Robotica (2001) volume 19, pp. 535–542. Printed in the United Kingdom © 2001 Cambridge University Press

Locomotive gait generation for inchworm-like robots using finite state approach

I-Ming Chen,* Song Huat Yeo, and Yan Gao

School of Mechanical and Production Engineering, Nanyang Technological University, Nanyang Ave, Singapore 639798.

(Received in Final Form: December 5, 2000)

SUMMARY
The gait of a multi-segment inchworm robot is a series of actuator actions that will change the shape of the robot to generate a net motion. In this paper, we model the multi-segment inchworm robot as a finite state automaton. Gait generation is posed as a search problem on the graph described by the automaton with prescribed state transitions. The state transitions are defined based on the kinematics of robot locomotion. The auxiliary actuator concept is introduced. Single-stride and multi-stride gait generations are discussed. Single-stride gaits exhibit fault-tolerant and real-time computation features that are necessary in actual applications. Both computer simulation and experimental hardware platform are developed for various aspects of gait generation and planning.

KEYWORDS: Inchworm robot; Gait generation; Finite state model.

1. INTRODUCTION
An inchworm-like robot is a mobile robot that imitates the locomotion pattern of a natural inchworm. This kind of robots usually consists of interconnected actuating modules that can either deform in the direction of travel (extensors) or produce friction against the environment (grippers). The locomotion of the robot is through a series of cyclic actuator actions to change the shape of the robot termed gaits. The gaits exploit the constrained nature of a robot's interaction with its environment to generate net body motion. Applications of inchworm robots are for the inspection and material delivery tasks in narrow and highly constrained environments. In this paper, we study the gait generation problem for a class of multi-segment inchworm robots using simple *binary* actuators (i.e. with only on and off states). As the number of segments in the robot increases, the types of gaits increase as well. Different gaits exhibit different kinematic and dynamic behaviors, which are crucial to the application environment. Thus, how to produce useful gaits becomes an important problem.

The gait generation algorithms can be applied to the control and motion planning of inchworm-type robotic endoscopes for medical inspection[1,2] and pipe inspection systems.[3] Hyper-redundant robot (snake-like or inchworm-like robot) locomotion has been studied by Chirikjian and Burdick[4] and Poi *et al*.[5] It is formulated based on the continuous "backbone curve" of the robot. Locomotion gaits are generated based on the wave theory. The works of Kelly and Murray[6] and Murray et al[7] further describe robot locomotion in terms of the geometric phase associated with a connection on a principal bundle and develop local trajectory generation strategy. Basically, the mechanical systems studied in above works assume that the length of the body can change continuously. For inchworm robots used as inspection tools or material delivery systems, precise control of the robot's position is usually not necessary with the assistance of the supporting equipment. Therefore, simple actuators with only binary actions, i.e. on and off, are frequently used in the design of the grippers and the extensors.[8] The condition between the on and off states is considered as transient behavior, which will settle within a very short time. Thus, the length of the robot will assume only a numbers of finite values. Based on this fact, we can use the finite automaton model[9] to study the inchworm robot with only binary actuation actions. The action of the inchworm actuators can be described as states. A gait becomes a sequence of state transitions that follow the kinematics of locomotion. Since a finite automaton can be expressed as a directed graph with the states as the nodes and the transitions as the arcs, the gait generation problem becomes a graph search problem. The advantage of using this approach is that it reduces the effort in gait generation from a continuous state space to a binary state space. Also, online gait generation and planning can be achieved directly with additional structures imposed on the finite automaton model.

Single-stride gait and multi-stride gait generations are both investigated. The concept of *auxiliary grippers and extensors* are introduced and used in both cases. The strategy for single-stride gait allows both forward and backward motions, and has fault-tolerant feature. On-line gait generation and gait change can be achieved through single-stride gaits. Multi-stride gait generation leads to the standing wave gait, which is the fastest gait in terms of the number of state transitions in a complete gait cycle. Computer simulation of various gait generation algorithms is implemented. A simple experimental inchworm robot platform based on the solenoid actuators is built to verify the result of simulation and to study the dynamic behavior of the gaits.

2. SIMPLE INCHWORM ROBOT MODEL
As actual robots may adopt different mechanical design, here we assume a simplified model of the inchworm robot

*Corresponding author.

