Evolution, Robustness, and Adaptation of Sidewinding Locomotion of Simulated Snake-Like Robot

Ivan Tanev [1], Thomas Ray[2,3], and Andrzej Buller[1]

[1] ATR Network Informatics Laboratories, 2-2-2 Hikaridai,
"Keihanna Science City", Kyoto 619-0288, Japan
[2] ATR Human Information Science Laboratories, 2-2-2 Hikaridai,
"Keihanna Science City", Kyoto 619-0288, Japan
[3] Department of Zoology, 730 Van Vleet Oval, Room 314,
University of Oklahoma Norman, Oklahoma 73019, USA
`{i_tanev,ray,buller}@atr.jp`

Abstract. Inspired by the efficient method of locomotion of the rattlesnake *Crotalus cerastes*, the objective of this work is automatic design through genetic programming, of the fastest possible (sidewinding) locomotion of simulated limbless, wheelless snake-like robot (Snakebot). The realism of simulation is ensured by employing the Open Dynamics Engine (ODE), which facilitates implementation of all physical forces, resulting from the actuators, joints constrains, frictions, gravity, and collisions. Empirically obtained results demonstrate the emergence of sidewinding locomotion from relatively simple motion patterns of morphological segments. Robustness of the sidewinding Snakebot, considered as ability to retain its velocity when situated in unanticipated environment, is illustrated by the ease with which Snakebot overcomes various types of obstacles such as a pile of or burial under boxes, rugged terrain and small walls. The ability of Snakebot to adapt to partial damage by gradually improving its velocity characteristics is discussed. Discovering compensatory locomotion traits, Snakebot recovers completely from single damage and recovers a major extent of its original velocity when more significant damage is inflicted. Contributing to the better understanding of sidewinding locomotion, this work could be considered as a step towards building real Snakebots, which are able to perform robustly in difficult environments.

Keywords: genetic programming, locomotion, snake-like robot

1 Introduction

Wheelless, limbless snake-like robots (Snakebots) feature potential robustness characteristics beyond the capabilities of most wheeled and legged vehicles – ability to traverse terrain that would pose problems for traditional wheeled or legged robots, and insignificant performance degradation when partial damage is inflicted. Some useful features of Snakebots include smaller size of the cross-sectional areas, stability, ability to operate in difficult terrain, good traction, high redundancy, and complete sealing of the internal mechanisms [2,3,12]. Robots with these properties open up several critical applications in exploration, reconnaissance, medicine and inspection. However, com-

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 627–639, 2004.

pared to the wheeled and legged vehicles, Snakebots feature (i) smaller payload, (ii) more difficult thermal control, (iii) more difficult control of locomotion gaits and (iv) inferior speed characteristics. Considering the first two drawbacks as beyond the scope of our work, and focusing on the drawbacks of control and speed, we intend to address the following challenge: how to develop control sequences of Snakebot's actuators, which allow for achieving the fastest possible speed of locomotion.

Although for many tasks, handcrafting the robot locomotion control code can be seen as a natural approach, it might not be feasible for developing the control code of Snakebot due to its morphological complexity. While the overall locomotion gait of Snakebot might emerge from relatively simply defined motion patterns of morphological segments of Snakebot, neither the degree of optimality of the developed code nor the way to incrementally improve the code is evident to the human designer [7]. Thus, an automated mechanism for solution evaluation and corresponding rules for incremental optimization of the intermediate solution(s) are needed [6]. The proposed approach of employing genetic programming (GP) implies that the code, which governs the locomotion of Snakebot is automatically designed by a computer system via simulated evolution through selection and survival of the fittest in a way similar to the evolution of species in the nature. The use of an automated process to design the control code opens the possibility of creating a solution that would be better than one designed by a human.

