Evolving Motion of Robots with Muscles

Siavash Haroun Mahdavi and Peter J. Bentley

Department of Computer Science
University College London
Gower St. London WC1E 6BT, UK
{mahdavi, p.bentley}@cs.ucl.ac.uk

Abstract. The objective of this work is to investigate how effective smart materials are for generating the motion of a robot. Because of the unique method of locomotion, an evolutionary algorithm is used to evolve the best combination of smart wire activations to move most efficiently. For this purpose, a robot snake was built that uses Nitinol wire as muscles in order to move. The most successful method of locomotion that was evolved, closely resembled the undulating motion of the cobra snake. During experimentation, one of the four Nitinol wires snapped, and the algorithm then enabled adaptive behaviour by the robot by evolving another sequence of muscle activations that more closely resembled the undulations exhibited by the earthworm.

1 Introduction

We have grown used to software that crashes several times a day, faulty hardware that results in data loss and Sojourner Mars Rovers that get stuck on rocks. The problems are rare because our technology is very carefully designed. But the problems are significant and often devastating because our technology is not adaptive – it does not have graceful degradation. When something goes wrong, a terminal failure is common. In contrast, natural systems are able to work in many different environments, under very different conditions. Living designs are able to continue working even when damaged. The research described here is inspired by the adaptability of evolved solutions in nature.

The aims of this project are to create a self-adapting snake (SAS). The SAS uses muscles in order to move, made from a smart material call Nitinol. The sequence of muscle activations was controlled by a finite state machine, evolved using a genetic algorithm to maximise the distance travelled by each individual. Technology like this may be used to make devices that need to change their morphology to suit their environments, like a chameleon changing colour to avoid detection. Applications for this type of self-adapting form are obviously vast and so this project seeks to investigate whether a simple 'snake' made out of foam and wood could learn to morph its body in such a way as to move efficiently.

2 Background

2.1 Evolutionary Robotics

The field of adaptive robotics is not a new one. Many robots that use standard methods of movement like motors and pistons have been evolved from the design stage and all the way through to the control stage [1]. For a thorough survey, readers are advised to read [5] and [10]. Further examples include Jakobi, who endeavoured to build a simulation within which control systems for real robots could evolve [7]. Hornby evolved L-Systems to generate complicated multi-component virtual creatures [4]. Husbands used evolutionary robotics techniques to develop control networks and visual morphologies to enable a robot to achieve target discrimination tasks under very noisy lighting conditions [6]. Mihalachi and Munerato created a robot snake able to move like an earthworm [11][12].

The use of smart materials in robotics has already been investigated. For example, Kárník looked at the possible applications of walking robots, which use artificial muscles with smart materials as a drive [8]. Mills has written a book aimed at beginners which teaches them how to make simple eight-legged robots using smart materials as actuators [13]. However, no one has used smart materials and evolutionary algorithms together in a robot before. This work uses nitinol wires as muscles within a robot snake, with the muscle activation sequences evolved using genetic algorithms and finite state machines.

2.2 Smart Material

Nitinol, an alloy made of **Ni**ckel and **Ti**tanium, was developed by the **N**aval **O**rdinance **L**aboratory. When current runs through it, thus heating it to its activation temperature, it changes shape to the shape that it has been 'trained' to remember. The wires used in this project simply reduce in length, (conserving their volume and thus getting thicker), by about 5-8 %.

Shape Memory Alloys, when cooled from the stronger, high temperature form (Austenite), undergo a phase transformation in their crystal structure to the weaker, low temperature form (Martensite). This phase transformation allows these alloys to be super elastic and have shape memory [3].

The phase transformation occurs over a narrow range of temperatures, although the beginning and end of the transformation actually spread over a much larger range of temperatures. Hysteresis occurs, as the temperature curves do not overlap during heating and cooling, see Fig. 1 [3]. With thick wires, this could bring about problems for the SAS as the NiTi wires would take some time before returning to their original lengths, however, due to the very small diameter of the NiTi wires used (~0.15mm), the hysteresis was almost negligible as the cooling to below the Martensite temperature, (Mf), was almost instantaneous [2].

