

Developing Serpentine Robot Control to use Lateral Affordances

William R Hutchison*, Betsy J Constantine and Jerry Pratt

MeMeMe Inc, 1470 Birchmount Rd, Scarborough, ON M1P 2G1, Canada

Abstract

This paper describes the unique challenges in developing control of complex lateral movements needed by a serpentine robot to ascend steep slopes by climbing over and around affordances/obstacles on the slope. The research extends previous serpentine robot work developing control of sagittal movements for climbing up stairs and over uneven parallel bars. Effective lateral control was developed using an iterative combination of learning, a genetic algorithm, and developer programming. The robot's many simultaneous movements were controlled mostly as a function of very local sensory inputs and little centralized coordination.

Keywords: Robots; Serpentine robot; Genetic algorithm; Local sensory inputs.

Introduction

Snakes can traverse as wide a range of terrains as almost any animal, with unique advantages in terrains with narrow passageways and wide horizontal and vertical gaps that can be spanned by their long bodies. In hopes of duplicating this range of mobility, a number of serpentine robots have been developed in recent years. Early serpentine robots tended to imitate snake locomotion, so had active joints but no wheels or tracks on the segments [1-3]. The most effective serpentine robots across a wide range of terrains have active joints and active tracks or wheels on the segments, including the Souryu-I [4], Millibots [5], Moira [6] and OmniTread OT-4 [7-9]. Much research has focused on developing methods for controlling lateral movements involving only ventral (downward) contact, but many of the terrains of interest for these robots require the use of lateral affordances, sometimes referred to as obstacles. Our previous work has focused on sagittal (i.e., up and down) control to use affordances such as steps and bars, while the current study focuses on lateral movements to use affordances typically found in natural environments. Transeth et al. [10] provide a sophisticated analysis of the physics of lateral movements with affordances, but they do not provide a control strategy to enable the robot to actively locate and use affordances not appropriately located along the robot's default path.

Controlling Serpentine Robots

Like other complex robots, serpentine robots are very challenging to control because of their many degrees of freedom (DOF) as well as the complexity of sensors needed in many terrains.

Approaches to controlling serpentine robots

Three approaches that have been used to control serpentine robots are briefly described.

Manual operation: The earliest form of control for these robots was manual control, where operators could see the robot and terrain. Manual operation enables demonstrations of the robot's mobility, but the complexity of movements and the need for complete overview means it cannot be used in most situations of interest to potential users. For example, the OT-4 robot in manual mode requires three operators for six joints, and usually speed will be well below the robot's physical limits even in terrains requiring control mainly in only one plane at a time.

Aim and propagate: The most common control approach potentially usable in real applications takes advantage of the fact

that the bodies of active drive serpentine robots usually follow approximately the same path as the head. The human operator controls head movements while observing the robot or viewing a display from a camera mounted on the robot, and automated control propagates the head's actions back to following segments, controlling each joint when it reaches the same location along the path. Propagation algorithms have been based on either fairly straightforward odometry or analytic methods such as clothoids adapted from rail track analyses [3,11,12]. Passive and active propagation of head movements can be adequate for certain terrains, such as maneuvering on level ground through obstacles, up easy stairs, or through a sewer line. However, for the more demanding terrains, there are several major flaws with propagation of head movements as a sole strategy:

1. The head is a special case whose joint positions often cannot or should not be copied by subsequent joints;
2. Deformation of the terrain from compression or displacement may require different movements for different segments;
3. Changes in robot orientation from rolling, slipping, and pitching often alter movement requirements for subsequent segments; and
4. Operators often need automated assistance with head movements, not just body movements.

Fixed gaits

Evolutionary control development methods, including genetic algorithms (GAs) [13] and direct policy search [14], have been applied successfully to a number of robot control problems. Dowling demonstrated the use of GAs to develop control for a range of serpentine movement patterns, including snakelike gaits (e.g., lateral undulation, side winding) and non snake like gaits (e.g., rolling like a wheel).