Evolving a Snakebot's locomotion (and in general, behavior of any robot) could be performed as a first step in the sequence of simulated off-line evolution (phylogenetic learning) on the software model, followed by on-line adaptation (ontogenetic learning) of evolved code on a physical robot situated in a real environment [8]. Off-line software simulation facilitates the process of Snakebot's controller design because the verification of behavior on physical Snakebot is extremely time consuming, costly and often dangerous for Snakebot and surrounding environment. Moreover, in some cases it is appropriate to initially model not only the locomotion, but also to co-evolve the most appropriate morphology of the artifact (i.e. number of phenotypic segments; types and parameters of joints which link segments; actuators' power; type, amount and location of sensors; etc.) [1,9,10] and only then (if appropriate) to physically implement it as hardware. The software model, used to simulate Snakebot should fulfill the basic requirements of being quickly developed, adequate, and fast running [4]. Typically slow development time of GP stems from the highly specific semantics of the main attributes of GP (e.g. representation, genetic operations, fitness evaluation) and can be significantly reduced through incorporating off-the-shelf software components and open standards in software engineering. To address this issue, we developed a GP framework based on open XML standard and ensure adequacy and runtime efficiency of Snakebot simulation, we applied the Open Dynamic Engine (ODE) freeware software library for simulation of rigid body dynamics.

The *objectives* of our work are (i) to explore the feasibility of applying GP for automatic design of the fastest possible locomotion of realistically simulated Snakebot and (ii) to investigate the robustness and adaptation of such locomotion to unanticipated environmental conditions and degraded abilities of Snakebot. Inspired by the fast sidewinding locomotion of the rattlesnake *Crotalus cerastes*, this work is motivated by our desires (i) to better understand the mechanisms underlying sidewinding locomotion of natural snakes, (ii) to explore the phenomenon of emergence of locomotion of complex bodies from simply defined motion patterns of the morphological segments comprising these bodies, (iii) to verify the feasibility of employing ODE for realistic software simulation of a Snakebot, and (iv) to investigate the practicality of building real Snakebots.

The remainder of this document is organized as follows. Section 2 emphasizes the main features of the GP proposed for evolution of locomotion of simulated Snakebot. Section 3 presents empirical results of evolving locomotion gaits of Snakebots and discusses the emergence of sidewinding. The same section elaborates on robustness and adaptation of sidewinding to unanticipated environmental conditions and partial damage of Snakebot. Finally, Section 4 draws a conclusion.

2 Approach

2.1 Representation of Snakebot

Snakebot is simulated as a set of identical spherical morphological segments ("vertebrae"), linked together via universal joints. All joints feature identical (finite) angle limits and each joint has two attached actuators ("muscles"). In the initial, standstill position of Snakebot the rotation axes of the actuators are oriented vertically (vertical actuator) and horizontally (horizontal actuator) and perform rotation of the joint in the horizontal and vertical planes respectively (Figure 1). Considering the representation of Snakebot, the task of designing the fastest locomotion can be rephrased as developing temporal patterns of desired turning angles of horizontal and vertical actuators of each segment, that result in fastest overall locomotion of Snakebot.

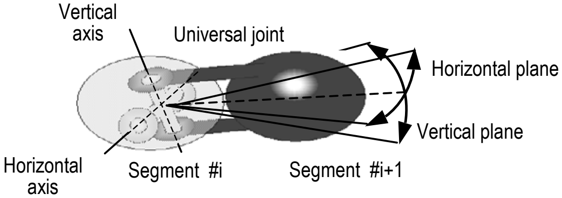


Fig. 1. Morphological segments of Snakebot linked via universal joint. Horizontal and vertical actuators attached to the joint perform rotation of the segment `#i+1` in vertical and horizontal planes respectively.

2.2 Algorithmic Paradigm

GP. GP [5] is a domain-independent problem-solving approach in which a population of computer programs (individuals' genotypes) is evolved to solve problems. The simulated evolution in GP is based on the Darwinian principle of reproduction and survival of the fittest. The fitness of each individual is based on the quality with which the phenotype of the simulated individual is performing in a given environment. The major attributes of GP - function set, terminal set, fitness evaluation, genetic representation, and genetic operations are elaborated in the remaining of this Section.

Function Set and Terminal Set. In applying GP to evolution of Snakebot, the genotype is associated with two algebraic expressions, which represent the temporal patterns of desired turning angles of both the horizontal and vertical actuators of each morphological segment. Since locomotion gaits are periodical, we include the trigonometric functions `sin` and `cos` in the GP function set in addition to the basic algebraic functions. The choice of these trigonometric functions reflects our intention to verify the hypothesis (first expressed by Petr Miturich in 1920's) that undulative motion mechanisms could yield efficient gaits of snake-like artifacts operating in air, land, or water. Terminal symbols include the variables `time`, `index` of morphological segment of Snakebot, and two constants: `Pi`, and `random` constant within the range [0, 2]. The main parameters of the GP are summarised in Table 1.

