Emergent Motion Characteristics of a Modular Robot through Genetic Algorithm

Sunil Pranit Lal, Koji Yamada, and Satoshi Endo

Complex Systems Laboratory, Department of Information Engineering,
Faculty of Engineering, University of the Ryukyus,
1 Senbaru, Nishihara, Okinawa 903-0213, Japan
sunil@eva.ie.u-ryukyu.ac.jp,{koji,endo}@ie.u-ryukyu.ac.jp

Abstract. In this paper we present an approach to transform individual behaviour of homogenous sub-systems to yield desired global behaviour of the overall system. As a test bed we consider the brittle star robot as the system which is composed of homogenous modules (sub-systems). Using genetic algorithm, the rotational motion of the individual modules is translated into rectilinear motion of the robot. We argue that given a set of sub-system level behaviours, it is better to discover intermediate system level infinitesimal behaviours as a stepping stone to developing desired system level global behaviour.

Key words: Emergence, behaviour, motion control, modular robot, genetic algorithm

1 Introduction

The notion that the whole is greater than the sum of the parts is very applicable when considering the concept of emergence in complex systems. Emergent structures in nature, such as ant colonies, neural networks and cellular automata provide inspiration for developing equivalent computational models for solving difficult problems [1]. Given a complex system, in our case a modular robot, we are interested in studying how the local interactions of the modules can lead to emergence of locomotion. For this purpose we leverage off genetic algorithm (GA) as a means to transform the individual modular behaviour into desired system level behaviour

Inspired by biological adaptations, genetic algorithm is essentially a search technique used extensively to solve optimization related problems [2]. In the literature many notable contributions have been made in the field of robotics and control using evolutionary approach. Kamimura et al. [3] used neural oscillator as a central pattern generator for controlling a modular robot (M-TRAN II). The optimal parameters for the pattern generator were evolved using GA. Reil and Husbands [4] successfully simulated bipedal straight-line walking using recurrent neural network whose parameters were evolved by GA. Porting genetically evolved neural network controller for a hexapod robot from simulation model to actual hardware was demonstrated by Gallagher et al. [5]. One of the conclusions reached by them was that the evolved controller performed extremely well in real world in spite of the fact that inertia, noise and delays were not taken into account in the simulation.

The motion of the brittle star-typed robot considered in this paper requires coordinated movement of the modules. In prior research, cellular automata-based control model [6] and neural network control model [7] were developed. The parameters evolved for these control models were highly specific to the given task such as rectilinear locomotion. From this we came to the realization that if the robot is required to traverse any arbitrary path, a new set parameters need to be evolved each time, which is quite a time consuming process. This has lead us to develop a novel approach where by the infinitesimal movements of the robot is evolved from the interactions of the individual modules and then these infinitesimal movements can be logically combined to produce desired global motion characteristics of the robot. The effectiveness of the proposed approach was verified using simulation model of the robot developed using Open Dynamics Engine [8]

The paper is organized as follows: The decomposition of the motion control problem in terms of state transitions is presented in Sect. 2. Following it is a descriptive account of evolution of infinitesimal motion patterns using GA in Sect. 3. Section 4 discusses a logical method of combining the infinitesimal movements to produce desired motion characteristics, and finally Sect. 5 concludes this paper.

2 Motion Control Problem Formulation

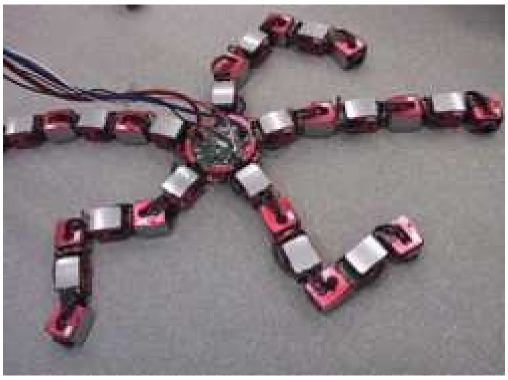
Given a system consisting of homogenous sub-systems with a set of behaviour and constraints, we are interested in knowing how the interaction of the subsystems can lead to emergence of system level behavior. In particular we consider the brittle star robot (Fig. 1(a)) as the system which is composed of homogenous modules (sub-systems). Each module (Fig. 1(b)) incorporates an onboard micro controller (BASIC stamp 2sx), actuator (RC Servo Futaba S5301) and two touch sensors.

Moving the robot requires coordinated movement of the individual modules. Thus the problem is how to transform the individual behaviour of modules (rotational motion) into emergent global behaviour of the robot, such as walking in a straight line.