Uniform gaits can be effective on smooth surfaces or easy terrains,

*Corresponding author: William R. Hutchison, MeMeMe Inc, 1470 Birchmount Rd, Scarborough, ON M1P 2G1, Canada, Tel: (416) 972-0973; Fax: (416) 972-9592; E-mail: hutchison.w@gmail.com

Received September 10, 2013; Accepted October 09, 2013; Published October 11, 2013

Citation: Hutchison WR, Constantine BJ, Pratt J (2013) Developing Serpentine Robot Control to use Lateral Affordances. Adv Robot Autom 2: 108. doi: 10.4172/2168-9695.1000108

Copyright: © 2013 Hutchison WR, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

but uniform gaits will fail in complex natural terrains because the terrain may vary along the length of the serpentine body, requiring different kinds of movements at different points along the body. Even a simple transition from one terrain type to another may require different behaviors of segments before and after the transition point in the terrain, with each behavior chosen based on local sensory input.

Problems in controlling serpentine robots: Serpentine robots are difficult to program or manually operate in real time because many coordinated movements requiring interpretation of complex sensor data from external and internal sensors are required to traverse natural terrain features.

In addition to the general problems associated with control of high-DOF robots, serpentine robots present three major control challenges.

1. Tight coupling prevents independent movement of segments, because:

- The limits on joint angles potentially constrain the positions of all coupled segments,
- Actuating each joint affects all attached segments, and
- The leverage of distal coupled segments can often prevent movement when torque is applied.

This is in stark contrast with legged robots where most target leg movements can be executed relatively independently of each other.

2. Situational awareness is much more of a challenge for both autonomous control and teleoperation because most of body is behind the head. Some solutions and their drawbacks include:

- A rear-mounted camera, which has been implemented or planned on several robots [11], but many terrains do not allow seeing the front from behind (or vice versa);
- Touch sensors, which are valuable for knowing what is around the body, but contact is often lost and 3D spatial memory is very challenging; and
- Odometry, which is essential, but existing contact and visual odometry are often poor due to longitudinal and lateral slippage, rolling, terrain gaps, turning, and 3-dimensionality.

3. The extended body of a serpentine robot makes it very likely that different kinds of movement strategies will be necessary simultaneously at different points along the body. Each movement must be suited to its local terrain conditions, making uniform gaits and propagation of head movements ineffective in many situations.

Developing control of the Omni Tread OT-4 serpentine robot

The current study aimed to develop teleoperational control that would succeed in a range of challenging terrains. In this approach, the operator sets the overall direction and the control system provides assistance to the operator for close-in head movements and completes control of all other body movements. The goal was to have a single control system handle all target terrains automatically, including transitions between them, rather than having specialized control for each terrain. Not only would the latter approach require knowing when to switch, but even the best switching would sometimes fail in transitions or in heterogeneous terrains.

The OmniTread OT-4 serpentine robot: The OmniTread OT-4 robot, shown in Figure 1, 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



Figure 1: The Omni Tread OT-4 serpentine robot climbing over parallel bars under automated control.

length is 94 cm and it weighs 4.0 kg. Each segment has four sides, each side nearly covered by two tracks. The tracks are driven by worm gears on a shaft from a single electric motor in the center segment. Each two-DOF 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. Gas for the bellows can be provided by onboard cylinders or an onboard compressor. The OT-4 is equipped with accelerometers on each segment to sense pitch and roll and potentiometers on each joint to sense joint pitch and roll.

Control Developed With the Seventh Generation Control System

Control of the OmniTread OT-4 serpentine robot was developed using the Seventh Generation (7G) Control System, a software system implementing a combination of reinforcement learning and genetic algorithms along with extensive support for sensory preprocessing, simulations, and user interface components [15,16]. Control was developed in simulation in a series of iterations using the Yobotics! Simulation Construction Set [17]. In each iteration, the developer made manual changes to a set of programmed behaviors (“scripts”) and used learning and a GA in repeated simulation runs to optimize the sensory and behavior parameters [16]. The robot’s movements were grouped into four logical movement groups: sagittal and lateral head movements and sagittal and lateral bod movements. For example, the head sagittal movement group includes lift, push down, relax, hold, and continue previous action. Note that if every joint’s sagittal and lateral movements were specified as a movement group, the agent would have at least 12 groups. Following a strategy that can often be applied to many high-DOF robots, however, a single movement group produced default sagittal behaviors for all joints behind the first one. The default behaviors could be overridden in various situations for the different segments (e.g., special tail behaviors). The same strategy was applied to lateral body behaviors.