Table 1. Main parameters of GP

Category	Value
Function set	{sin, cos, +, -, *, /}
Terminal set	{time, segment_ID, Pi, random constant, ADF}
Population size	200 individuals
Selection	Binary tournament, ratio 0.1
Elitism	Best 4 individuals
Mutation	Random subtree mutation, ratio 0.01
Fitness	Velocity of simulated Snakebot during the trial
Trial interval	180 time steps, each time step account for 50ms of "real" time
Termination criterion	(Fitness >100) *or* (Generations>30) *or* (no improvement of fitness for 16 generations)

The rationale of employing automatically defined function (ADF) is based on empirical observation that the evolvability of straightforward, independent encoding of desired turning angles of both horizontal and vertical actuators is poor, although it allows GP to adequately explore the search space and ultimately, to discover the areas which correspond to fast locomotion gaits in solution space. We discovered that (i) the motion patterns of horizontal and vertical actuators of each segment in fast locomotion gaits are highly correlated (e.g. by frequency, direction, etc.) and that (ii) discovering and preserving such correlation by GP is associated with enormous computational effort. ADF, as a way of introducing modularity and reuse of code in GP [5] is employed in our approach to allow GP to explicitly evolve the correlation between motion patterns of horizontal and vertical actuators as shared fragments in algebraic expressions of desired turning angles of actuators. Moreover, the best result was obtained by (i) allowing the use of ADF as a terminal symbol in algebraic expression of desired turning angle of vertical actuator only, and (ii) by evaluating the value of ADF by equalizing it to the value of currently evaluated algebraic expression of desired turning angle of horizontal actuator.

Fitness Evaluation. The fitness function is based on the velocity of Snakebot, estimated from the distance which the center of the mass of Snakebot travels during the trial. The real values of the raw fitness, which are usually within the range (0, 2) are multiplied by a normalizing coefficient in order to deal with integer fitness values within the range (0, 200). A normalized fitness of 100 (one of the termination criteria shown in Table 1) is equivalent to a velocity which displaced Snakebot a distance equal to twice its length. The fitness evaluation routine is shown in Figure 2.

Representation of Genotype. Inspired by its flexibility, and the recently emerged widespread adoption of document object model (DOM) and extensible markup language (XML), we represent evolved genotypes of simulated Snakebot as DOM-parse trees featuring equivalent flat XML-text. Our approach implies that both (i) the calculation of the desired turning angles during fitness evaluation (functions `EvalHorizontalAngle` and `EvalVerticalAngle`, shown in Figure 2, lines 18 and 20 respectively) and (ii) the genetic operations are performed on DOM-parse trees using off-the shelf, platform- and language neutral DOM-parsers. The corresponding XML-text representation (rather than S-expression) is used as a flat file format, feasible for migration of genetic programs among the computational nodes in an eventual distributed implementation of the GP. The benefits of using DOM/XML-based representations of genetic programs are (i) fast prototyping of GP by using standard built-in API of DOM-parsers for traversing and manipulating genetic programs, (ii) generic support for the representation of grammar of strongly-typed GP using W3C-standardized XML-schema; and (iii) inherent Web-compliance of eventual parallel distributed implementation of GP.

The slight performance degradation in computing the desired turning angles of actuators by traversing the DOM/XML-based representation of genetic programs during fitness evaluation is not relevant for the overall performance of GP. The performance profiling results indicate that fitness evaluation routine consumes more than 99% of GP runtime, however, even for relatively complex genetic programs featuring a few hundred tree nodes, most of the fitness evaluation runtime at each time step is associated with the relatively enormous computational cost of the physics simulation (actuators, joint limits, friction, gravity, collisions, etc.) of phenotypic segments of the simulated Snakebot (routine `dWorldStep` in Figure 2, line 32), rather than computing the desired turning angles of actuators.

