that the agent is not merely reflexively reacting to stimuli, but is processing information about its environment— a kind of deliberation— and acting accordingly. Our task is to provide a methodology to synthesize the agent's dynamics, and thus "fill in the blanks" of Beer's template.

The proposed cognition is organized into two levels or layers, as shown in Figure 2. The inner control loop, Level 0, consisting of "environmental" model E_0 and controller C_0, corresponds to basic actuation (locomotion via a differential drive). The outer control loop, Level 1, consisting of plant P_1 and controller C_1, is the motion planning (obstacle avoidance and target seeking) level.

At each level, we express the problem the agent needs to solve as an environmental model— in effect, translating the problem into a domain where analog control-theoretic techniques are natural— and design a controller that regulates the plant and hence achieves our objectives. The cognition or "behavior-generator" thus takes the form of an analog controller. The architecture follows naturally from this plant-controller co-design concept. The E_i blocks model some aspect of our path-planning problem (based on a combination of formal derivation and qualitative concepts), the controllers C_i are derived to achieve desired aims by considering the dynamics of the relevant E_i.

A. Level 0 Dynamics

Level 0 consists of the blocks C_0 and E_0, is the lowest level of the scheme, and encompasses a closed-loop differential drive system. The E_0 block models a pair of dc motors, each of which is described by the block diagram in Figure 3.

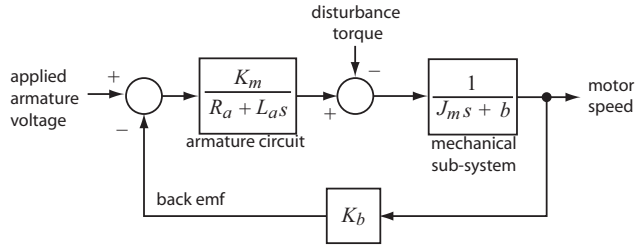

Fig. 3. Model for each motor. The block diagram describes an armature-controlled dc motor, with parameters $K_m = 0.00837$ Nm/A, $R_a = 10.4\,\Omega$, $L_a = 150\,\mu$H, $J_m = 6.7\times10^{-8}$ Nms^2/rad, $b = 10$ Nms, $K_b = 8.3747$ mV/(rad/s), and Laplace variable, s.

The translational speed, v_τ, and rotational rate, ω_ρ, of the agent are related to the left and right motor speeds, denoted by ω_L and ω_R, respectively, as

$$\omega_L = -\left(\frac{1}{\rho}\right) v_\tau + \left(\frac{\lambda}{2\rho}\right) \omega_\rho \tag{2}$$

and

$$\omega_R = \left(\frac{1}{\rho}\right) v_\tau + \left(\frac{\lambda}{2\rho}\right) \omega_\rho \tag{3}$$

where ρ represents tire radius and λ represents the center-to-center distance between the left and right tires. No slip conditions are assumed. The controller, C_0, is comprised of a pair of PID-type regulators, each of the form $c\frac{s+d}{s}$, where c and d are real constants.

It is important to note that from the perspective of Level 1, the closed loop system consisting of C_0 and E_0 is assumed to be the identity system, which is valid if the dynamics of Level 0 are sufficiently fast.

B. Level 1 Dynamics

The Level 1 system encompasses Level 0 (in its entirety) denoted by L_0, controller C_1 and model E_1. Level 1 represents both seeking and obstacle avoidance. Obstacle avoidance is handled implicitly by the control system which treats the obstacle sensor output as a exogenous or "noise" input that is suppressed.

The E_1 model is derived using geometry and describes the kinematic relationship between the velocity of the agent and convergence to the target. That is,

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{B}_{1,2}(\boldsymbol{\eta}) \begin{bmatrix} v_\tau \\ \omega_\rho \end{bmatrix} \\ \boldsymbol{\sigma}_1 &= \boldsymbol{\eta} \end{aligned} \tag{4}$$

in which $\boldsymbol{\eta} \in \mathcal{R}^2$ is the state vector, $\boldsymbol{B}_{1,2}(\boldsymbol{\eta}) = \begin{bmatrix} -\cos(\eta_2) & 0 \\ \frac{\sin(\eta_2)}{\eta_1} & 1 \end{bmatrix}$. While the model is not explicitly realized (it represents the target sensor), it is used in the analytic derivation of the controller, C_1. The C_1 control design problem is represented in Figure 4. The block $C_{1,1}$ describes a behavioral constraint expressed as

$$C_{1,1} : \begin{cases} \dot{\boldsymbol{\xi}}_{1,1} = \boldsymbol{A}_{1,1}\boldsymbol{\xi}_{1,1} + \boldsymbol{B}_{1,1}\boldsymbol{e}_1 \\ \boldsymbol{e}_B = \boldsymbol{G}_{1,1}\boldsymbol{\xi}_{1,1} \end{cases} \tag{5}$$

