

Development of Control for a Serpentine Robot

William R. Hutchison, Betsy J. Constantine, Johann Borenstein, and Jerry Pratt

Abstract—This paper describes the development and testing of control of the OmniTread OT-4 robot by the Seventh Generation (7G) Control System. Control of OT-4 was developed in the Yobotics 3D simulator by an iterative process combining genetic algorithm, learning and analytic programming techniques. The control system developed in simulation was tested by controlling the real OT-4 robot in the laboratory. The performance of the real OT-4 robot under 7G control on stairs, parallel bars, a slalom course, and stairs with obstacles corresponded well to the simulated performance on which development of the control system was based.

I. INTRODUCTION

Serpentine robots have the potential to traverse a wide range of terrains and have unique advantages for terrains with narrow openings as in pipes and collapsed buildings and for very rough terrains with wide horizontal and vertical gaps that can be spanned by their long bodies.

A wide variety of serpentine robot designs have been developed, including several that use active wheels or tracks in addition to active joints for more effective but less snake-like locomotion. Among tracked robots, Souryu-I [1] and Millibots [2] had tracks that rotate around each segment, while MOIRA [3] and OmniTread OT-4 [4]–[6] have separate tracks on top, bottom, and the two sides.

The current study used the OT-4 robot, shown in Fig. 1, which has seven segments, each 8.2 cm wide, 8.2 cm high, and 10.3 cm long (center segment is 10.9 cm long). Its total length is 94 cm and it weighs 4.0 kg. Each segment has two tracks almost covering each of its four sides. The tracks are driven by worm gears on a shaft from a single electric motor in the center segment. Each two-degree-of-freedom joint is activated by four pneumatic bellows capable of a wide range of lengths and pressures, enabling the body to be stiff to cross wide gaps or relaxed to comply with uneven terrain. The maximum joint angle is approximately 40 degrees. At the time of this study, gas for the bellows was provided by onboard cylinders.



Fig. 1. The OmniTread OT-4 serpentine robot climbing parallel bars.

II. CONTROLLING SERPENTINE ROBOTS

Serpentine robots are very challenging to control, since a single operator cannot feasibly control their many degrees of freedom. The OT-4 robot can be controlled manually by three operators, each with two joysticks to control the six joints. This approach, however, was intended only for developing and testing the robot. In addition to the cost of having three operators, the operators require direct overview of the robot and terrain, making it impractical for teleoperation. For the OT-4 to be useful for real world applications, a control system that simplified operator control requirements was needed.

Several strategies have been used to simplify control. Takayama and Hirose [1] reduced the requirement on the operator for the three-segment, two-joint Souryu-I by programming the two joints to move in identical directions. Another strategy for reducing the control requirements is to use passive joints that need no actuation in at least one dimension, such as relaxing the joint in the sagittal¹ plane to comply with rough terrains. Often a joint will passively position itself effectively by relaxing, both in the sagittal plane (because gravity helps the robot conform to and maintain contact with terrain) and the lateral plane (from yielding to lateral force against objects or from being pulled by preceding segments). The Millibot [2] has active sagittal but passive lateral joints. The KOHGA robot [7] employed a mix of active and passive joints as a compromise approach to simplify control requirements. The OT-4 is especially well suited to using passive compliance because its bellows can vary the degree of stiffness independent of varying positions.

For many applications of serpentine robots, an appropriate mode of control is assisted remote teleoperation, in which the operator sets the overall goal direction and does medium scale path planning but is assisted by automated control of

This work was supported by the Intelligence Technology Innovation Center.

W. R. Hutchison is with Behavior Systems, Boulder, CO 80304 USA (phone: 720-289-0737; e-mail: whutchi@behaviorsys.com).

B. J. Constantine is with Context Systems, Carlisle, MA 01741 USA (e-mail: bconstantine@earthlink.net).

J. Borenstein is with The University of Michigan, Ann Arbor, MI 48109 USA (e-mail: johannb@umich.edu).

J. Pratt is with Yobotics, Inc., Cincinnati, OH 45206 USA (email: jpratt@yobotics.com)

¹ The terms sagittal and lateral will be used to refer to up-down and side-to-side movements of the robot body, respectively.

most of the robot's degrees of freedom. In this approach a human operator, viewing a video display of camera and other sensor data, controls movements of the head of the robot while an automated control system controls movements of the rest of the robot body using sensor data to determine movements of the robot body that are appropriate to local terrain features.

