Distributed Central Pattern Generator Control for a Serpentine Robot

Jörg Conradt [*]

Institute of Neuroinformatics
ETH / University Zurich
Winterthurerstrasse 190
CH - 8057 Zürich

conradt@ini.phys.ethz.ch

Paulina Varshavskaya

AI Laboratory
Massachusetts Institute of Technology
200 Technology Square
USA - Cambridge, MA 02139

paulina@ai.mit.edu

Abstract We have built a biologically and neurally inspired autonomous mobile robotic worm. The main aim of the project is to demonstrate elegant motion on a robot with a large number of degrees of freedom (DOFs) under the control of a simple distributed neural system as found in many animals' spinal cord. Our robot consists of individually controlled segments that exhibit Central-Pattern-Generator (CPG) -driven biomorphic motion. An important aspect of the project is to achieve a level of modularity while closely mimicking the neural control of e.g., the lamprey. This paper presents our robotic platform and the distributed CPG control algorithms. We will mainly focus on the architecture of the initial system and on future developments, and also report some preliminary experimental results.

Motivation

Among the vast body of work related to serpentine robots (see [3] for an extensive overview, and [4]-[6]), the issues of modularity and robustness as well as distributed biologically plausible control have seldom been addressed in a single study. The motivation behind our WormBot project [2] is to demonstrate elegant robust robotic motion based on simple, yet biologically plausible design principles in a high-DOF system. We investigate in the motion generated by multiple 1-DOF segments that are individually controlled by local CPGs, but achieve overall motion stability through short- and long-range couplings. A robotic platform to evaluate motion in such a system is not commercially available. Thus the focus of the project reported here is twofold: On the one hand, we developed the serpentine robot WormBot, which allows us to explore issues pertaining to the control of a high number of DOFs, modularity and inter-module communication. We aim for a simple, inexpensive and flexible design. On the other hand, we implement a biologically plausible and neurally inspired CPG control algorithm, which runs on physically distributed hardware.

Previous Work

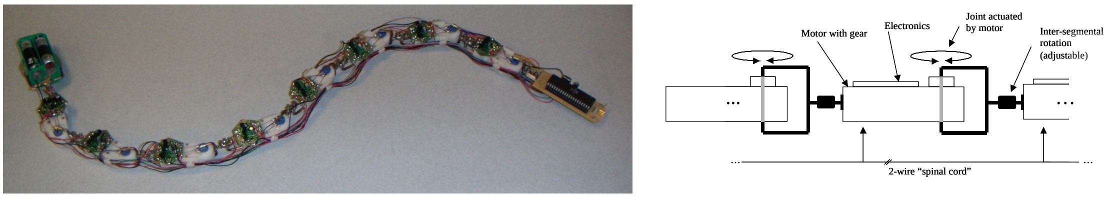
There is a large body of work on central pattern generators (CPGs), both in theoretical modeling of biological systems (e.g.,[1]) and in application to robotic joints. E.g. in [7], Williamson used coupled oscillators to control series elastic actuators for compliant robot arms. Some of the physical robots developed in previous studies present very elegant mechanical designs or life-like movement. As another example, the robotic snakes of [5] and [6] demonstrate ingenious modular designs. However, most such robots are not readily scalable to a large number of segments, and exhibit fixed patterns or central-driven motion. Also recently, researchers in reconfigurable robots have demonstrated modular distributed designs such as PolyBot [8], which in one configuration exhibits serpentine motion.

Our approach, however, is to place emphasis on the implementation of neurally inspired CPG-based control in a simple physical system that allows scalability and modularity. Dowling [3] did explore biologically inspired serpentine gaits, with impressive results in simulation but he only performed a few experiments on a real robot.

Robotic Platform

The WormBot has a segmented design (see Figures 1 and 2 for photographs of the two current prototypes). In both prototypes, all segments are identical, except the head and tail, which provide additional functionality.

[*] Corresponding author



(a) **(b)**

Figure 1. **(a)** Version 1 of the WormBot. Virtual distributed control from the microcontroller in the robot’s head. **(b)** Mechanical design of the WormBot. Segments can be mounted at any angle, allowing for motion in 3D.