Development of sagittal control

Initial development focused on controlling the OT-4 in two terrains, stairs and a series of uneven parallel bars that could be traversed by controlling only sagittal movements. The sagittal control program is briefly described as follows. The head was commanded to lift when obstacles were sensed within about half a segment length ahead of the robot if the top of an obstacle was above a line sloping upward ahead of the “chin.” If the head was lifted nearly to its limit and the obstacle was still above it, the second joint was lifted, and similarly for the third joint. When the bottom track of the head touched an edge (defined as contact on a narrow section of track) behind its “chin,” the head was commanded to push down until the segment was slightly beyond

horizontal (i.e., with “chin” down). The main control strategy for body movements attempted to keep the bottom of each segment as horizontal as possible whenever its bottom track touched an edge by rotating the joints ahead and behind appropriately in opposite directions. When an edge was sensed at the very front of a segment, the joint ahead of the edge was kept closed to keep the gap at the joint from getting hung up on the edge. If a joint received conflicting commands from the segment ahead of it and the segment behind it, the command from the segment behind the joint was executed. If joints #2 and #3 received commands from both the head (e.g., to lift for obstacles) and neighboring body segments (e.g., to lower a joint), the command from the head was executed. If a joint received no command, the default action was to reduce the current torque slightly. The initial test terrains stairs and parallel bars did not require lateral movements, so the control system simply kept the joints straight laterally.

The sagittal control program enabled the simulated OT-4 to climb 20.3 cm high stairs and over a series of parallel bars spaced at random heights and gaps near the robot’s physical limits. The control developed in simulation was then tested on the real OT-4 in corresponding terrains in the laboratory. Figure 1 shows the OT-4 traversing the parallel bars. For the tests, the 7G control system used input from human observers to substitute for an array of IR sensors on the head and touch sensors along the bottom and sides of each segment that were modeled in simulation but not present on the real OT-4 [18]. The autonomous control system enabled the OT-4 to climb the stairs and parallel bars terrains in the laboratory test courses, moving at higher speeds than could be demonstrated with human operators.

Extending serpentine control to lateral movements

Although sagittal control was effective for unobstructed stairs and parallel bars, lateral movements are often necessary for mobility and stability, especially for climbing where there are few surfaces with low enough slope for traction. In the sagittal plane, gravity inescapably arranges pressure between the robot and objects beneath it, but in the lateral plane such force must usually be created by intelligent movements of the robot. The physics of force resolution that dominate sagittal plane movements for climbing can be extended in large part to lateral contact with objects, because the component of the force of gravity parallel to the surface resolves into forces normal to the lateral contact and parallel to it. The lateral friction combines with the friction on the bottom tracks to prevent slipping.

Implementing aim and propagate lateral control

As a first step in developing control of lateral movements, a version of the Aim and Propagate approach was implemented. The head joint was programmed to comply with horizontal joystick positions whenever the operator was also pressing an override button on the joystick. Unlike other approaches in which commanded joint positions for the head were propagated to all following joints, each joint was programmed to propagate the actual joint position of the previous joint when it reached the corresponding location in the terrain. This approach was much better at accommodating cumulative unplanned changes in position (e.g., from slipping, rotating, or terrain deformation) as the robot moved through rough terrain. Position data obtainable in simulations permits perfect odometry for propagation of movements, but on the real OT-4 position was estimated by applying a calibration factor to the product of track speed time’s elapsed time. This simple programming enabled effective manual control of the robot through both simulated and real slalom courses, as well as remote teleoperation

in simulated and real courses where the operator could rely only on camera input from the front of the robot.