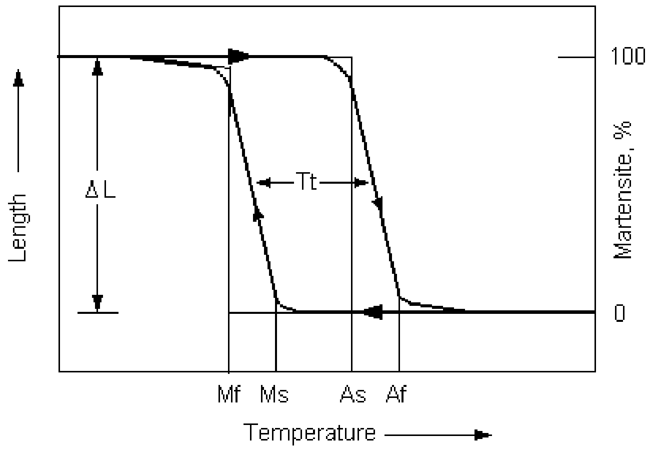

Fig. 1. Graph showing change in length during heating and cooling. The hysteresis is represented by Tt.

2.3 Snake motion in nature

This section reviews four different methods of undulation by which real snakes move [12]. An analysis of the most efficient muscle activation sequences is located in the analysis section at the end of the paper.

1. The grass snake undulates in horizontal plane. A horizontal S shape is created in its tail, this then moves along the entire length of the snake's body until the S finally reaches the head. The disadvantage with this motion is that there is a lot of rubbing along the length of the S, which consumes a lot of energy. The advantage with this form of undulation is that the snake is very stable whilst in motion.
2. The adder undulates in a vertical plane. The motion is very similar to that of the grass snake, except that the horizontal S is now replaced with a vertical U which moves from tail to head. The advantages with this motion are medium energy consumption and minimal rubbing. The disadvantage with this motion is with the instability of the snake, which is due to the vertical undulations.
3. The cobra undulates and extends in a horizontal plane. The S starts in the middle of the snake's body, it then repeats in both directions until it covers the entire body. The reverse is then performed until the cobra is once again straight. The advantages are that there is no problem with stability and that the speed is raised. The disadvantages are the high-energy consumption and a lot of rubbing.
4. The earthworm extends in a horizontal plane. This extension is parallel to the body and moves along its length. Unlike the grass snake and adder, the extensions travel from its head to its tail. The advantages are that there is no problem of stability, reduced energy consumption and rubbing is negligible. The disadvantages are that module elongation is necessary.

3 Building the Self-adapting Snake (SAS)

3.1 Body

The SAS body went through several designs before reaching the final design. The main criterion was that it provided a restoring force great enough to restore the wires to their original lengths after each activation. Foam was chosen for the job for all but one of the snake designs where plastic wire insulators were used [2].

Prototype snakes were made from foam with string replacing the NiTi wires. This was done in order to experiment with different skeletal designs. The strings were pulled in such a way as to decrease their lengths by around 5% (thus mimicking the NiTi wires). The main consideration when choosing a snake design was to see by how much the shape of the snakes deformed when the strings were pulled [2].

The SAS used in the experiments used four NiTi wires (diameter=1.5mm, activation (Austenite) temperature=70^{o}C, recommended current 200mA), that were connected all together at one side with a nut and bolt, which in turn was connected to a piece of normal wire. This wire ran through the middle of the snake and supplied the power, much like a spinal chord carries nerves impulses to muscles through the body, see Fig. 2. The total weight of the SAS was approximately 50g.

Fig. 2. Four wires were arranged at the top, bottom, left and right of the SAS.

3.2 Brain

A finite state machine was used to determine the wire activation sequences; these were then evolved using a standard single-point crossover genetic algorithm with roulette wheel selection.