The first design, as shown in Figure 1, consists of eight segments, each comprising a motor with gearbox, a potentiometer for position readings, and a motor-driving circuit. The extra tail segment provides power to all segments and the head from 4 AA batteries. The head segment provides the ‘brain power’ to the robot from a single microcontroller Atmel Mega163, which runs eight CPGs - one for each segment of the worm. While this design does not demonstrate true distributed control, it allows easy reconfiguration and movement in three dimensions, as shown in Figure 1b. Each segment actuates (rotates) one neighbor. A single rotational connection offers an adjustable angle between two neighboring segments, allowing us to quickly adapt the robot's shape for planar motion (0 degree connections, as e.g., in the snake or lamprey), for motion in 3D (90 degree connections, as e.g. in the worm), or for any angle in-between.

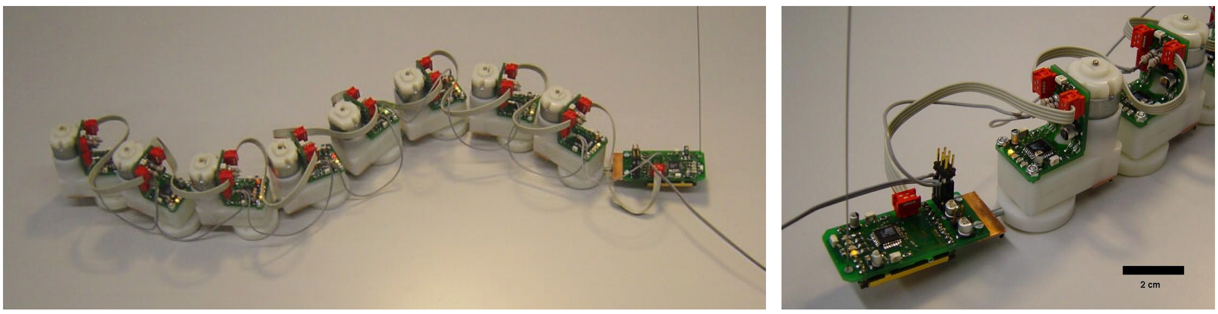
The second design (shown in Figure 2) achieves a much greater degree of modularity by providing each segment with its own re-programmable microcontroller Atmel Mega8, additional sensors, and a communications interface. Each segment’s microcontroller runs a local individual CPG, biased by current position and torque stimuli and actuates the corresponding motor using PWM signals. The sensors available on the second prototype robot are three light-sensors in orthogonal directions, a temperature sensor and sensors for the segment’s internal states (rotary position, applied motor torque, available voltage of power supply battery). A two-wire communication interface connecting all segments allows fast and flexible information exchange within the robot. In the current setup, segments communicate all sensor readings and internal states to all other segments, such that individual short- and long-range coupling between segments can be adjusted in software. The software coupling allows flexible adaptation during operation, e.g., for changing gait or direction of motion. The head segment in the second prototype is also connected to the communication bus, and exchanges data with a PC over a wireless connection. Thus, users can interface to the robot at runtime to adjust CPG parameters (e.g. coupling strengths, motion amplitude and phase-shifts) during otherwise autonomous operation.

Distributed Control

Version 2 of the WormBot exhibits true distributed control, with each module being driven by its own microcontroller. The motors are actuated by CPG oscillators coupled through the following relationship, adapted from [1]:

$$\dot{q}_i = w_i + \sum_{j=1}^{N} a_{ij} \sin\left(q_j - q_i - (i-j)\cdot f\right) \qquad (1)$$

where $?_i$ is the state of the *i*'th oscillator on its limit cycle, $?_i$ is the frequency of the *i*'th oscillator when not coupled to any other, a_{ij} the coupling strength from the *j*'th to the *i*'th oscillator, and N is the number of oscillators in the system. f denotes the desired phase shift between neighboring segments and $(i-j)$ the spatial distance between segments *i* and *j*.



(a) **(b)**

Figure 2. **(a)** Version 2 of the WormBot. Modules are planar but exhibit true distributed control with an individual microcontroller on each segment. **(b)** Close-up view of the head and the first segment. The black bar in the image corresponds to 2cm.

In the current implementation, we only consider near neighbor coupling between oscillators, even though it was shown that this is a great oversimplification of the couplings in the control of living systems. In the current prototype

$$a_{ij} = a \quad \text{for all } i, j \text{ such that } |j - i| = 1 \text{ and}$$
$$a_{ij} = 0 \quad \text{otherwise.}$$

We are working on replacing this arrangement with different arbitrarily complex re-configurable coupling functions, such as e.g., suggested by Verschure and Cohen (personal communication). In the updated CPG control, sensor readings (as e.g. motor-torques, light, temperature, etc.) will also have a direct influence on the CPGs, such that the robot can achieve simple behavior (e.g., obstacle avoidance or light following).