This paper describes the use of the Seventh Generation (7G) Control System [8]–[10] to develop a control system for the OT-4 robot that would assist a single operator to control the OT-4 by remote teleoperation. The work focused on controlling the OT-4 robot as it moved through several challenging terrains:

- Stairs of various configurations,
- parallel bars with random separations and heights, including wide gaps,
- a level slalom course among rocks, and
- stairs with rock obstacles.

A specific objective of the control development effort was to develop a single control system that would handle all terrains automatically, including transitions between them, instead of developing specialized control for each terrain.

III. METHOD FOR DEVELOPING CONTROL OF A SERPENTINE ROBOT

In the 7G Control System, an iterative process that combined learning, genetic algorithms, simulations and analytic programming was used to develop 7G control of the OT-4 robot.

A. The Seventh Generation (7G) Control System

The 7G Control System includes a neural network for reinforcement learning (RL), a customized genetic algorithm to optimize agent and training parameters and an automated training system, as shown in the diagram in Fig. 2. The core of the system, the 7G Robot Control Agent, is a reinforcement learning system implemented as a fully connected neural network for state-action value function approximation. At each discrete time step, the network receives sensory input and computes the estimated value for each action. The action selection system selects the highest valued action in each movement group² to be performed. The learning system then uses a temporal differences method [11] to modify the strengths of the connections between the active sensory input and actions performed at each time step. For this application, reinforcement learning strengthened actions that produced forward movement (computed by a simulation system) and weakened actions that did not.

The network architecture includes lagged nodes, which provide short term memory for previous situations and

² Mutually exclusive actions (e. g., raise joint 1, raise joint 2) are grouped into parallel movement groups, each movement group containing all possible actions for that group (e. g., raise joint 1, lower joint 1, do not change joint 1). Selected actions, one from each group, are performed simultaneously.

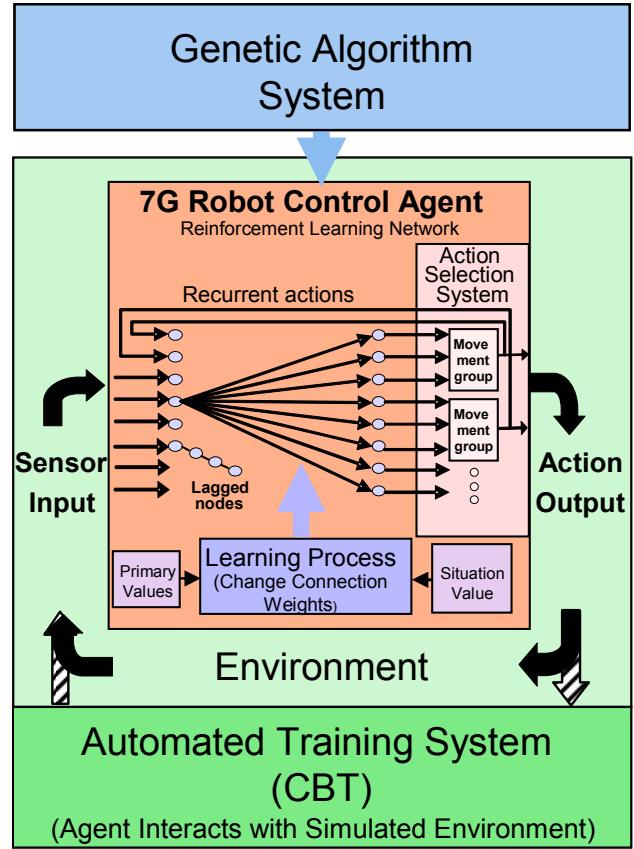


Fig. 2. Schematic diagram showing major elements of the 7G Control Agent. Sensory data from the environment enter the 7G Robot Control Agent, which generates actions in the environment.