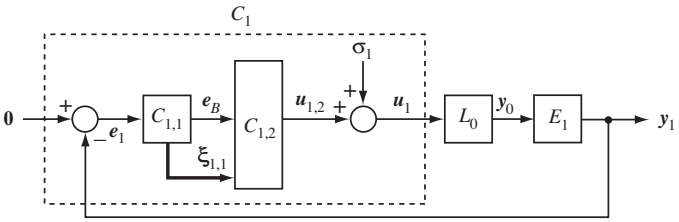

Fig. 4. Setup for design of C_1.

where $\boldsymbol{e}_B$ is the output of a low pass filter given by the stable controllable state model $(\boldsymbol{A}_{1,1}, \boldsymbol{B}_{1,1}, \boldsymbol{G}_{1,1})$. The regulation of $\boldsymbol{e}_B$ about zero imposes that the agent should arrive at the target, but not necessarily "right away" since $\boldsymbol{e}_B$ varies slowly relative to the error $\boldsymbol{e}_1$. The control $\boldsymbol{u}_1$ is derived using Lyapunov synthesis [23] based on the Lyapunov function

$$V_1 = \boldsymbol{\xi}_{1,1}^T \boldsymbol{P} \boldsymbol{\xi}_{1,1} + \frac{1}{2\varepsilon} \boldsymbol{e}_B^T \boldsymbol{e}_B + \boldsymbol{z}^T \boldsymbol{z} \tag{6}$$

where $\boldsymbol{P}$ is a positive definite matrix, $\varepsilon > 0$, and $\boldsymbol{z}$ is the backstepping error state (see [23] or [24]) given by $\boldsymbol{z} = \boldsymbol{y}_1 - \boldsymbol{u}_\alpha$ where $\boldsymbol{u}_\alpha$ is the virtual control (associated with backstepping). We derive that

$$\boldsymbol{u}_{1,2} = \boldsymbol{B}_{1,2}^{-1}(\boldsymbol{\eta}) \left[\boldsymbol{A}_1 \boldsymbol{\xi}_{1,1} - \rho \boldsymbol{A}_2(\boldsymbol{e}_B) \boldsymbol{s}(\boldsymbol{e}_B) \right] - \frac{\kappa}{2} \boldsymbol{z} \tag{7}$$

where $\boldsymbol{A}_1$ is a constant matrix, $\boldsymbol{A}_2(\cdot)$ and $\boldsymbol{s}(\cdot)$ are nonlinear mappings involving saturation and pulse functions, and ρ and κ are positive constants. This provides that $\|\boldsymbol{\xi}_{1,1}\|$ is bounded, $\|\boldsymbol{z}\|$ is bounded, and $\|\boldsymbol{e}_B\|$ is bounded and tends to zero as time approaches infinity. We have assumed, however, that $\boldsymbol{\sigma}_1$ is bounded (with a known bound) and, technically, this assumption is not accurate. This is a key approximation with our approach— that obstacles may be represented by bounded disturbances on the system. In fact, upon contact, the obstacle sensor output due to an immovable obstacle ought to match the agent's incident velocity and, therefore, $\boldsymbol{\sigma}_1$ should not be bounded (it is, however, since we've specified class-L functions to describe virtual forces). Nevertheless, the control $\boldsymbol{u}_{1,2}$, provides an arguably useful behavior that is geared toward lightweight machines (which, while not guaranteeing target convergence, may be adequate for the required application).

V. Simulations

The agent is simulated in three distinct obstacle courses, placed at various initial positions and orientations. We include several representative plots here with full animation videos available directly from the following URL[2]:
http://www.ece.tamu.edu/~takis/robotics_main_page_2.html

In these simulations, the robot is indicated by a conical shape, and its trace is shown in each plot. The target is depicted as a green torus. No physical contact is made between the robot and the obstacles (no physics is modeled in these simulations); the robot reacts to the local

[2]The reader may also navigate to the following URL and take the "Research" link: http://www.ece.tamu.edu/~takis.