Experiments in Motion

With the current simple CPG algorithms we are able to achieve highly accurate coupled motion (rotation) in both our prototypes. Figure 3 demonstrates that the robot achieves stable planar traveling waves using coupling, with its head and tail held fixed on the table.

The robot is not moving forward when releasing its head and tail, but it is slowly sliding sideways on a flat surface. If the robot had a shell with differential friction on the bottom, the generated sideway forces would be transferred into forward motion. We are currently designing an appropriate shell to allow forward travel in the planar configuration.

All experiments described below are performed after turning the robot 90 degrees sideways, such that the traveling wave is expanding vertically and the robot lifts several of its segments off the surface.

In our experiments we have mainly concentrated on adjusting parameters of all the segments' CPGs simultaneously. Initially, we varied the desired phase shift f (equ. 1) between consecutive segments from 0 (no phase shift) to 0.5 (half a cycle phase shift). For values close to 0, all segments bend in the same direction at any time. Hence, the robot only twitches on the surface. A phase shift of $^1/_8$ corresponds to a traveling wave with one cycle along the robot's body in the eight-segment worm. In this configuration the robot moves rapidly forward. However, during the motion cycle its head and tail are lifted simultaneously, and only the middle segment touches ground. This is a physically unstable situation, in which the robot might fall sideways. Slightly increasing f to about $^{1.5}/_8$ results in 1.5 waves along the body, which solves this problem. Then, at least two physically separated segments touch the ground at any time. There is clearly a tradeoff between stability and speed: increasing f still produces forward motion but decreases the speed of the robot. When f approaches 0.5, all neighboring segments swing in anti-phase. Then, the robot only slightly wiggles up and down without any forward motion. For $f > 0.5$ the wave's traveling direction is reversed, and the robot moves in the opposite direction.

Another parameter we varied is the traveling speed w of the CPGs. With increased w the waves speed up and thus the robot moves forward faster. Unfortunately, the speed reduction in the motor gearboxes limits the maximal available rotational speed, such that w cannot be increased beyond about 2 Hz. Also, the low frequency of coupling updates (about 10 Hz) limits w, as bigger w require more frequent updates (faster drift in phase).

In addition to exploring the physical motion of the snake we also varied the coupling strength a of the CPGs to evaluate stable coupling behavior. It turned out that reliable couplings are only achieved for very small values of a, or for $a \approx 1$. For very small values, synchronization is accomplished after a relatively long time. In contrast, for $a \approx 1$ every CPG corrects for all of the difference to its neighbor at every phase update, so that synchronization is achieved quickly. All tested intermediate coupling coefficients resulted in extremely unstable oscillating behavior.

One last set of experiments investigated the robustness of the system. We electrically disconnected an individual segment at various positions in the robot during motion, which initially did not cause any harm to the moving robot. After some time, however, the two groups of segments in front and behind of the disconnected segment lost their synchronization. Using nearest neighbor coupling only cannot propagate the segments phase

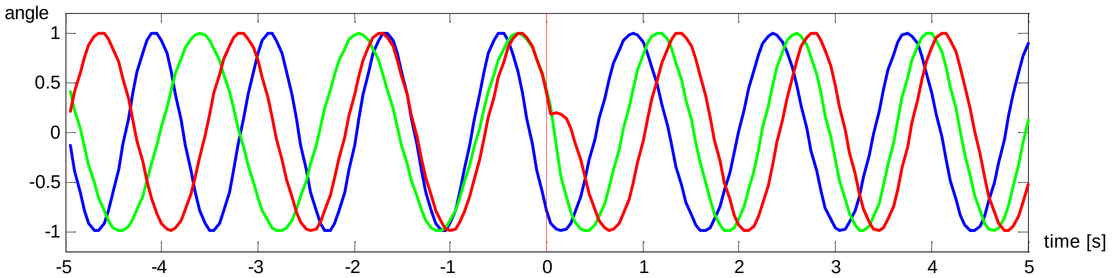


Figure 3. Recorded desired angular positions of the first three segments in the eight-segment worm robot. At time < 0 coupling between segments is deactivated and all oscillators swing with their individual resonance frequency. At time = 0 (vertical line) next-neighbor coupling is activated and the three CPGs synchronize

information across a disabled segment, making the current system susceptible to losing synchronization.