The 7G Genetic Algorithm system evolves and optimizes parameters of agents and training methods to maximize a fitness function. During training of the network with the Automated Training System, sensory data are input by the training system and actions are delivered to the training system (striped arrows).

The reinforcement learning (RL) network contains direct sensor-to-movement connections. Learned network connection weights implement reactive behaviors in single activations of the network, with more complex functions implemented in multiple activations with recurrence. Output nodes have continuously changing values based on network activations.

The *action selection system* selects the most valuable behavior for each movement group in each sensor-behavior cycle. One action from each of the movement groups will be executed simultaneously.

Lagged nodes input the past states of an input node. *Recurrent connections* allow motor outputs to be sensed as inputs, making it possible to learn complex behavior such as sequences. Recurrence also enables a single-layer network to learn nonlinear relationships.

Primary values are the basis for the robot's goals via the RL process. The RL learning process also uses *situation value*, i.e., changes in the environment that are predictive of primary value.

actuator states, and recurrent connections, which provide the ability to learn coordinated locomotion sequences. To improve efficiency of learning in the reinforcement learning system, the 7G system includes a computer-based training (CBT) system to administer research-based training methods.

An integrated genetic algorithm (GA) customized for 7G evolves and optimizes parameters of agents and training methods to maximize the fitness function. The GA generates a population of control agents and works with the CBT to

train each agent in the population and test its fitness. The fittest agents are retained, parameters are modified, and new agents are trained and tested for fitness. The GA does not directly evolve network weights, but optimizes sensory, actuation, learning, and training parameters that enable best reinforcement learning. Examples of robot parameters in the GA are torques, distances within which to lift above obstacles ahead, and thresholds for touch sensor locations to trigger different behaviors. Examples of agent parameters include learning rate and length of short term memory for different types of sensory events. In this research, the GA was essential when hand-tuning of parameters did not produce the desired performance.

B. Using Scripts with Learning in the Background

Even with 7G's automated development tools, human expertise was used to program scripts relating a set of conditions to actions. The 7G agent's neural network can select either the script or a specific action in each movement group. If a specific action is chosen, normal neural network learning applies. If the script is chosen, learning is applied both to the script as an action and to the specific actions specified in the script. This approach requires that scripts always output actions that are also available as individual actions in the network. Then, depending on whether the script is successful or not, the reinforcement learning process reinforces or punishes both the script itself and the specific actions that the script specified. When the script fails, the network eventually stops selecting the script and tries other actions in the situations where the script failed.

C. 7G Control of OT-4 Robot Developed in Simulation

Learning and evolutionary methods require very large numbers of training trials and agents, so using a real robot to develop control is rarely feasible. It is far more efficient to conduct the control development process in a simulated environment where automated tools are available.

For the work reported here, simulations of the OT-4 traversing obstacle courses were developed by Hutchison using the Yobotics Simulation Construction Set [12], a physics-based 3D simulation system. The simulation model of the OT-4 was developed by Pratt of Yobotics in cooperation with Borenstein, developer of the OT-4 robot. Sensors that either existed (joint position sensors, accelerometers for tilt and roll) or would be feasible for the real OT-4 robot (IR distance sensors, and force sensors along the length of tracks) were simulated. Although the complex behavior of track grousers was not modeled explicitly, the track model with 40 ground contact points per track moving around the path of the tracks approximated the effect of grousers moving in contact with sharp edges. The functionality of the bellows was modeled as applied torque and joint damping. The maximum joint angle was set at 40 degrees in accordance with the robot specifications. Static and dynamic coefficients of friction for the tracks were initially set at 0.7 based on measurements of the OT-4 on

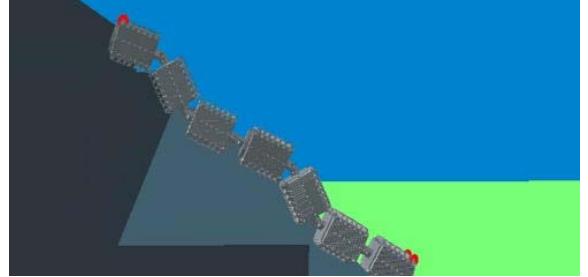


Fig. 3. Simulation of OT-4 robot climbing stairs.

typical stairs materials. Simulation of the OT-4 on smooth slopes confirmed that the simulator implemented friction correctly.