Each member of the population was made up of a sting of 0s and 1s. Each string consisted of two segments, ‘sequence’ and ‘next time’, for a simple example see Fig. 3. The ‘sequence’ was the part that was sent to the SAS, and determined which wires were to be activated at that particular time. The ‘next time’ was the part that told the program to which time slot in the current row it should then jump. The recommended technique for activating the NiTi wires is to pulse them. This is because prolonged heating causes ‘hotspots’ to occur and these damage the wire and reduce its life span. Therefore, each sequence is in fact pulsed 7 times before moving on to the next sequence [2].

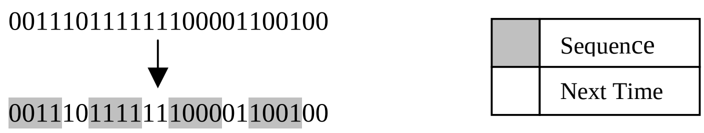

Fig. 3. Example string using 24 instead of 640 bits. Binary string is split into ‘sequence’ and ‘next time’ segments.

For the SAS, the ‘sequence’ length was four bits, and the ‘next time’ length was six bits, therefore the total length of each string was 2^6 x (4 + 6) = 640 bits.
The program was allowed to send 20 sequences to the SAS before stopping, lasting a total of approximately 40 seconds. Finite state machines that acted as repeating patterns were easily created if the jump points pointed to each other in a loop of some sort. Indeed, very rarely was there a string that didn’t loop at all.

The computer interfacing hardware, though quite complex, had two simple tasks to perform. The first was to supply enough power to each NiTi wire (~200mA). This was achieved with the use of some Darlington amplifiers. The second task to perform was the ability to activate muscles in parallel. For this, the microcontroller used was the Motorola MC68H(R)C908JK1. It had 20 pins and with the structure of the circuit board was capable of activating up to 12 pins in parallel (PTBx and PTDx excluding PTB0 and PTB3) [9].

4 Experimental Setup

The SAS was connected to the microcontroller circuit board using a long lead (~600mm) so that it did not have to pull much weight when trying to move. The SAS was placed on grid paper and a starting point was marked out. As the program started, an initial population of randomly generated finite state machines was created. Each individual finite state machine would then be sent to the SAS for evaluation.

The fitness of any member was determined solely on how far the snake travelled (in mm) in the forward direction [2].

The final configuration consisted of the SAS being placed on top of a platform about 500mm above the level of the circuit board and power supply. The wires were then draped over the edge of the platform. As the SAS moved forward, the wires were pushed down over the edge and so out of the way of the snake. There was concern over whether the weight of the wires was assisting in pulling the SAS along but much experimentation with the SAS ensured that the wires were pushed along and did not pull. The important factor was that the resistance of the wires was more or less constant no matter how far the SAS moved [2].

The distance travelled by the robot snake could be measured to the nearest millimetre. However, experimentation needed to be done to observe the true accuracy by which the distance travelled should be stated. To test this the SAS was sent the same sequence five times and the distances travelled were noted. The results show that the distance travelled was always within the nearest millimetre, and so the distances travelled by the SAS could be stated to the nearest millimetre without being over accurate, see table 1.

Table 1. Distance travelled when same sequence is sent to the SAS 5 times.

Test no.	Dist/mm
1	18
2	18
3	17
4	18
5	18

Following this, a series of experiments were performed to investigate the evolution of movement for the SAS. The SAS's motion was evolved for 25 generations and the corresponding fitnesses were stored.

5 Results

Because of the length of time required to perform experiments (approx 20 minutes per generation), two runs were executed. The results for the first run are given in Fig 4. As can be seen in the graph below, the distance travelled did not improved at all. In fact there seems to be a downward trend in fitness.