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press

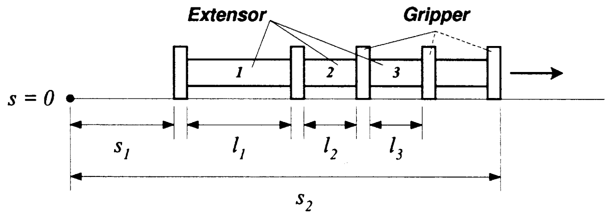

Fig. 1. Inchworm robot model.

actuators as depicted in Figure 1. A robot with n segments has n extensors and $(n+1)$ grippers. Each gripper or extensor can be actuated independently. All grippers employ identical modular mechanical design. The main function of the gripper is to provide friction for robot locomotion. Without loss of generality, the width of the gripper is assumed to be a constant. Similar modular design concept is applied to the extensor as well. The stroke lengths of all extensors are the same. The fully retracted and fully stretched lengths are denoted as $l_{\min}$ and $l_{\max}$, respectively. The intermediate length of the extensor, l_i, is within the range of $l_{\min} \leq l_i \leq l_{\max}$. We assume that the robot moves in an environment with one-dimensional topology, for example, a straight pipe with rigid wall, or curved human intestine with elastic wall. The actual robot design can make the gripper and extensor modules negotiate the bend in the constrained environment and compliant to the shape of the surroundings.

3. KINEMATICS OF INCHWORM LOCOMOTION

We first use an inchworm robot whose body length can change continuously to illustrate the principle of locomotion. For an n-segment robot shown in Figure 1, the kinematic constraint to allow the robot move in a 1-D space is given by

$$\dot{s}_1 + \sum_{1}^{n} \dot{l}_i - \dot{s}_2 = 0 \qquad (1)$$

where s_1 and s_2 are the positions of the rear gripper and the front gripper. According to Kelly and Murray,[6] $(s_1, s_2) \in G \equiv (R,+)\times(R,+)$ is the group variable that reflects the position of the robot as a subgroup of the group of rigid motion, and $(l_1, l_2, \ldots l_n) \in M \equiv R^n$ is the base space variable that describes the body motion (each segment). The expression of (1) is a connection on the trivial principal bundle $Z = M \times G$. It describes how trajectories in the base space M are related to the motion in the group G. The gait is thus a closed path in M that will generate net motion in G[7]. Figure 2 and 3 demonstrate the gaits of two-segment and three-segment inchworm robots respectively and their corresponding closed paths in M. Both gaits will generate a net motion with one stride length $(l_{max} - l_{min})$ along the x-direction. Other types of gaits with different stride lengths can be achieved with the choice of initial conditions of the base space variables and the closed path to be taken in M. One will notice that as the number of the segment increase, the dimension of the base space increase as well. To generate suitable gaits (or the closed paths) in high dimensional spaces become difficult as the number of possible paths to be taken increases.

4. MODELING INCHWORM AS AN AUTOMATON

Based on the simple mechanical design and actuation principle, we can model a multi-segment inchworm robot whose grippers and extensors have only simple binary actuation actions as a finite automaton.[9] The extensor and gripper each has only binary value states "0" and "1".

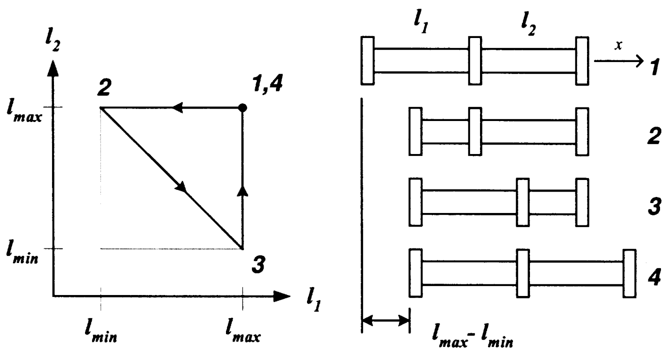

Fig. 2. Two-extensor robot gait.

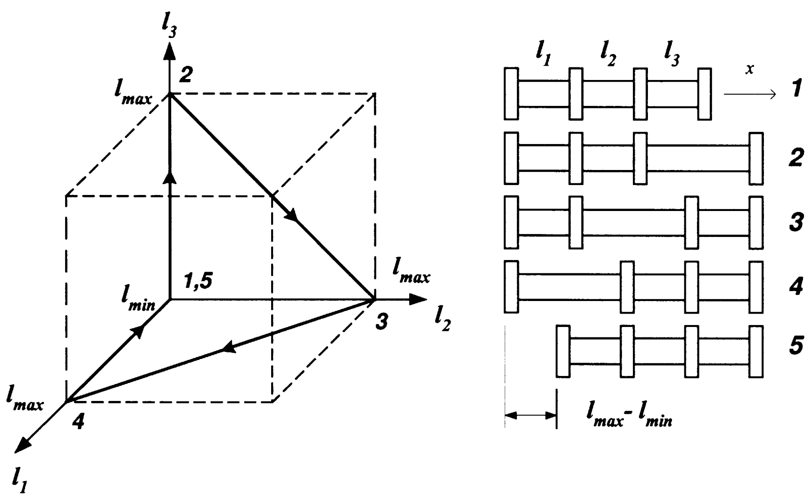

Fig. 3. Three-extensor robot gait.