Genetic Operations. Binary tournament selection is employed – a robust, commonly used selection mechanism, which has proved to be efficient and simple to code. Crossover operation is defined in a strongly typed way in that only the DOM-nodes (and corresponding DOM-subtrees) of the same data type (i.e. labeled with the same tag) from parents can be swapped. The sub-tree mutation is allowed in strongly typed way in that a random node in genetic program is replaced by syntactically correct subtree. The mutation routine refers to the data type of currently altered node and applies randomly chosen rule from the set of applicable rewriting rules as defined in the grammar of strongly typed GP.

ODE. We have chosen Open Dynamics Engine (ODE) [11] to provide a realistic simulation of physics in applying forces to phenotypic segments of Snakebot, for simulation of Snakebot locomotion. ODE is a free, industrial quality software library for simulating articulated rigid body dynamics. It is fast, flexible and robust, and it has built-in collision detection. The ODE-related parameters of simulated Snakebot are summarized in Table 2.

```
function Evaluate(GenH, GenV: TGenotype): real;
// GenH and GenV is a pair of algebraic expressions, which define the
// turning angle of the horizontal and vertical actuators at the joints
// of simulated Snakebot. GenH and GenV represent the evolved genotype.
Const
 TimeSteps         =180; // duration of the trial
 SegmentsInSnakebot=15;  // # of phenotypic segments in simulated Snakebot
var
 t, s                  : integer;
 AngleH, AngleV        : real;    // desired turning angles of actuators
 CurrAngleH, CurrAngleV: real;    // current turning angles of actuators
 InitialPos, FinalPos  : 3DVector;// (X,Y,Z)
begin
 InitialPos:=GetPosOfCenterOfMassOfSnakebot;
 for t:=0 to TimeSteps-1 do begin
  for s:=0 to SegmentsInSnakebot-1  do begin
    // traversing XML/DOM-based GenH using DOM-parser:
    AngleH     := EvalHorizontalAngle(GenH,s,t);
    // traversing XML/DOM-based GenV using DOM-parser:
    AngleV     := EvalVerticalAngle(GenV,s,t);
    CurrAngleH := GetCurrentAngleH(s);
    CurrAngleV := GetCurrentAngleV(s);
    SetDesiredVelocityH(CurrAngleH-AngleH,s);
    SetDesiredVelocityV(CurrAngleV-AngleV,s);
   end;
  // detect collisions between the objects (phenotypic segments,
  // ground plane, etc.):
  dSpaceCollide;
  // Obtain new properties (position, orientation, velocity
  // vectors, etc.) of morphological segments of Snakebot as a result
  // of applying all forces:
  dWorldStep;
end;
FinalPos := GetPosOfCenterOfMassOfSnakebot;
return GetDistance(InitialPos, FinalPos)/(TimeSteps);
end;
```

Fig. 2. Fitness evaluation routine

3 Results

This section discusses empirical results verifying the feasibility of applying GP for evolution of the fastest possible locomotion gaits of Snakebot for various fitness and environmental conditions. In addition, it investigates the properties of the fastest locomotion gait, evolved in an unconstrained environment from two perspectives: (i) robustness to various unanticipated environmental conditions and (ii) gradual adaptation to degraded mechanical abilities of Snakebot. These challenges are considered as relevant for successful accomplishment of various practical tasks during anticipated exploration, reconnaissance, medicine and inspection missions.

Table 2. ODE-related parameters of simulated Snakebot

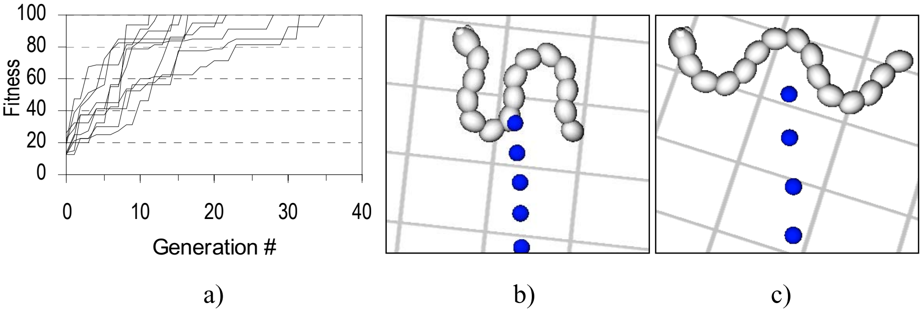
Parameter	Value
Number of phenotypic segments in snake	15
Model of segment	Sphere, R=0.2
Type of joint between segments	Universal
Initial alignment of segments in Snakebot	Along Y-axis of the world
Number of actuators per joint	2
Orientation of axes of actuators	Horizontal – along X-axis and Vertical – along Z-axis of the world
Operational mode of actuators	dAMotorEuler
Max force of actuators	12
Actuators stops (angular limits)	±50°
Friction between segments and surface (μ)	5
Sampling frequency of simulation	20 Hz