Models of the obstacle courses were constructed by Hutchison using available 3D models of rectilinear boxes. The 7G control software controlled the joint torques and track speed of the robot model in the simulation. The simulated robot and terrain are shown in Fig. 3.

IV. STAIR CLIMBING

Given the extension of long serpentine bodies over a length of terrain, many situations require different behaviors at different parts of the body to respond to local terrain features. Control cannot be based on implementing gaits that control the whole body. Uniform gaits may emerge at times, but only in homogeneous terrains or for global maneuvers such as arching to climb narrow passages. Stair climbing provides an example of the need for local control of different parts of the body.

Control of stair climbing is challenging due to the many movements that must vary continuously as the robot climbs, complicated by the tight coupling that creates many conflicts between joints for setting desired angles and causes many segments to be affected by every joint's movements. Although complex, control of stair climbing is a simplification over control of movements in complex natural terrains, since it is principally a problem of controlling sagittal head and body movements. Control of lateral movements is minimal since straight stairs can be climbed without turning.

An iterative process was used to develop effective control of stair climbing. An initial script for stair climbing was created and simulations were run in the Yobotics simulator on a simulated staircase of five steps with rises of 14.0, 15.2, 16.5, 17.8, and 19 cm (typical stairs have 17.8 cm rise). After several iterations of manually refining the script and observing the simulated behavior of the robot, 7G's integrated genetic algorithm and learning capabilities were applied. The GA was set up to evolve all parameters inherent in the script (about 20, depending on the script), along with other sensory and learning parameters. The fitness function was defined as the distance to the farthest point reached by the robot under control of a given control agent. The reinforcement learning network was set up with the script as

one of the actions selectable by the value-based action selection system, as described above. The GA and learning processes were iterative as well, with the developer continually refining the script on the basis of GA and learning results.

A. Integration of 7G Control with the OT-4 Robot

When robot control is developed in simulation, it is useful to cycle back and forth between simulations and live tests to confirm and maintain validity and to guide calibration of the simulations. Accordingly, 7G control of the OT-4 that was developed in simulation was tested in the laboratory by controlling the real OT-4 robot.

To test control of the real robot, the 7G control system was integrated with the control system for the OT-4 robot. Sensor data from the robot was sent to 7G for interpretation and processing. Control commands for each joint were sent from 7G to the OT-4 robot for execution of joint actions. In addition, joystick signals (stick position and override button) generated by a human operator for teleoperation of the head segment were sent from the joystick via the OT-4 system to 7G, which sent target joint angles for the joint behind the head segment to the OT-4 for execution.

B. Accommodations to Enable Testing of Control of Real OT-4 Robot

At the time of testing, the OT-4 robot had two potentiometers in each joint that provided information on joint angles and two accelerometers in each segment that provided information on the segment's pitch and roll relative to ground. The simulations showed, however, that to control stair climbing the OT-4 needed to sense not only joint angles and the pitch and roll of each segment, but it also needed to sense distance from the head to obstacles in the environment as well as the location of force applied to the tracks by the terrain. At the time of the validation tests, the OT-4 had neither distance sensors nor force sensors installed.

Fake Force Sensor Data. Since the OT-4 lacked force sensors for the real world test, it was proposed that humans generate fake force sensor signals. The method employed was for a human observer to use a joystick to input a signal for each segment indicating if and where force was being applied to a track. This required six joysticks (tail segment was ignored for now) operated by three people, each person assigned to watch two segments. The human observers watched the stair-climbing robot from its left side, such that the head of the robot was to the observer's left, as shown in Fig. 4. If force from the edge of a step was applied to a segment track, fake force sensor signals were generated by the humans via joystick movements as described in the figure. If no force was applied to the segment, or if force was applied uniformly along the track (e.g., when the segment was flat on the ground), the joystick was not moved. The fake force signals were sent to the 7G control system and interpreted as though they were data from force sensors on the tracks.