Combining lateral and sagittal control

The next step was to test a combination of sagittal control of the head and body joints with the Aim and Propagate lateral control. The real stairs terrain was modified by adding large irregularly shaped rocks along a winding path up a staircase, as shown in Figure 2 [18]. The operator’s joystick position determined the head orientation whenever the operator pressed the override button. This hybrid control enabled the real OT-4 to climb this challenging course, given that the joystick operator could directly observe the entire robot and terrain. The most obvious control weakness was that when the tracks slipped, the simple approximation to odometry became very inaccurate, causing lateral movements to propagate before reaching the corresponding location. This produced obviously suboptimal movements, but the robot could still move forward because the lateral movements pressed the side tracks against rocks to enable forward movement, although with unnecessary stress on joints and on the terrain. The operator learned through experience to add a certain amount of continuous lateral head oscillation in order to avoid persisting in stuck positions. It was also noted that propagating each joint position from the previous joint rather than from the head enabled recovery from odometry errors within one segment length of movement after slippage ended. While accurate odometry can potentially improve performance, subsequent testing in simulation with perfect odometry showed that propagation alone is insufficient for terrains such as this. However, propagation provided a valuable default behavior that could be overridden by more intelligent control developed in the next phase.

Developing intelligent lateral control

The previous tests showed that a relatively small set of general movement strategies was useful in a wide range of terrains. Much of their effectiveness is due to movements being controlled mostly on the basis of local sensory input, producing several “subgait” level behaviors along the length of the robot at the same time. The next phase of development added intelligent lateral control. A large and important class of terrains requires much more sophisticated lateral movements than the wiggling movements that had been successful in narrow passageways between rocks. Lateral objects have potential value for mobility and stability in many terrains, but they are essential for some terrains such as climbing slopes with few narrow passageways or horizontal affordances.



Figure 2: The Omni Tread OT-4 robot on stairs terrain with rocks, requiring control of sagittal and lateral movements.

There are two general classes of lateral movement strategies for climbing slopes. One is what was observed among the rocks on the stairs, which was to arch by bending multiple joints to press against objects on both sides of the path. The other class is necessary in terrains without objects on both sides, where the best movement is to move above and behind objects along the way to push against surfaces that produce upward force components. This phase of development focused on the latter class because a physics analysis suggests that it is a direct extension of the existing sagittal control. Moreover, it is consistent with the overall development strategy of focusing first on very local control of movements based on mostly local sensory input, as opposed to the more global arching behavior for narrow passages.

Terrain for developing lateral control

The terrain designed for this phase is a ramp, shown in Figure 3, that is too steep simply to climb in a straight line (i.e., the tangent of the slope is greater than the coefficient of static friction), but the slope contains objects analogous to rocks or posts along the path on which the robot can climb. As in the sagittal control problem, touch sensors provide simple and relatively unambiguous information about the location of objects, although they require intelligent control to maintain effective contact with those objects as well as some short-term memory to remember objects when contact is temporarily lost.

Method for developing intelligent lateral control

Following the same iterative development procedure used to develop sagittal control, effective lateral control was developed for this terrain. In this terrain, lateral body movements must use sensor input to relate effectively to the objects in order to climb successfully. The robot must approach nearby affordances, move behind or above them, turn back in the goal direction at appropriate times, and avoid getting hung up on edges between joint spaces on its sides.

The new lateral climbing behaviors were triggered only if the robot sensed that it was on a slope steep enough to risk slipping. As with sagittal collision avoidance, the head turns away to avoid hard collisions with objects. However, just after passing objects, the head turns toward the object to touch it along the side of the head. When on a slope, all segments turned toward objects touching their sides. Short-term memory based on lagged touch sensor input enabled segments to remember the location of objects in order to turn toward them when contact was temporarily lost.