3.1 Evolution of Fastest Locomotion Gaits

Figure 3 shows the fitness convergence characteristics of 10 independent runs of GP (Figure 3a) and sample snapshots of evolved best-of-run locomotion gaits (Figure 3b and Figure 3c) when fitness is measured in *any* direction in an unconstrained environment. Despite the fact that fitness is unconstrained and measured as velocity in any direction, *sidewinding* locomotion (defined as locomotion predominantly perpendicular to the long axis of Snakebot) emerged in all 10 independent runs of GP, suggesting that it provides superior speed characteristics for Snakebot morphology. The dynamics of evolved turning angles of actuators in sidewinding locomotion result in characteristic circular motion pattern of segments around the center of the mass as shown in Figure 4a. The circular motion pattern of segments and the characteristic track on the ground as a series of diagonal lines (Figure 4b) suggest that during sidewinding the shape of Snakebot takes the form of a rolling helix. Figure 4 demonstrates that the simulated evolution of locomotion via GP is able to invent the improvised “wheel” of the sidewinding Snakebot to achieve fast locomotion.

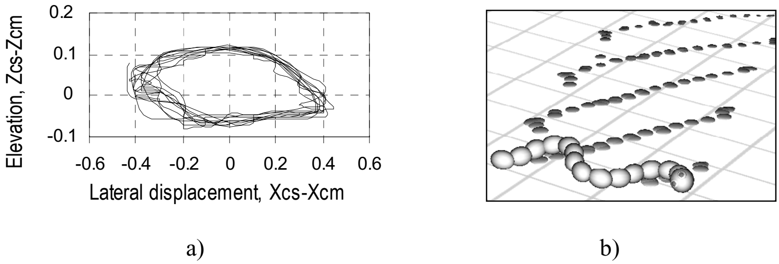
In order to verify the superiority of velocity characteristics of sidewinding locomotion for Snakebot morphology we compared the fitness convergence characteristics of evolution in unconstrained environment for the following two cases: (i) unconstrained fitness measured as velocity in any direction (as discussed above and illustrated in Figure 3a), and (ii) fitness, measured as velocity in forward (non-sidewinding) direction only. The results of evolution of forward locomotion, shown in Figure 5 indicate that non-sidewinding motion, compared to sidewinding, features much inferior velocity characteristics.

The results of evolution of rectilinear locomotion of simulated Snakebot confined in narrow "tunnel" are shown in Figure 6. The width of the tunnel is three times the diameter of the cross-section (which equals to the diameter of the segment) of Snakebot. Compared to forward locomotion in unconstrained environment (Figure 5), the velocity in this experiment is superior, and comparable to the velocity of sidewinding (Figure 3). This, seemingly anomalous phenomenon demonstrates the ability of simulated evolution to discover a way to utilize the walls of "tunnel" as a source of (i) extra grip and (ii) locomotion gaits which are fast yet unbalanced in an unconstrained environment. As Figure 6c illustrates, as soon as Snakebot clears the tunnel, the gait flattens and velocity (visually estimated as a distance between the traces of the center of gravity of Snakebot) drops dramatically.



a) b) c)

Fig. 3. Fitness convergence characteristics of 10 independent runs of GP for cases where fitness is measured as velocity in any direction (a) and snapshots of sample evolved best-of-run sidewinding locomotion gaits of simulated Snakebot (b, c), viewed from above. The dark trailing circles depict the trajectory of the center of the mass of Snakebot. Timestamp interval between each of these circles is fixed and it is the same (10 time steps) for both snapshots.