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press

DEFINITION 1: The extensor state of an n-segment inchworm robot is an n-tuple vector with binary numbers, $q=(x_1, x_2, \ldots, x_n)$, *where* $x_i \in \{0, 1\}$. When $x_i=0$, Extensor i is retracted; $x_i=1$, Extensor i is fully stretched.

DEFINITION 2: The gripper state of an n-segment inchworm robot is an (n+1)-tuple vector with binary numbers, $p=(y_1, y_2, \ldots, y_n, y_{n+1})$, where $y_i \in \{0,1\}$. When $y_i=0$, Gripper i is released; $y_i=1$, Gripper i is fully activated.

Because the locomotion of the inchworm robot is achieved by the deformation of the extensors, the inchworm gait can be defined solely on the extensor states q. The gripper state p can be determined by the transition function between the consecutive extensor states.[8]

DEFINITION 3: The gait of an n-segment inchworm robot is a sequence of extensor states $(q_0, q_1, \ldots, q_f)$ such that $q_0=q_f$.

Definition 3 implies that the gait is a closed trajectory in the state space of the extensor $\{0.1\}^n$ (or a "digitized" base space M). For instance, the gait of the two-segment robot shown in Figure 2 is (1,1)→(0,1)→(1,0)→(1,1). The gait in Figure 3 is (0,0,0)→(0,0,1)→(0,1,0)→(1,0,0)→(0,0,0). Note that the transition between any states of the robot's extensors must satisfy the kinematic constraints imposed by the environment.

DEFINITION 4: The gait generator of an n-segment inchworm robot is a 5-tuple $A=(Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of inchworm states, Σ is a non-empty set of event labels called an *input alphabet*, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and δ is the *transition function* mapping $\delta: Q \times \Sigma \to Q$. That is, $\delta(q, a)$ is a state for each state q and input symbol $a \in \Sigma$.

Alternatively, A can be represented by a directed graph in which the nodes are the states in Q, the arcs are the transition defined by the function δ, and the set of labels for the arcs are the alphabets in Σ. The gait generation problem becomes finding a path (a sequence of arcs) from the root node (denoted the initial state q_0) to the desired node (denoted the final state q_f) as illustrated in Figure 4.

One can draw an analogy between the continuous inchworm robot model stated in Section 2 and the finite-state inchworm model. The base space M corresponds the binary state space $\{0,1\}^n$; the kinematic constraints are equivalent to the state transition functions that follow

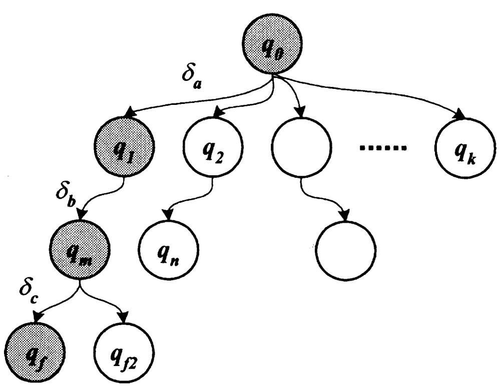

Fig. 4. Search on the graph formed by A.

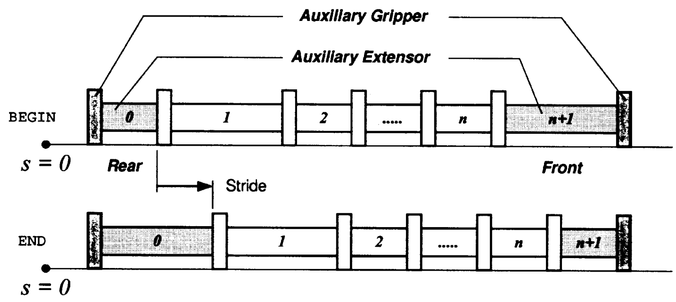

Fig. 5. Auxiliary grippers and extensors.

kinematic rules that we will study in the following section. Trajectory generation (gait generation) in the base space then becomes a search problem on the directed graph A. Through this transformation, standard graph search algorithms[10] can be utilized in the gait generation.

5. SINGLE-STRIDE GAIT GENERATION

The inchworm robot will move only one stride-length with respect to their previous location after a cycle of single-stride gait motion. To achieve this type of motion, only two extensors will be coordinately actuated at the same time. Thus, this type of gaits could exhibit in the inchworm robot with any number of segments.