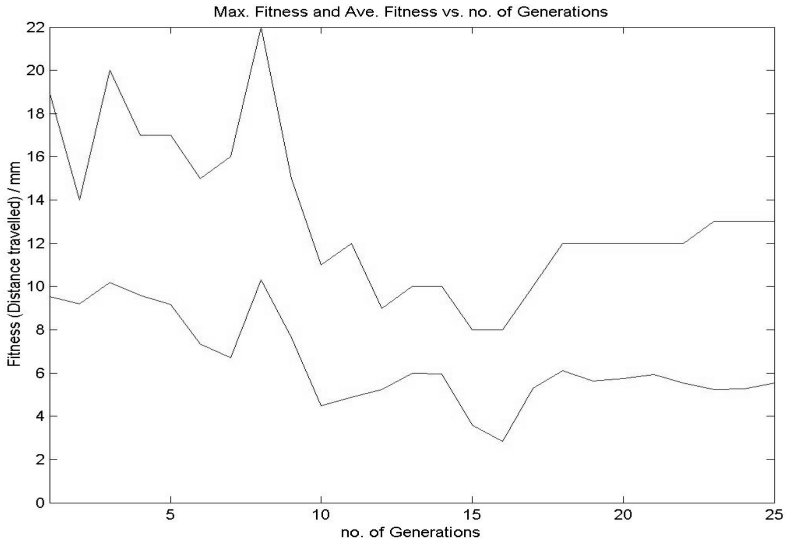

Fig. 4. Maximum fitness and average fitness plotted at each generation.

As each member of the population represents a very complicated FSM, finding suitable cut off points that would improve the fitness of the member without completely changing the pattern of sequence activations is very difficult. Such members, once mutated, induce motion that is nothing like the motion of the original parent, and so the fitness maybe considerably lower. Therefore, an individual that travels a particularly long distance has no guarantee of creating offspring that travel anywhere as far. This is likely to be the main cause of the decrease in overall fitness of the population. To overcome this problem, an elitist genetic algorithm was used for the second run.

This GA works simply by taking the n (n = 2 for the experiment) best members of the population and in addition to using them to generate members of the next generation, they are also placed directly into the next generation. In doing such, they naturally ensure that the maximum distance travelled during the coming generations never decreases, and they are also given another chance influence the next population if they still have the highest fitness. With this change, the maximum fitness never decreased (nearly always increasing), see Fig. 5.

Ideally the second experiment would have been carried out for as many generations as the previous experiment had but it had to be temporarily halted after only 8 generations. This was because one of the NiTi wires snapped!

The NiTi wire snapped at exactly the corner of the hole that it was fed through. It is believed that it snapped due to the sharp corner that it rubbed against and not because of overheating [2]. This would normally spell the end for any other robot, however one of the hypotheses of this research is that such techniques enable self-adaptation through evolution and so it seems only logical that if the SAS was truly self-adapting that it should be able to adapt to the loss of a muscle and learn other ways to move.

To test this, evolution was continued from the point at which the NiTi wire snapped (beginning of 8th generation). Every member of the population probably used the 4th wire to move at some point during its sequence and so the fitness of the whole populations dropped from an average distance of 13.25mm to an average of only 1.95mm. The members that did slightly better than the rest were probably members that had moved quite proficiently and had also used less of wire four. So these sorts of sequences were found more and more abundantly in the following generations. It took around 10 generations for any sequence to be found that moved more than 6mm. Then came a sequence that simply alternated between the top and bottom wire. This caused the SAS to move over 20mm. Since elite selection was being implemented, this sequence was not lost and variations spread throughout the proceeding generations increasing the fitness of the whole population considerably, see Fig. 5.