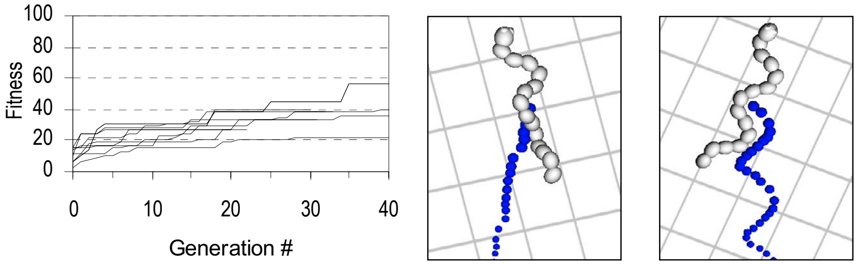
a) b)

Fig. 4. Trajectory of the central segment (cs) around the center of mass (cm) of Snakebot for a sample evolved best-of-run sidewinding locomotion (a) and traces of ground contacts (b).

The final experiment discussed in this section is intended to verify the ability of GP to evolve not only periodic locomotion gaits but also standstill postures, such as elevation of the head of Snakebot. The best-of-run postures (as shown in Figure 7) feature well-balanced, standstill elevation of the head. The elevation is approximately 3 diameters of Snakebot's segments, or about 20% of overall length of creature.

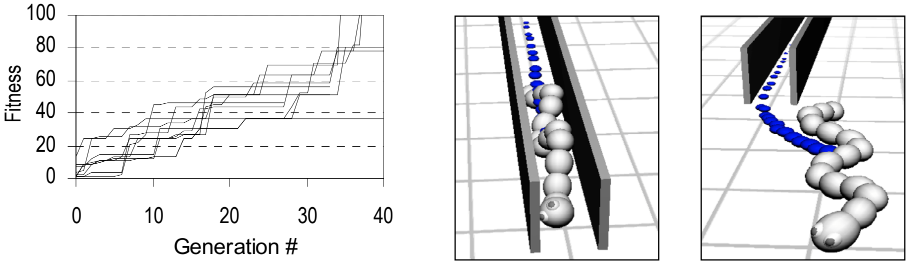
3.2 Robustness of Evolved Sidewinding Locomotion

Within the scope of our work we consider the robustness of sidewinding locomotion as the ability of the sidewinding Snakebot to retain its velocity when situated in a challenging environment. Robustness is qualitatively demonstrated by the ease with which the sidewinding Snakebot, initially evolved in unconstrained environment overcomes a pile of 80 boxes (Figure 8), burial under 80 boxes (Figure 9), rugged terrain with 200 randomly positioned obstacles with uniform random distribution of size in the range 0.1 to 1 of the diameter of the cross-section of Snakebot (Figure 10) and, finally, walls with height equal to the diameter of the cross-section of Snakebot (Figure 11). Snakebot, controlled by a sample of the 10 best-of-run genetic programs, obtained from experiments with evolving sidewinding in an unconstrained environment as elaborated in Section 3.1, overcomes the environmental challenges with an average of 60-80% of its original velocity.



a) b) c)

Fig. 5. Fitness convergence characteristics of 10 independent runs of GP for cases where fitness is measured as velocity in forward direction (a) and snapshots of sample evolved best-of-run forward crawling locomotion gaits of simulated Snakebot (b, c). Timestamp interval between the traces of the center of the mass is the same as for sidewinding locomotion gaits, shown in Figure 3b and 3c. The distance between the traces of center of the mass in both forward and sidewinding locomotion gaits comparatively illustrates the achieved velocity in both cases



a) b) c)

Fig. 6. Fitness convergence characteristics of 10 independent runs of GP when simulated Snakebot is confined in narrow “tunnel” (a) and snapshots of sample evolved best-of-run gaits at the intermediate (b) and final stages of the trial (c)

Fig. 7. Snapshots of sample evolved best-of-run standstill postures featuring elevated head of Snakebot: front view (left) and view from above (right).