Videos of most of the above-mentioned experiments are available on our web pages[♣].

Discussion

We mentioned in the beginning of the previous section that differential friction on the robot's lower shell is required for planar serpentine motion. However, we expect that a robot composed of alternating horizontally and vertically actuated segments can lift some segments slightly and thus increasing the pressure on the ground at other segments. Exploiting this difference in pressure will allow the robot to travel forwards. We plan to explore these properties with a new robot composed of many more segments.

The CPG control seems to be highly reliable, despite the limitation of nearest neighbor coupling. We timed the operating frequencies of the microcontrollers, which are generated by an internal RC circuit. They differ by about 15%, such that without coupling but with identical w in all segments their speed will also differ by about 15%. Despite this significant difference, the independent CPGs typically synchronize quickly (fig. 3). Once a traveling wave has stabilized, it will not degrade except for artificially varying the different w so large as to prevent the coupling system from compensating for the different speeds.

We additionally observed that particular combinations of w, f, and motion amplitude provide maximally fast forward motion of the robot. Further increasing individual parameters will not result in greater speed. Increasing e.g., w will cause the CPGs to oscillate so fast that the motors cannot reach the desired angle anymore, and thus reduce the overall motion. It is yet to be determined how to optimize these parameters.

Future Work

We have presented here only the earliest stages of a research project that opens up many directions for future work. In the most immediate future we will continue to refine the mechanical hardware design of the WormBot and add individual batteries to every segment. We then plan to manufacture about 100 segments. Eventually, the robot may benefit from adding a small wireless video camera on the head.

A more important step for immediate development is implementing more advanced CPG algorithms on the individual segments (e.g. as suggested by Cohen and Verschure, personal communication). Experiments in directed motion in 2 and 3D and in aspects of reliability under extreme conditions such as failure and/or damage could then follow. Ultimately, simple behaviors (e.g. obstacle avoidance, photo-taxis) will be implemented on the robot.

In addition, we are hoping that the WormBot design could provide an interesting platform for neural systems scientists to test theories of neural control of motion. We do not suggest that the robot is a good approximation to a biological system. Rather, we hope that the distributed nature of the robot's control will allow for interesting experiments pertaining to the function of the spinal cord or other, possibly artificial, neural systems.

Conclusion

We have built a modular and robust robotic worm, which demonstrates truly distributed control. We believe that our system is inherently scalable to a system with many DOFs. Preliminary experimentation suggests that we can indeed generate robotic motion with an implementation of a simple model of neural control proposed for the spinal cord of a lamprey.

Acknowledgements

The authors wish to especially thank Avis Cohen and also other organizers and staff from the 2002 Telluride Workshop on Neuromorphic Engineering for invaluable insights at the origin of this project. The presented research is partially supported by a grant from the Institute for Neuromorphic Engineering's RCN program.

References

[1] Cohen, A.H., Holmes, P.J., Rand, R.H. The Nature of the Couplings Between Segmental Oscillators of the Lamprey Spinal Generator for Locomotion: A Mathematical Model, in *Journal of Mathematical Biology*, 1982 Vol.13, pp.345-369.

[2] Cohen, A.H., Shamma, S., Indiveri, G., Horiuchi, T., with Douglas, R., Koch, C. and Sejnowski, T. *Report on the 2002 Workshop on Neuromorphic Engineering*, pp.34-36

[3] Dowling, K.J. *Limbless Locomotion: Learning to Crawl with a Snake Robot*, 1997 PhD Thesis, CMU

[4] Henning, Hicknan and Choset, Motion planning for Serpentine Robots, *ASCE Space and Robotics*, Albuquerque, New Mexico 1998.

[5] Kimura, H., Hirose, S. Development of Genbu: Active wheel passive joint articulated mobile robot, *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, October 2002.

[6] Mori, M. and Hirose, S. Three-dimensional serpentine motion and lateral rolling by Active Cord Mechanism ACM-R3, *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, October 2002.

[7] Williamson, M. Neural Control of Rhythmic Arm Movements, *Neural Networks Special Issue on Neural Control of Movement*, 1998

[8] Yim, M., Duff, D.G., and Roufas, K.D., PolyBot: a Modular Reconfigurable Robot, *Proceedings of the 2000 IEEE Intl. Conference on Robotics and Automation*, San Francisco, CA, April 2000.

♣ http://www.ini.ethz.ch/~conradt/projects/WormBot