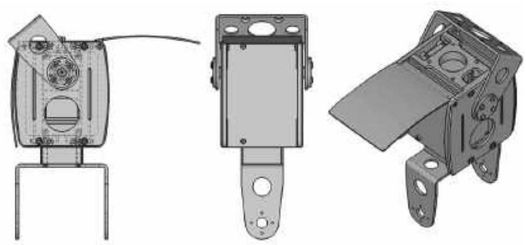
One way to visualize the module is to consider it as a state machine. Since the motion of any given module is basically rotation between an allowable range, the state of j^{th} module in the i^{th} leg at time, t is given by (1).

$$S_{ij}^t \in \{\theta \mid \theta_{min} \leq \theta \leq \theta_{max}\} . \tag{1}$$

This makes the robot a collection of state machines, or to put it differently, a state machine of state machines. The state of the robot with n legs and m modules per leg at time, t is thus given by (2).



(a)



(b)

Fig. 1. 1(a)The brittle star robot 1(b) Individual module connected to make up the leg

$$R^t = \{S^t_{ij} \mid 0 \leq i < n, 0 \leq j < m\} . \tag{2}$$

Therefore the problem of motion control of the robot becomes the task of finding optimal sequence (O_T) of state transitions, which would transform the robot producing desired locomotion.

$$O_T = R^0, R^1, R^2, \ldots, R^T . \tag{3}$$

It should not take much imagination to realize the shear magnitude of the search space that needs to be explored to find near optimal solution, and thus the need for evolutionary computational approach.

3 Evolving Motion Characteristics using Genetic Algorithm

Genetic algorithm or for that matter any other evolutionary algorithm provide solution which is highly suited to a specific problem. For instance, using GA to find optimal sequence of state transitions of the robot to traverse a given path will generate a sequence for that specific path, and as such it will be futile for any arbitrary path.

Differing from conventional wisdom, our approach (Fig. 2) is to evolve infinitesimal system level behaviour from sub-system level behaviour and then logically combine the infinitesimal behaviour to yield desired global behaviour of the system.

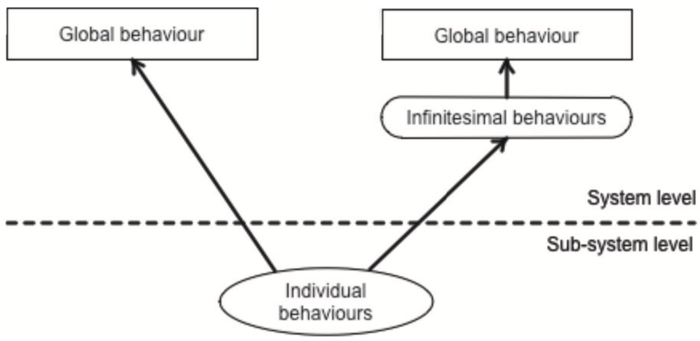


Fig. 2. Approaches in deriving global behaviour from individual behaviour.

Given the dimensions of the robot, length of a single leg being 54cm, it was decided that 2cm would suffice as the infinitesimal straight line distance. While the robot can move this distance in any direction, we decided to limit it to 8 directions corresponding to the cardinal and primary inter-cardinal points on a compass.

3.1 Genetic Encoding

We begin by discretizing the angular range $[-\frac{\pi}{3}, \frac{\pi}{3}]$ of the modules into 16 states which provided fairly granular control of the robot. It is not known as a priori the number of state transitions (R^t) required for each of the eight possible infinitesimal behaviour corresponding to robot's movement in eight direction, thus variable length chromosome was used. To enable smooth and continuous motion across any combination of infinitesimal behaviours, the robot is required to start at predetermined initial state and return to that initial state at the end of state transition for any infinitesimal behaviour (Fig. 3). Finally, the state transitions for movement in each direction was evolved separately and independent of each other.

With 30 modules in the robot and 16 states per module, the equivalent binary representation of the state of the robot requires 120 bits. The state search space grows exponentially (2^{120q}) with the number of states (q) encoded in the chromosome.

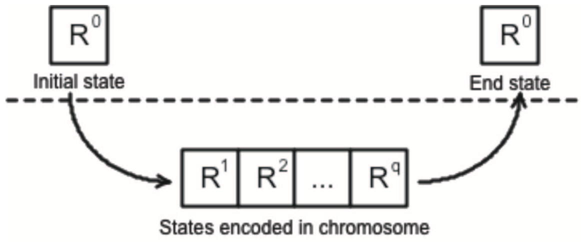


Fig. 3. State transition from predetermined initial state through intermediate states encoded in the chromosome, and back to the initial state.