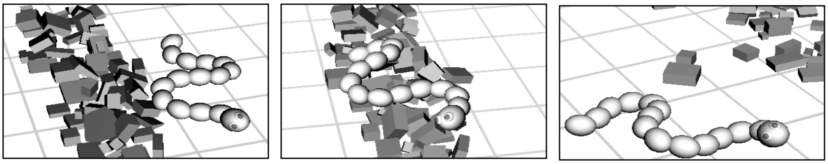


Fig. 8. Snapshots illustrating the robustness of sidewinding in clearing a pile of boxes: initial (left), intermediate (middle) and final (right) stages of the trial

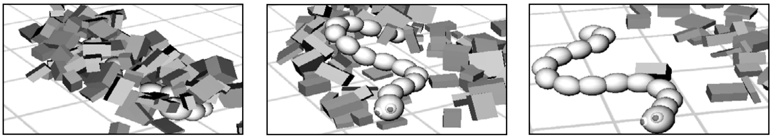


Fig. 9. Snapshots illustrating the robustness of sidewinding in emerging from burial under a stack of boxes: initial (left), intermediate (middle) and final (right) stages of the trial

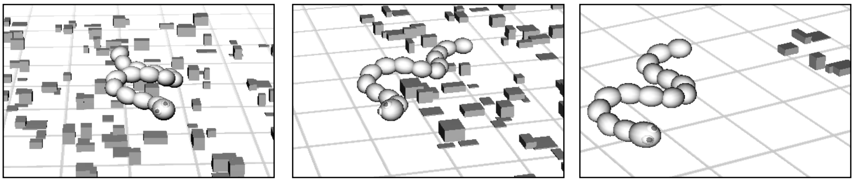


Fig. 10. Snapshots illustrating the robustness of sidewinding in rugged terrain area: initial (left), intermediate (middle) and final (right) stages of the trial

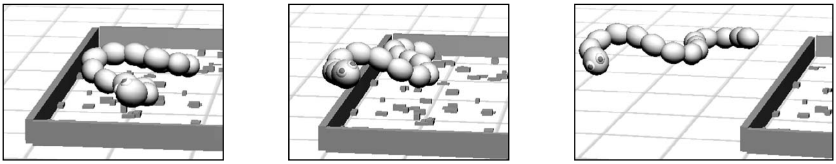
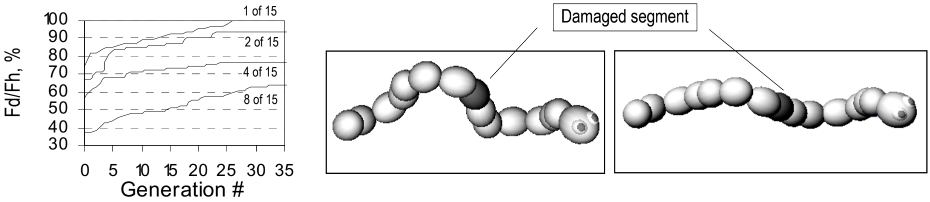


Fig. 11. Snapshots illustrating the ability of simulated sidewinding Snakebot in clearing walls forming a "pen": initial (left), intermediate (middle) and final (right) stages of the trial. Height of the walls is equal to the diameter of cross-section of simulated Snakebot.

3.3 Adaptation

The ability of sidewinding Snakebot to adapt to partial damage to 1, 2, 4 and 8 (out of 15) segments by gradually improving its velocity by simulated evolution via GP is shown in Figure 12. Demonstrated results are averaged over 4 independent runs for each case, where GP is initialized with a population comprising 190 randomly created individuals, plus 10 best-of-run genetic programs obtained from experiments with evolving sidewinding in an unconstrained environment as elaborated in Section 3.1. The damaged segments are evenly distributed along the body of Snakebot. Damage inflicted to a particular segment implies a complete loss of functionality of both horizontal and vertical actuators of the corresponding joint. As Figure 12a illustrates, Snakebot completely recovers from damage to single segment in 25 generations, attaining its previous velocity, and recovers to average of 94% of its previous velocity in the case where 2 (13% of total amount of 15) segments are damaged. With 4 (27%) and 8 (53%) damaged segments the degree of recovery is 77% (23% degradation) and 64% (36% degradation) respectively. Figure 12b shows a snapshot of frontal view of sidewinding Snakebot adapted to damage of a single segment. Compared to the sidewinding locomotion of Snakebot before the adaptation (Figure 12c), the adapted locomotion gait features much higher elevation of the middle part of the body. This elevation compensates the complete lack of functionality of actuators in the damaged segment.



a) b) c)

Fig. 12. Adaptation of sidewinding Snakebot to damage of 1, 2, 4 and 8 segments (a), snapshots of frontal view of sidewinding, adapted to damage of single segment (b) and sidewinding before the adaptation (c). Fd is the best fitness in evolved population of damaged snakebots, and Fh is the best fitness of 10 best-of-run healthy sidewinding Snakebots.