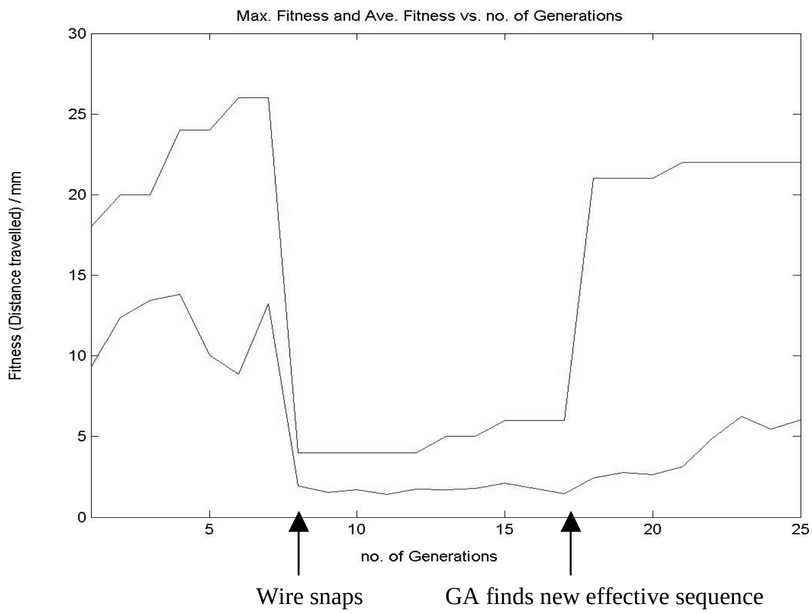

Fig. 5. Maximum fitness and average fitness plotted at each generation.

6 Analysis

During the evolution of the SAS, numerous interesting methods of locomotion were carried out. Though most of them were unsuccessful, this section seeks to analyse the physics behind two of the more efficient evolved sequences.

Though the sequence lengths could vary from a length of one to a maximum of sixty-four, the sequences that travelled the furthest seemed to have short repetitive loops.

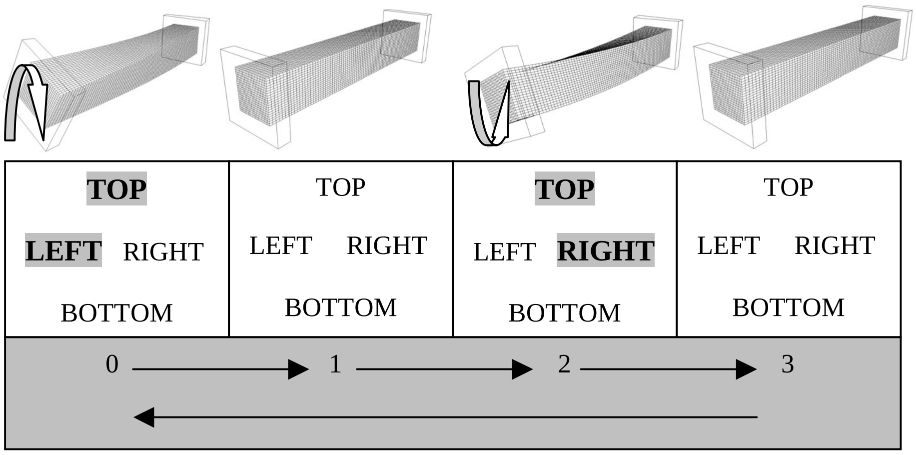

Fig. 6. Activation sequence of best sequence found resembles cobra undulations.

The best sequence found before the wire snapped could be compared to a simple but effective version of the cobra undulations, see Fig. 6. The SAS makes S shapes by alternating between activating the top and right wires, with the top and left wires. This S shape is along the whole length of the SAS and so provides the cobra-like undulations.

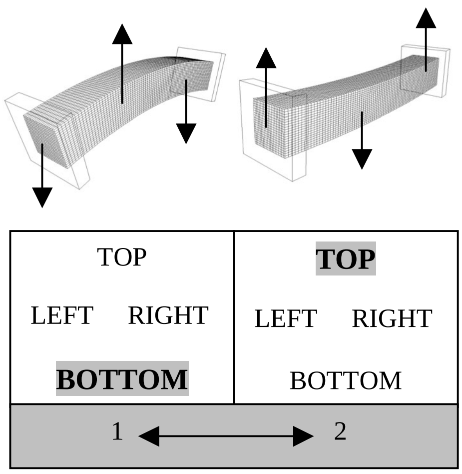

Fig. 7. Activation sequence of best sequence after damage resembles earthworm undulations.