3.2 Fitness Function

The fitness of each chromosome is evaluated by applying state transitions encoded in the chromosome to the simulated model of the robot. The robot starts at an initial position and is required to move towards a fixed target position located 2cm away from the initial position in the desired direction.

As the robot moves we would like to minimize the deviation of the robot from the desired path to the target, as well to minimize the euclidean distance between the robot's final position and the target position (Fig. 4). In addition it is highly desirable to minimize the number of state transitions as doing so not only reduces the search space but also improve the robot's performance in terms of speed in reaching the target. Taking these issues into consideration the derived fitness function (F), which needs to be minimized, is a function of deviation from path (D_{path}), deviation from target (D_{target}) and number of states (N) in excess of minimum number of states (N_{min}) that can be encoded in the chromosome.

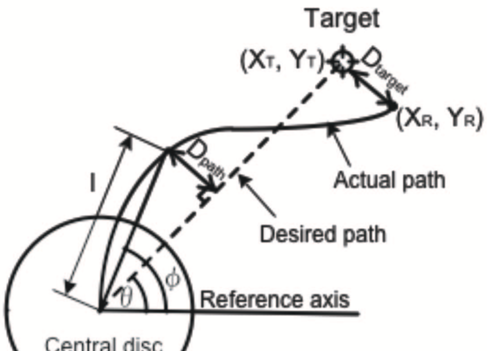


Fig. 4. Derivation of the fitness function.

$$F = D_{path} + D_{target} + N \ . \tag{4}$$

$$D_{path} = \mathbf{max} \left| l sin(\phi - \theta) \right| \ . \tag{5}$$

$$D_{target} = \sqrt{\left(x_T - x_R\right)^2 + \left(y_T - y_R\right)^2} \ . \tag{6}$$

$$N = N_{actual} - N_{min} \ . \tag{7}$$

3.3 Genetic Operators

Based on the fitness of the chromosomes in the population, GA operations; namely selection, crossover and mutation are applied to the whole population. The selection process screens the individuals such that the fitter individuals have higher probability of making it through to the next generation. The roulette wheel selection method was used to perform selection. It is worth mentioning that the roulette wheel selection method is customized for maximization problems, thus the fitness function, F was transformed to $F' = -F + C$, where C is positive constant.

The crossover operation essentially emulates mating process by exchanging and combining genes from selected parent chromosomes to produce offsprings, with the hope that they may have better fitness than the parents. The selected pairs of chromosomes are crossed over using one-point crossover at randomly chosen locus with a crossover probability of P_C. Crossover operation is skipped for a chromosome pair if the offspring produced would exceed the maximum chromosome length in terms of maximum number of states encoded (N_{max}).

Mutation is a way to introduce diversity within the population, thus enabling better exploration of the search space. After the selection and crossover operation, a two fold mutation operation was applied to the entire population with a mutation probability, P_M. Given the binary representation and variable length of the chromosome, the mutation operation involved random bit flips, and resizing chromosome between $[N_{min}, N_{max}]$ by deleting states from chromosome at random position, or inserting randomly created states at random positions.

3.4 Simulation Results

The simulation was carried out over numerous trails involving different target positions, and using parameters shown in Table 1. In each trial, initial population of chromosomes of size (POP_SIZE) was randomly generated and GA was executed for a number of generations (MAX_GEN). For each generation, the fitness of all the chromosomes in the population is evaluated after which genetic operators are applied to the population to create the next generation.

The motion characteristics of the best evolved state transitions in terms of amount of deviation from path and target is highlighted in Table 2. The

Table 1. Summary of simulation parameters.

Parameter	Value
P_C	0.80
P_M	0.005
N_{min}	1
N_{max}	6
POP_SIZE	55
MAX_GEN	400

fitness of the best chromosomes approached the optimal value of zero. The small differences in the robot's performance can be attributed to the fact that the orientation of the pentagonal robot varies with respect to the frame of reference for each of the direction of movement. For example, to move in 0° direction, 3 legs face forward and 2 legs face backward, however to move in 180° direction, 2 legs face forward and 3 legs face backward, thus the resulting differences in the motion characteristics.

It is worth mentioning that at the beginning of GA, the best chromosomes had variable lengths, however towards the end all of them converged to encoding just a single state of the robot. In other words, including the predetermined initial and final states, the total number of state transitions required to effect motion of 2cm in any of the directions turned out to be three.

Table 2. Characteristics of the best evolved state transitions for each of the directions.