4 Conclusion

We presented an approach to automatic design through genetic programming, of sidewinding locomotion of simulated limbless, wheelless artifacts. The software model used to simulate Snakebot should fulfill the basic requirements of being quickly developed, adequate, and fast running. To address the first of these issues, we employed an XML-based GP framework. To address the issues of adequacy and runtime efficiency of Snakebot simulation we applied the Open Dynamic Engine (ODE) – a freeware software library for simulation of rigid body dynamics. The empirically obtained results demonstrate that the complex locomotion of sidewinding emerges from relatively simple motion patterns of phenotypic segments (vertebrae). The evolved locomotion pattern of each segment is such that the segment is rotating in a circle-like trajectory around the center of the mass of the simulated Snakebot. This suggests that evolved sidewinding locomotion can be viewed as a process of rolling of the body of the simulated Snakebot in a helix shape, effectively inventing a kind of improvised wheel. The efficiency of sidewinding locomotion is much superior to locomotion in the forward direction, suggesting that sidewinding is the fastest possible locomotion for the simulated limbless wheelless robots with the characteristics used in this study (morphology, limits of actuator forces, joint type, joint movement limits, etc.). Robustness of the sidewinding Snakebot, initially evolved in unconstrained environment (considered as ability to retain its velocity when situated in unanticipated environment) was illustrated by the ease with which Snakebot overcomes various types of obstacles such as piles of and burial under boxes, rugged terrain and walls. The ability of Snakebot to adapt to partial damage by gradually improving its velocity characteristics was discussed. Discovering compensatory locomotion traits, Snakebot recovers completely from single damage and recovers a major extent of its original velocity when more significant damage is inflicted. Contributing to the better understanding of sidewinding locomotion, this work could be considered as a step towards building real limbless, wheelless robots, which featuring unique engineering characteristics are able to perform robustly in difficult environments.

Acknowledments. The authors thank Katsunori Shimohara for his immense support of this research. The research was conducted as part of "Research on Human Communication" with funding from the Telecommunications Advancement Organization of Japan.

References

1. Bongard, J. C., R. Pfeifer: Evolving Complete Agents Using Artificial Ontogeny, in Hara, F. & R. Pfeifer, (eds.), Morpho-functional Machines: The New Species, Springer-Verlag, pp. 237-258 (2003)
2. Dowling, K.: Limbless Locomotion: Learning to Crawl with a Snake Robot, doctoral dissertation, tech. report CMU-RI-TR-97-48, Robotics Institute, Carnegie Mellon University (1997)
3. Hirose, S.: Biologically Inspired Robots: Snake-like Locomotors and Manipulators, Oxford University Press (1993)
4. Jacobi, N.: Minimal Simulations for Evolutionary Robotics. Ph.D. thesis, School of Cognitive and Computing Sciences, Sussex University (1998)
5. Koza, J.R.: Genetic Programming 2: Automatic Discovery of Reusable Programs, The MIT Press, Cambridge, MA (1994)
6. Mahdavi, S., Bentley, P.J.: Evolving Motion of Robots with Muscles. In Proc. of EvoROB2003, the 2nd European Workshop on Evolutionary Robotics, EuroGP 2003 (2003) 655-664
7. Morowitz, H.J.: The Emergence of Everything: How the World Became Complex, Oxford University Press, New York (2002)
8. Meeden, L., Kumar, D.: Trends in Evolutionary Robotics, Soft Computing for Intelligent Robotic Systems, edited by L.C. Jain and T. Fukuda, Physica-Verlag, New York, NY (1998) 215-233
9. Sims, K.: Evolving 3D Morphology and Behavior by Competition, Artificial Life IV Proceedings, MIT Press (1994) 28-39

10. Ray, T.: Aesthetically Evolved Virtual Pets, Leaonardo, Vol.34, No.4 (2001) 313 – 316
11. Smith, R.: Open Dynamics Engine (2001-2003) http://q12.org/ode/
12. Zhang, Y., Yim, M. H., Eldershaw, C., Duff, D. G., Roufas, K. D.: Phase automata: a programming model of locomotion gaits for scalable chain-type modular robots. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003); October 27 - 31; Las Vegas, NV (2003)