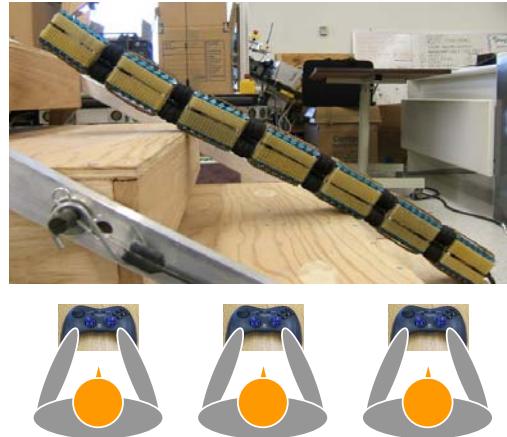


Fig. 4. Fake Force and Distance Sensor Data. Fake force sensor data was entered by three observers. Each sent fake force signals for two segments. If force from the edge of a step was applied to a segment track, the observer pushed the joystick for that segment forward. If the force was at the front of the track, the joystick was pressed forward and to the left, if in the middle, forward in the middle, or at the back, forward and to the right. If the segment moved across the edge of a step with the track touching continuously, the observer would press the joystick forward, moving the joystick continuously from forward-left to forward-right, mapping the location of the force. The observer assigned to segment 1 also sent fake distance signals.

Fake Distance Sensor Data. Since the OT-4 lacked distance sensors, the observer assigned to segment 1 generated simple fake distance sensor data by pressing a button on a joystick whenever an obstacle was within half a segment length from the front of the robot. The button press was sent to 7G and interpreted as though it were sensor data from an IR (or equivalent) distance sensor. The 7G algorithm developed in the simulations used the distance data to determine the appropriate target joint angles. In the stair climbing case, the commanded joint angles raised up to three segments sequentially as necessary to raise the front of the head above the obstacle, based on continuous input from the fake distance sensor button.

C. Testing OT-4 Climbing Stairs Under 7G Control

A test of the validity of 7G control of the OT-4 climbing stairs was conducted in Borenstein's University of Michigan Challenges Arena (UMCA), where several physical obstacle courses had been constructed for testing serpentine robots [13]. The adjustable steps in a variable staircase were set to a 16.5 cm rise and 28 cm tread.

For the test, the OT-4 was positioned on the floor perpendicular to the stairs and approximately 60 cm in front of the first step. Three observers, each with two joysticks for fake sensor signal input, were positioned as described in Fig. 4. After the drive motor was started manually (and kept at constant speed), the OT-4 autonomously climbed all four steps under 7G control based on real joint angle and accelerometer sensor data and fake force and distance sensor data.

V. PARALLEL BARS

Control of the OT-4 by 7G was also tested on a branch-like obstacle course, which consisted of an array of parallel bars of varying heights, as shown in Fig. 1.

Parallel bars differ from stairs in their variability in both height changes (both positive and negative) and lateral separation. The control agent developed for stair climbing was used for the parallel bars with one critical modification to accommodate the wide gaps between bars. The default relaxation response that was appropriate for stairs caused the robot to droop into a large gap, so an additional element was added to the script to prevent the front segments from drooping too far into a gap.

A. Testing Control of OT-4 on Parallel Bars

With the above addition to the script developed for stair climbing, and with no additional adaptation of parameters by the GA, the robot successfully traversed the parallel bars course. In an additional test in simulation, the OT-4 robot traversed a composite terrain composed of stairs followed by a series of parallel bars, with no changes to the parameters of the control system.