Direction (degrees)	Fitness	D_{path} (mm)	D_{target} (mm)
0°	1.40	1.03	0.37
45°	1.70	1.30	0.40
90°	1.32	1.02	0.30
135°	1.39	1.23	0.16
180°	2.00	1.46	0.54
225°	3.20	1.98	1.22
270°	1.83	1.68	0.15
315°	2.51	1.61	0.90

4 Discussion

In this section we consider a simple method of logically combining the "infinitesimal" movements in producing desired motion. Basically the goal is for the robot to traverse along a specified path. At any given point in time along the path, the robot has a set eight directions to choose from. The control strategy adopted involves determining the position of the center of the robot in each of the eight directions at the next step. Among these positions, the position which has the least deviation from the path and is in the direction of travel, is chosen as the direction in which the robot will move next (Fig. 5).

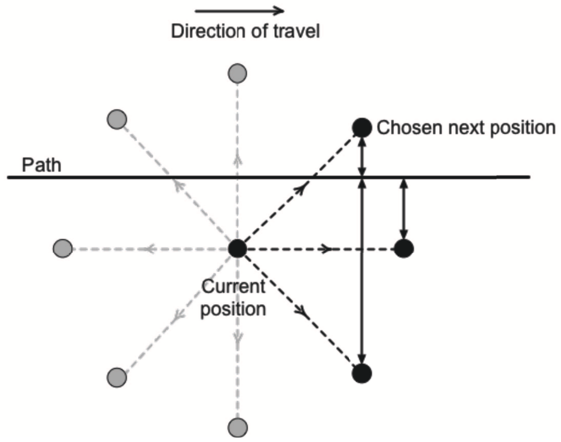


Fig. 5. Control strategy in choosing the next move.

The state transitions evolved in the previous section focused on rectilinear movements of 2cm. We are interested in knowing if these infinitesimal movements can be effectively combined to cover greater distances. Given the task of moving in a straight line distance of 20cm, Fig. 6 shows the simulation results of robot using combination of infinitesimal movements compared with a robot using state transitions specifically evolved to traverse the given path. As can be excepted, the genetic algorithm yields near optimal state transition for the specific path, thus the specially evolved robot outperforms the robot with combined infinitesimal movements in terms of total number of state transitions to reach the target. The small errors associated with the individual infinitesimal movement (Table 2) compound to significant error margins when those movements are combined, thus shedding some light on the observed fluttering behaviour of the robot around the desired path as it constantly adjusts its course to stay on the path.

While the motion characteristics produced by combining infinitesimal movements is far from optimal, a strong argument in favor of this approach is that it is highly adaptable to different problems. Consider a scenario where the robot is required to traverse an arbitrary path. Using GA to evolve a set of state transitions for that path, aside from consuming generation after generation of computational time, will yield a result which is specifically suited to that path and will be of no value in traversing some other path. On the other hand, once the infinitesimal movements have been evolved they can be combined in a number of ways to adapt to any arbitrary path without requiring any additional training time as shown in Fig. 7.

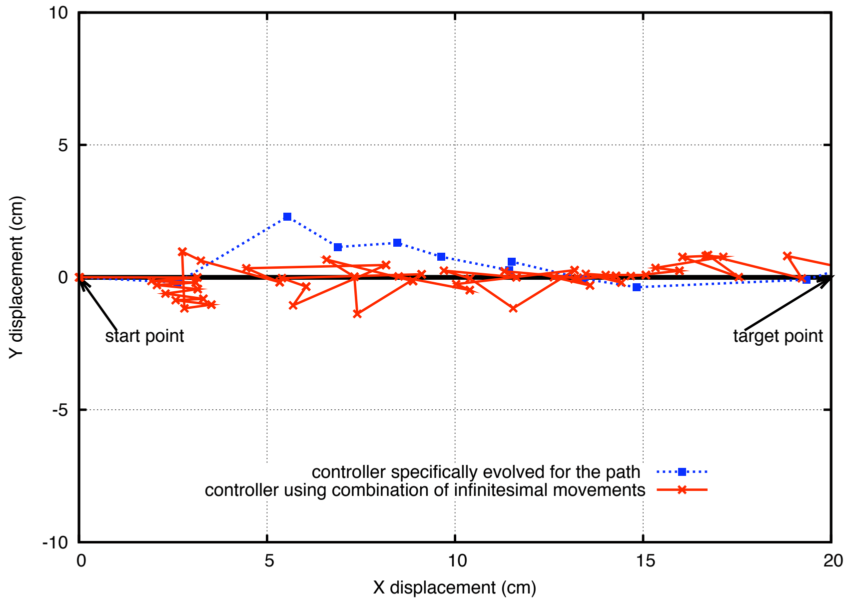


Fig. 6. Displacement of the central point of the robot using different control strategies.