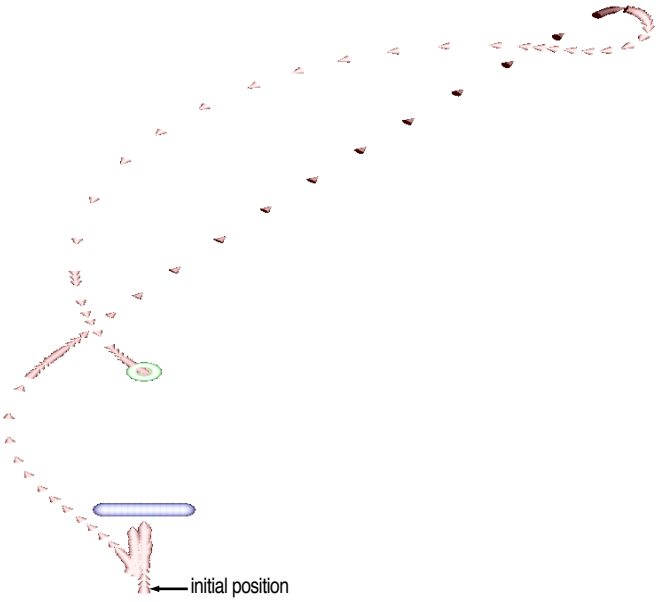

Fig. 5. Simulation #1.

obstacle detector output $\boldsymbol{\sigma}_1$ only. Simulations are conducted using Matlab™ and Simulink™, and execute rapidly on a PowerPC G4 processor (even for complex obstacle courses, simulations are completed within a couple of minutes). A custom C/C++ OpenGL application was developed to render the simulation [25].

The agent's controller parameters are the same in all simulations, and were chosen through a trial-and-error process. The simulated obstacle detection sensor characteristics also have a considerable influence on the agent's behavior. The sensors are based on an exponential function with a decay rate that is adjusted to yield different resolutions.

In the first pair of simulations, with results shown in Figures 5 and 6, we note how the agent deals with a fairly wide obstruction. In each simulation, the agent is initially positioned "behind" the wall, on the side furthest from the target. In Figure 5, we note that the agent first makes a number of forward and reverse motions, apparently attempting to pass through the obstacle by overcoming the virtual force. Within a short time, the agent begins to steer and takes a circuitous path to the target. We surmise that the filtering provided by $C_{1,1}$ makes such trajectories admissible. A close-up view of the first simulation is shown in Figure 7. In Figure 6, the agent is again positioned initially behind the wall, but is facing in the direction opposite the target. The resulting path is much more direct and straightforward, likely because the agent does not experience an opposing virtual force.

The next pair of simulations, Figures 8 and 9, show the agent coping with a regular arrangement of spherical obstacles. Both of these simulations appear to suggest the agent's obstacle avoidance strategy of steering. Convergence is successful in both cases.

In the final pair of simulations shown in Figures 10 and 11,

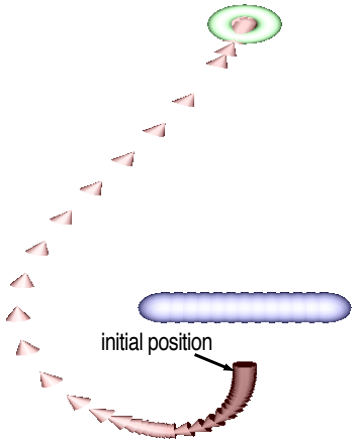

Fig. 6. Simulation #2.

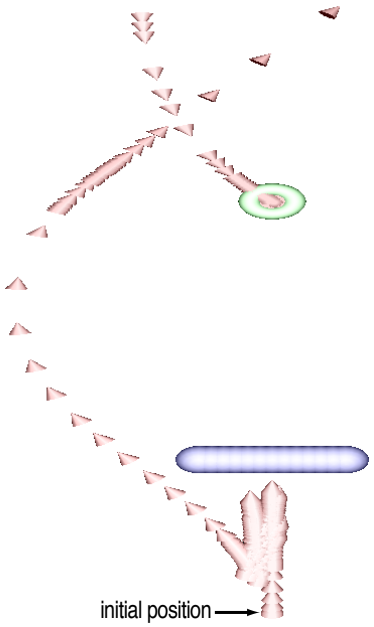

Fig. 7. Simulation #1, closer view (with trajectory truncated).

the agent is placed in a challenging maze-like environment. The first figure shows an unsuccessful attempt, since the agent, even after considerable time, fails to converge to the target. From a different initial position and orientation, however, the agent makes it to the target (in relatively little time), as shown in Figure 11.