VI. ASSISTED TELEOPERATION

Control for assisted teleoperation of a serpentine robot must include control of lateral movements as well as the sagittal movements that were required for stair climbing. While the concept of assisted teleoperation assumes that the operator will control the general direction of the head of the robot, effective control of segments behind the head requires assistance from an automated control system. The objective of this phase of control development was to automatically control lateral body movements in response to the head movements commanded by an operator using a joystick.

Terrains containing various obstacles on a flat smooth surface were developed both in simulation and in the real world. Avoiding obstacles in such a terrain required lateral movements. Control of lateral movements was developed in simulated terrains, building on the control system used for stair climbing and parallel bars to demonstrate that a single control system could successfully control lateral movements as well as sagittal movements for head and body.

A. Control of Lateral Movements by Propagation

In contrast to the typical approach of propagating head movement to all following joints, the approach to propagation in the current work makes each joint's movement the same as the movement executed by the previous joint at the same location in space. The reason for this approach is that the information implicit in the prior joint action is the most recent and the response requirements that vary with body position are more likely to be similar. This approach depends on the ability to estimate the location of each joint as it moves through space. For this test distance was approximated as speed times elapsed time, multiplied by

a calibration factor. It will be important in the future, however, to have odometry that adjusts for slippage and turning. The propagation approach was implemented by keeping a circular buffer with the matrix of estimated odometer locations for each joint at the time of each sensor/control cycle.

Shan provided a partial precursor to this approach [14]. Shan developed an algorithm for his segmented robot that controlled contact with obstacles during motions on 2D smooth surfaces. The purpose was primarily to avoid damage to the robot, but sometimes the motions facilitated forward movement due to the resolution of the forces, as was the case in the current approach.

B. Teleoperation Window

Remote teleoperation requires some form of data display, typically video, for the operator to use to determine appropriate control commands. Since the OT-4 robot did not have a camera installed at the time of the tests, a wireless, battery-operated camera with wide angle lens was mounted temporarily. The camera sent color video at 15 frames per second to the 7G control system for display on the teleoperation window. Images from the video camera were rotated to maintain world coordinates (up in space remained up in the image, even when the robot rolled to any angle) based on accelerometer data.

C. Testing Control of Lateral Movements

Assisted teleoperation of the real OT-4 was tested in an obstacle course consisting of rocks placed on a smooth floor such that there was no straight path through the rocks. The operator's task was to move through the rock bed while turning to avoid the rocks. The operator sat behind a screen that blocked direct viewing of the robot and the obstacle course, using the video displayed in the teleoperation window to determine control commands for the head of the robot. The operator used one joystick to determine head position (up, down, left, right) and a throttle speed/direction slider to determine motor speed/direction. Joystick commands were sent to the 7G control system, which issued control commands for every joint based on approximate odometry and propagation. Joystick commands were interpreted relative to world coordinates as seen in the righted video.

Joints behind the head were controlled by 7G using propagation as described above. Although more precise odometry would have improved the performance of the robot in the slalom course, the success of the robot's traversal of the slalom course demonstrated that the same system that controlled sagittal movements could also control propagation-based lateral movements.

VII. CLIMBING STAIRS WITH OBSTACLES

The ability of the 7G control system to control both sagittal and lateral movements was tested directly. For this

test rocks were placed on the stairs used in the first test such that there was no straight path up the stairs, requiring the robot to move laterally to avoid the rocks while climbing the stairs. The 7G control system that was developed to control sagittal movements for stair climbing and then extended to control lateral movements for assisted teleoperation was used without modification.

A. Testing Control of OT-4 Climbing Stairs with Rocks

Under 7G control, the OT-4 robot successfully climbed four stairs (16.5 cm rise and 28 cm tread) while moving laterally in response to joystick commands from an operator to avoid the rocks on the steps, as shown in Fig. 5. The 7G control system provided autonomous control of sagittal movements for stair climbing, using the fake force and distance sensor input method described in the first test. To avoid the rocks on the stairs, an operator controlled lateral movements of the head segment, while the 7G propagation algorithm controlled lateral movements of the other joints. Unlike the previous test using a remote camera, the operator could see the robot and its terrain during this test.