As with the sagittal head movements, the segments behind the head were successively recruited for lateral movements as needed: if joint #1 is bending near its limit and segment #2 is not touching anything on the side opposite to joint #1's turning direction, joint #2 is commanded to turn in the same direction as joint #1, and so on through joint #5.



Figure 3: Terrain for developing intelligent lateral control of the OT-4 model in the Yobotics! Simulation Construction Set.

When no object was detected on either side or in front, the default head behavior was to turn toward the direction parallel to the slope of the ramp (a short-range goal that would be set by the operator in a fielded teleoperation application). The default behavior for the other joints was to propagate the angle from the previous joint at the same location.

A subtler behavior was to lift the head whenever the head was changing direction so as to avoid pivoting the body, but to push the head down with moderate force whenever the head was fixed in a sideways orientation in order to pull the body to follow the head. Because the robot can get hung up on a lateral edge just as on a vertical edge, a corresponding joint-closing behavior was implemented. Explicit sensor preprocessing inferred that a joint was hung up when the current and all lagged front sensors were touching only at the front of the adjoining segment, implicitly indicating no movement. Segments #5 and #6 just in front of the tail had additional behaviors that overrode the control scripts that otherwise were the same for all body joints, as follows. In addition to segments #5 and #6 commanding the joint behind them to turn toward whichever side they touch, those two rear joints are triggered by touch on the segment behind them to turn away from, rather than toward, the touch unless the contact is at the front (in which opening the joint risks hanging up).

The developers found that developing lateral control was significantly more difficult than developing sagittal control due to:

- Gravity not automatically maintaining contact with terrain affordances in this plane,
- Conflicts between movements for mobility and movements to maintain touch,
- The need to remember the location of objects while not contacting them,
- The need to learn sensory patterns to use as cues to approach objects, and
- The need to coordinate head lifting and downward pushing with lateral head and body movements to make turning effective (see below).

Results

The new control system developed in this phase retained the previously developed effective sagittal head and body control, retained control by propagation developed previously as the default lateral behavior, and added intelligent lateral movements as overrides in appropriate situations. Figure 4 shows a sequence of snapshots from a video made in the Yobotics simulator showing the simulated OT-4 robot climbing the ramp by pressing against the available posts and rocks in the simulated terrain¹. In Figure 4, panels h, i, and j, the rear segments of the robot can be seen maneuvering laterally to climb using a post, while the middle segments are moving sagittally to pull up over the ledge, and the head is turning sideways to avoid collision with the wall. All these movements must be executed correctly and simultaneously to succeed in climbing this terrain. The subgait control strategy succeeded in this case because each movement is in response to local terrain characteristics and movements can be executed independently of each other.

The combination of relatively simple behaviors often produces oscillation, such as turning toward and away from the objects.

¹The complete simulation video is available at: [http://www.behaviorsystems.com/files/ramp with posts and ledge.mov](http://www.behaviorsystems.com/files/ramp%20with%20posts%20and%20ledge.mov)

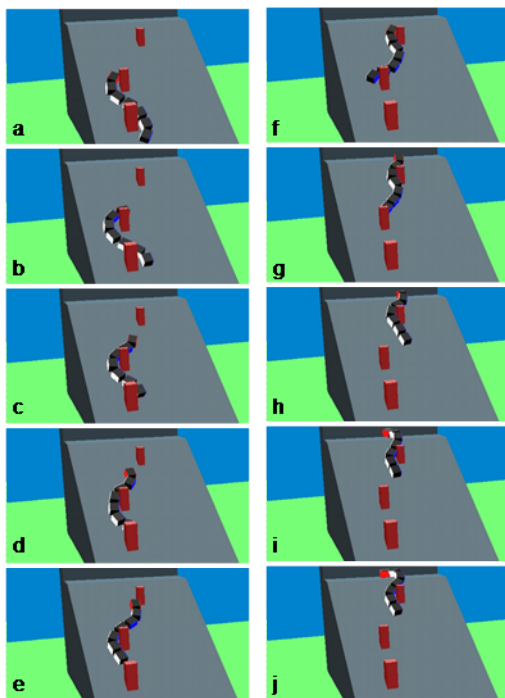


Figure 4: Snapshots from simulation of OT-4 robot climbing a steep ramp that required intelligent lateral movements to use posts to avoid slipping down the ramp, along with sagittal movements to pull itself over the ledge and lateral head movements to avoid hitting the wall.