VI. Conclusion

A hierarchical robotic control scheme developed using control-theoretic tools and inspired by neuroscience has been described and demonstrated. We answer the question posed in the introduction, namely, "How does a dynamical system encode behavior?" as "A dynamical system encodes behavior through the use of regulatory mechanisms— feedback control loops synthesized to achieve particular goals. The behaviors that result are simply the outputs of controllers." Because of its direct compatibility with dynamical systems,

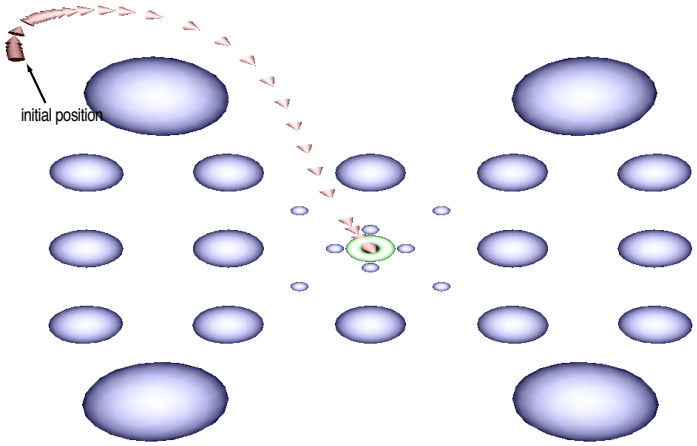

Fig. 8. Simulation #3.

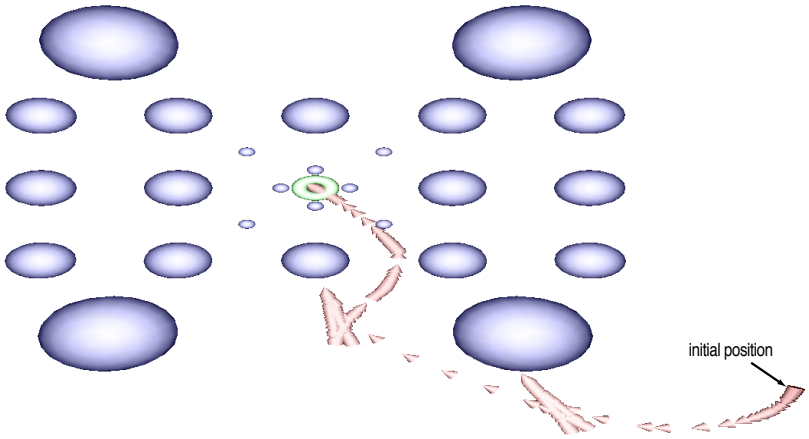

Fig. 9. Simulation #4.

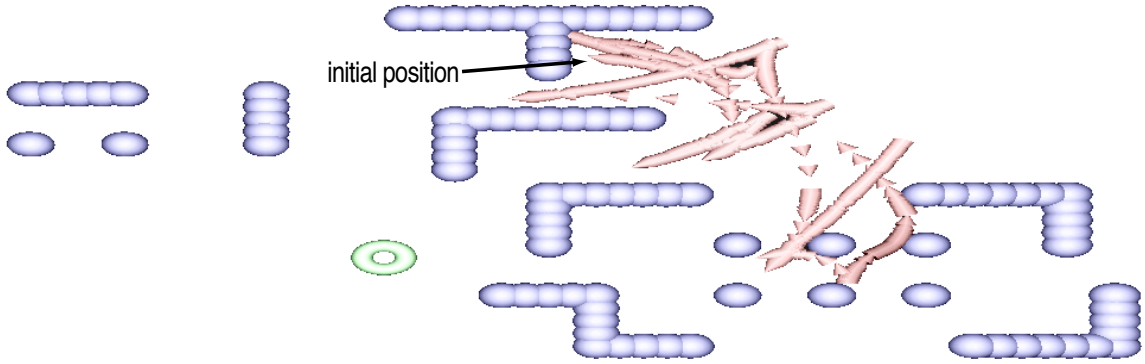

Fig. 10. Simulation #5.