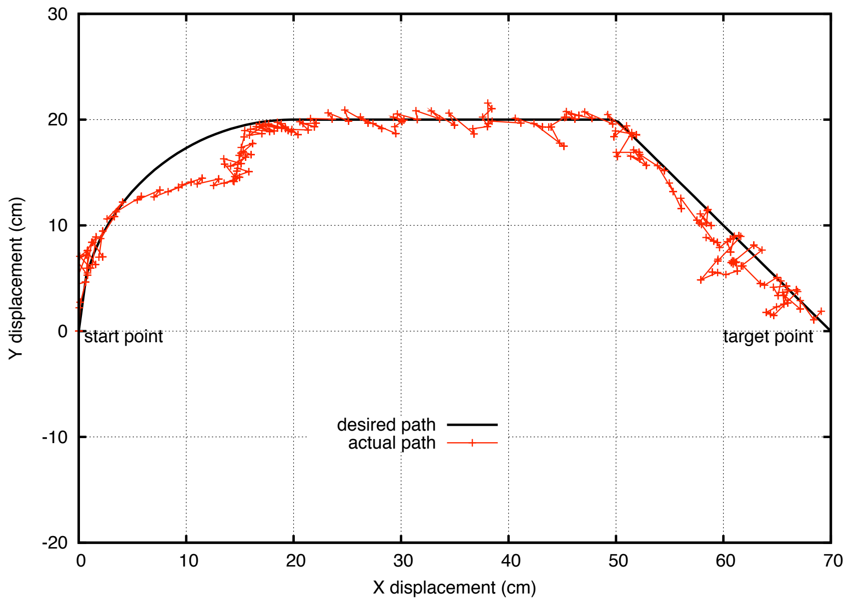


Fig. 7. Displacement of central point of the robot using combination of infinitesimal movements to traverse an arbitrary path.

5 Conclusion

In this paper we considered the emergence of global behaviour of a complex system consisting of homogenous sub-systems. Specifically we studied a modular robot as an instance of a complex system in an effort to derive global motion characteristics of the robot from individual modular behaviour. Our proposed approach consisted of evolving infinitesimal rectilinear movements from rotational motion of individuals modules, and then logically combining the infinitesimal movements to produce desired motion characteristics of the robot. This approach, though far from optimal, was highly adaptable to any arbitrary path that the robot was required to traverse.

The infinitesimal behaviour of the robot consisting of small rectilinear movements in eight directions was rather elementary. Additional infinitesimal behaviours such as clockwise and counter clockwise rotation can be added to increase the set of movements available to the robot. In combining the infinitesimal movements, the errors compounded causing performance degradation of the robot. Future work towards implementing the control strategy using controllers that can operate well in the presence of uncertainty such as fuzzy logic controller should prove fruitful.

References

1. Holland, J. H.: Emergence: From Chaos to Order. Addison Wesley, Reading, Massachusetts (1998)
2. Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Massachusetts (1989)
3. Kamimura, A., Kurokawa, H., Yoshida, E., Tomita, K., Murata, S., Kokaji, S.: Automatic Locomotion Pattern Generation for Modular Robots. In: IEEE International Conference on Robotics and Automation, pp.714–720. IEEE Press, New York (2003)
4. Reil, T., Husbands, P.: Evolution of central pattern generators for bipedal walking in a real-time physics environment. IEEE Transactions on Evolutionary Computation. 6(2), 159–168 (2002)
5. Gallagher, J. C., Beer, R. D., Espenschied, K. S., Quinn, R.D.: Application of evolved locomotion controllers to a hexapod robot. Robotics and Autonomous Systems. 19, 95–103 (1996)
6. Lal, S. P., Yamada, K., Endo, S.: Studies on motion control of a modular robot using cellular automata. In: Sattar, A., Kang, B. H. (eds.) AI 2006: Advances in Artificial Intelligence. LNAI, vol. 4304, pp. 689–698. Springer, Heidelberg (2006)
7. Lal, S. P., Yamada, K., Endo, S.: Evolving Motion Control for a Modular Robot. In: Ellis, R., Allen, T., Petridis, M., (eds.) Applications and Innovations in Intelligent Systems XV. pp. 245–258. Springer, London (2007)
8. Smith, R.: Open Dynamics Engine User Guide, `http://www.ode.org/`