Oscillation-reducing alternatives were tested, but the simpler control based on seeking and avoiding proved more robust across the variety of situations encountered. A number of observers of the simulations have commented that the behaviors, including the natural oscillation, give the impression of being more like an animal “trying” to climb the terrain than a programmed mechanical motion.

This last study was done in simulation and has not yet been validated by controlling the real OT-4 robot climbing an equivalent ramp. Previous validation tests of control of the real OT-4, however, have shown that 7G control developed in simulation transferred well to controlling the real OT-4 robot in similar terrains in the laboratory [18].

References

1. Hirose S (1993) *Biologically inspired robots: Snake-like locomotors and manipulators*. Oxford: Oxford University Press.
2. Nilsson M (1997) Snake robot free climbing in Proc. International Conf. on Robotics and Automation (ICRA), Albuquerque, NM.
3. Ikeda H, Takanashi N (1987) Joint assembly moveable like a human arm. US Patent 4683406 Assignee: NEC Corp.
4. Takayama T, Hirose S (2000) Development of Souryu-I connected crawler vehicle for inspection of narrow and winding space. 26th Annual Conf. of the IEEE Ind. Electronics Society (IECON), Nagoya, Aichi, Japan.
5. Brown H, Weghe J, Bererton C, Khosla P (2002) Millibot trains for enhanced mobility. *IEEE/ASME Trans. Mechatronics* 7: 452-461.
6. Osuka K, Kitajima H (2003) Development of mobile inspection robot for rescues activities: MOIRA. *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Las Vegas, NV.
7. Grzegorz G, Malik GH, Borenstein J (2005) The OmniTread serpentine robot for industrial inspection and surveillance. *Industrial Robot: An International Journal* 32: 139-148.
8. Borenstein J, Granosik G, Hansen M (2005) The Omni Tread serpentine robot design and field performance. *SPIE Defense and Security Conference: Unmanned Ground Vehicle Technology VII*, Orlando, Florida.
9. Borenstein J, Hansen J, Nguyen H (2006) The Omni Tread OT-4 serpentine robot for emergencies and hazardous environments. 2006 International Joint Topical Meeting: “Sharing Solutions for Emergencies and Hazardous Environments”, Salt Lake City, Utah, USA.
10. Transteth A, Leine R, Glocker C, Pettersen K, Liljeback P (2008) Snake Robot Obstacle-Aided Locomotion: Modeling, Simulations, and Experiments. *IEEE Trans. Robotics* 24: 88-104
11. Kamegawa T, Yamasaki T, Igarashi H, Matsuno F (2004) Development of the snake-like rescue robot KOHGA. *IEEE Intl. Conf. on Robotics and Automation*, New Orleans, LA, USA.
12. Klaassen B, Paap K (1999) GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control. *IEEE International Conference in Robotics and Automation*.
13. Dowling K (1997) *Limbless locomotion: Learning to crawl with a snake robot*. Ph.D. dissertation, Robotics Institute, Carnegie Mellon Univ of Pittsburgh, PA.
14. Ng AY, Coates A, Mark D, Varun G, Jamie S, et al. (2006) Autonomous helicopter flight via reinforcement learning. *Experimental Robotics IX* 21: 363-372.
15. Hutchison WR, The Seventh Generation (7G) System.
16. Hutchison WR, Constantine BJ, Borenstein J, Pratt J (2007) Developing control of a high-DOF robot using reinforcement learning, genetic algorithms, scripting, and simulation. *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*.
17. Yobotics! Simulation Construction Set.
18. Hutchison WR, Constantine BJ, Borenstein J, Pratt J (2007) Development of control for a serpentine robot. *IEEE Intl. Symposium on Computational Intelligence in Robotic Automation*.