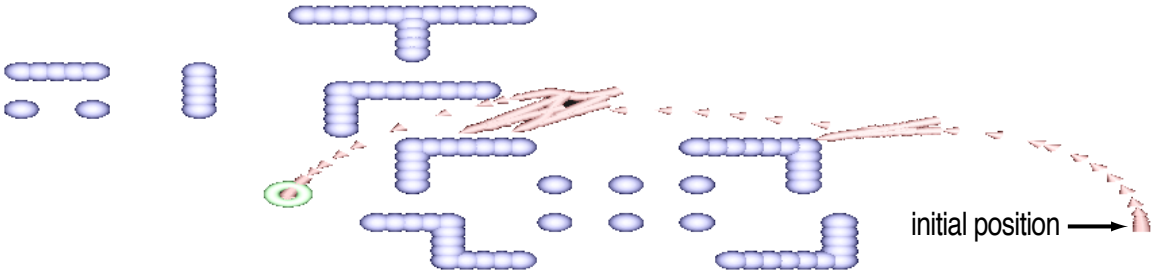

Fig. 11. Simulation #6.

control theory provides an ideal analog "programming language" to generate behavior for lightweight agents. We formulated target-seeking and subsumed obstacle avoidance as a tracking-control problem with disturbance-input suppression, and the approach appears to have practical merit (insofar as simulations may be considered valid empirical studies).

REFERENCES

[1] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, January 2007.

[2] R. Grabowski, L. E. Navarro-Serment, and P. K. Khosla, "An army of small robots," *Scientific American*, November 2003.

[3] R. Grabowski, L. E. Navarro-Serment, C. J. Paredis, and P. K. Khosla, "Heterogeneous teams of modular robots for mapping and exploration," *Autonomous Robots*, vol. 8, no. 3, pp. 293–308, 2000.

[4] M. J. Matarić, "Issues and approaches in the design of collective autonomous agents," *Robotics and Autonomous Systems*, vol. 16, no. 2-4, pp. 321–331, December 1995.

[5] D. Paluska, M. J. Matarić, and R. Ambrose, "Biomimetic robot control," in *Biologically Inspired Intelligent Robots*, Y. Bar-Cohen and C. Breazeal, Eds. SPIE Press, 2003, vol. PM122.

[6] S. Zhang, A. Mizutani, and M. V. Srinivasan, "Maze navigation by honeybees: Learning path regularity," *Learning & Memory*, vol. 7, pp. 363–374, 2000.

[7] R. Menzel and M. Giurfa, "Cognitive architecture of a mini-brain: the honeybee," *Trends in Cognitive Sciences*, vol. 5, no. 2, pp. 62–71, February 2001.

[8] M. Hammer and R. Menzel, "Learning and memory in the honeybee," *The Journal of Neuroscience*, vol. 15, no. 3, pp. 1617–1630, March 1995.

[9] Z. Huang. CyberBeeNet (Biology, Research, Beekeeping). [Online]. Available: http://www.cyberbee.net/

[10] J. Kumagai, "Halfway to mars," *IEEE Spectrum*, pp. 38–43, March 2006.

[11] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[12] M. A. Bedau, J. S. McCaskill, N. H. Packard, S. Rasmussen, C. Adami, D. G. Green, T. Ikegami, K. Kaneko, and T. S. Ray, "Open problems in artificial life," *Artificial Life*, vol. 6, pp. 363–376, 2000.

[13] R. A. Brooks, "The relationship between matter and life," *Nature*, vol. 409, January 2001.

[14] W. S. McCullough and W. Pitts, "A logical calculus of ideas immanent in nervous tissue," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[15] M. A. Arbib, *Brains, Machines, and Mathematics*, 2nd ed. Springer-Verlag, 1987.

[16] C. A. Skarda and W. J. Freeman, "Chaos and the new science of the brain," *Concepts in Neuroscience*, vol. 1, no. 2, pp. 275–285, 1990.

[17] M. A. Arbib, "Biological neurons and networks," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. The MIT Press, 1995.

[18] H. J. Chiel, R. D. Beer, R. D. Quinn, and K. S. Espenschied, "Robustness of a distributed neural network controller for locomotion in a hexapod robot," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 293–303, June 1992.

[19] R. E. Reeve and B. H. Webb, "New neural circuits for robot phonotaxis," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 361, pp. 2245–2266, 2003.

[20] M. A. Bedau, "Weak emergence," in *Philosophical Perspectives: Mind, Causation, and World*, J. Tomberlin, Ed., 1997, vol. 11, pp. 375–399.

[21] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[22] R. D. Beer, "A dynamical systems perspective on agent-environment interaction," *Artificial Intelligence*, vol. 72, pp. 173–215, 1995.

[23] H. K. Khalil, *Nonlinear Systems*. Prentice-Hall, 2002.

[24] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*. Wiley, 1995.

[25] D. Shreiner, M. Woo, J. Neider, and T. Davis, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Professional, 2005.