While the tracked robot climbed stairs, it was not unusual for the track to slip while in contact with the stair edge. This slippage caused errors in the simple approximation of odometry used for the propagation of lateral movements from one joint to the next. When major errors in odometry occurred due to slippage, the propagation algorithm caused joints to turn too soon. This sometimes had benefits that offset its negative effects, since this lateral movement increased pressure between the side tracks and the rocks, although at other times the pressure was on the wrong side to propel the robot forward and had the potential to damage the robot and environment. When slipping ceased and forward movement resumed, however, the corrected movements in the propagation table would update quickly—in fact, given the approach of propagating from the previous joint rather than the head, the table could be corrected within one segment length of movement.

VIII. DISCUSSION

This study succeeded in developing control for a robot with complex sensors and complex movement patterns in challenging terrains. The problem was found to be too complex to solve analytically, and to cope with this complexity it was necessary to use the integrated development tools in the 7G system: GA, scripts with learning, physics-based simulation, and a user interface enabling developers to participate in improving control. The development tools did not solve the entire problem at once, but repeated iterations with incremental improvement each cycle solved the series of terrain challenges. It was very encouraging that the control system developed in simulation could be implemented on the real robot and successfully control it in corresponding real world terrains.



Fig. 5. The OT-4 robot climbing stairs with rock obstacles under 7G control. The inset in the upper left corner shows a side view of the robot at the same instant.

REFERENCES

- [1] T. Takayama and S. Hirose, “Development of Souryu-I connected crawler vehicle for inspection of narrow and winding space,” *26th Annual Conf. IEEE Ind. Electronics Society (IECON)*, Nagoya, Aichi, Japan, pp 143-148, Oct. 22-28, 2000.
- [2] H. Brown, J. Weghe, C. Bererton, and P. Khosla, “Millibot trains for enhanced mobility,” *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 452-461, 2002.
- [3] K. Osuka, and H. Kitajima, “Development of mobile inspection robot for rescues activities: MOIRA,” *Proc. 2003 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, October 27-31, 2003.
- [4] G. Granosik, M. Hansen, and J. Borenstein, “The OmniTread serpentine robot for industrial inspection and surveillance,” *Intl. J. on Industrial Robots*, Special Issue on Mobile Robots, IR32-2, pp. 139-148, March 18, 2005.
- [5] J. Borenstein, G. Granosik, and M. Hansen, “The OmniTread Serpentine Robot Design and Field Performance,” *Proc. SPIE Defense and Security Conference: Unmanned Ground Vehicle Technology VII*, Orlando, Florida, March 28-April 1, 2005.
- [6] J. Borenstein and M. Hansen, “OmniTread OT-4 Serpentine Robot – New Features and Experiments,” *Proc. SPIE Defense and Security Conf., Unmanned Systems Technology IX*, Orlando, Florida, April 9-13, 2007.
- [7] T. Kamegawa, T. Yamasaki, H. Igarashi, and F. Matsuno, “Development of the snake-like rescue robot KOHGA,” *Proc. 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, pp. 5081-5086, April, 2004.
- [8] W. Hutchison and B. Constantine, “Selection by value: Biologically inspired learning for mobile robot control,” *Journal of Intelligence Community Research and Development*, 2005.
- [9] W. Hutchison and B. Constantine, “Approach to validating a simulation of robot behavior: Human control of real robot replicates simulated robot behavior,” unpublished.
- [10] W. R. Hutchison, B. J. Constantine, J. Borenstein, and J. Pratt, “Developing control of a high-DOF robot using reinforcement learning, genetic algorithms, scripting, and simulation, submitted to 2007 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems.
- [11] R. Sutton and A. Barto, *Reinforcement learning: An introduction*, Cambridge, MA, MIT Press, 1998.
- [12] www.yobotics.com.
- [13] J. Borenstein, “The University of Michigan Challenges Arena,” unpublished.
- [14] Y. Shan and Y. Koren, “Design and motion planning of a mechanical snake,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 4, pp. 1091-1100, 1993.