After the NiTi wire snapped, the GA eventually evolved a sequence that simply alternated between the top and bottom wires. Alternating between compression and extension means that the new method of locomotion that had been evolved much resembled the undulations of the earthworm, see Fig. 7. Though this did not propagate the SAS as efficiently as the sequences that had been found previously, this pattern of muscle activations recovered over 85% of the previous mobility when it only had 75% of its muscle intact.

7 Conclusion

The aims of this research were to investigate whether shape memory alloys, in particular NiTi wires, could be used to make a robot snake that could move. This was done using a genetic algorithm, which determined their sequence of activations via the evolution of a finite state machine.

The research discovered that not only can these shape memory alloys induce motion, but that the type of undulation exhibited by the robot snake is adaptive. The type of movement that the robot snake demonstrated was very similar to those of the cobra snake. During experimentation, one of the four NiTi wires snapped, and by continuing to evolve the finite state machine that controlled which NiTi wires were activated, the robot snake modified its type of movement to one more closely resembling an earthworm.

References

1. Bentley, P. J. (Ed.) (1999) *EvolutionaryDesign by Computers*. Morgan Kaufmann Pub.
2. Haroun Mahdavi, S. (2002) Evolving Motion Master’s Dissertation, MSc IS, University College London, Dept. Computer Science.
3. Hodgson, Darel E. (2002), *Shape Memory Applications, Inc.*, Ming H. Wu, *Memory Technologies*, and Robert J. Biermann, *Harrison Alloys, Inc.* www.sma.com/Shape Memory Alloys.htm
4. Hornby, Gregory S. and Pollack, Jordan. B. (2001). Evolving L-Systems To Generate Virtual Creatures. Computers and Graphics, 25:6, pp 1041-1048.
5. Husbands, Phil and Meyer, Jean-Arcady (Eds.) (1998) Evolutionary Robotics, First European Workshop, EvoRbot98, Paris, France, April 16-17, 1998, Proceedings. Lecture Notes in Computer Science 1468 Springer 1998, ISBN 3-540-64957-3
6. Husband Phil., Smith T., Jakobi N. & O'Schea M. (1999). *Better living through chemistry: Evolving GasNets for robot control*. Connection Science, (10)3-4:185-210.
7. Jakobi, Nick. (1997), `*Evolutionary robotics and the radical envelope of noise hypothesis*', Journal Of Adaptive Behaviour 6(2), 325--368
8. Kárník, L. (1999) The possible use of artificial muscled in biorobotic mechanism. In ROBTEP'99, Kosice, SF TU Kosice, 1999, pp. 109-114. ISBN 80-7099-453-3.
9. MC68HC908JK1 Microcontroller User’s Manual (2002). Motorola Literature Distribution: P.O. Box 5405, Denver, Colorado 80217
10. Meyer Jean-Arcady, Husbands Phil and Harvey Inman (1998) Evolutionary Robotics: A Survey of Applications and Problems. EvoRobots 1998: 1-21
11. Mihalachi, D. and Munerato, F. (1999) Snake-like mobile micro-robot based on 3 DOF parallel mechanism. In Proc. of PKM'99 MILAN November 30, 1999.
12. Mihalachi, D. (2000) *Systeme de Commande Temps Reel Distribue pour un Mini Robot Autonome de type Serpent*. Doctorat de L’Universite de Metz, These Présentée à L’U.F.R. Mathematique, Informatique, Mecanique, Automatique.
13. Mills, Jonathan W. (1999) Stiquito for beginners. ISBN 0-8186-7514-4, 1999.