5.1 Auxiliary grippers and extensors

To facilitate gait generation, one auxiliary extensor and one auxiliary gripper are added to the front and rear of the inchworm robot respectively as shown in Figure 5. Physically the auxiliary actuators do not exist; they only reside in the robot controller or the computer simulator.

Initially, the two auxiliary extensors are set to the retracted position (0) and the stretched position (1) respectively depending on the direction of travel. For example, if the robot shown in Figure 5 moves to the right, the initial setting of the right and the left auxiliary extensors should be 1 and 0 respectively. The right one is then called the *front* auxiliary extensor and the left one the *rear* auxiliary extensor. As one gait cycle is completed, the physical extensors return to the initial states but the front auxiliary extensor will be retracted and the rear one will be extended (Figure 5). The front and rear auxiliary grippers remain stationary through out the entire gait. When a new gait cycle starts the front and rear auxiliary extensors are reset to 1 and 0, respectively. If the robot moves to the left, the initial setting of the right and the left auxiliary extensors should be 0 and 1. The roles of the front and rear extensors are swapped. Obviously, the auxiliary grippers remain stationary and serve as the reference frame for the locomotion. The inchworm robot moves one-stride length *with respect to the auxiliary front and rear grippers*.

DEFINITION 5: *The augmented extensor state* of an n-segment inchworm robot is an (n+2)-tuple binary vector: $q'=(x_0, x_1, \ldots, x_n, x_{n+1})$. The states of the right and the left auxiliary extensors are x_0 and x_{n+1}, respectively.

Based on this definition, an inchworm gait becomes a sequence of augmented extensor states $(q'_0, q'_1, \ldots, q'_f)$ *such that* $q'_0=(x_0^0, q_0, x_{n+1}^0)$, and $q'_f=(x_0^f, q_0, x_{n+1}^f)$, where q_0 is the

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press

given extensor initial state. If $(x_0^0, x_{n+1}^0)=(0,1)$, then $(x_0^f, x_{n+1}^f)=(1,0)$. If $(x_0^0, x_{n+1}^0)=(1,0)$, then $(x_0^f, x_{n+1}^f)=(0,1)$. The gait generator becomes $A=(Q', \Sigma, \delta, q_0', F')$, where Q' is the set of the augmented extensor states; $F' \subseteq '$ is the final state set.

5.2 *Simple one-stride gaits*

A *one-stride gait* will make the inchworm robot move one-stride length when one gait cycle is completed. Based on the simplified inchworm robot model, the positions of every gripper will move one-stride length as well. To make a gripper move, it is necessary to coordinate the motion of the two neighboring extensors with simultaneous stretching and retraction actions. In state 1 of Figure 2, Gripper 2 cannot move because both Extensors 1 and 2 are stretched to the limit. Gripper 2 can only move one-stride length to the right from state 2 to state 3. This action is coordinated by the stretching of Extensor 1 and the contraction of Extensor 2 while the grippers at both ends remain stationary to provide friction for locomotion. Re-looking at state 1, Gripper 1 can be similarly coordinated by the extension of the rear auxiliary extensor, although it does not exist physically, and the contraction of Extensor 1 from state 1 to state 2. In this case, the rear auxiliary gripper and Gripper 2 remain stationary.

In terms of the binary actuator states, the coordinated stretching and contraction of neighboring extensors makes the state of the neighboring extensors change their value from (0,1) to (1,0) if the robot moves to the right, and vise versa, if the robot moves to the left. Based on this observation, we can use the following strategy to generate inchworm gaits for the forward motion (right):

<Forward motion gaits>

- **Initialization**

Given an inchworm robot with extensor state q_0, set the initial augmented state $q_0'=(0, q_0, 1)$.

- **State transition function**

Let $q_a'=(x_0^a, x_1^a, \ldots, x_n^a)$ and $q_b'=(x_0^b, x_1^b, \ldots, x_n^b)$. The state transition $q_b'=\delta_f(q_a', i)$ is defined as:

For any $i \in \Sigma=\{0, 1, \ldots, n\}$,

If $(x_i^a, x_{i+1}^a)=(0,1)$
Then $(x_i^b, x_{i+1}^b)=(1,0)$ and $x_k^b=x_k^a$ for $k\neq i, i+1$
else $q_b'=q_a'$

- **Final state**

One complete gait cycle will be terminated at $q_f'=(1, q_0, 0)$.

Note that each one-stride state transition only allows two extensors to move simultaneously. The other extensors remain stationary. The unaffected grippers still hold on to the environment with friction contact. With a given $q_0'=(0, q_0, 1)$, a sequence of labels in Σ can be generated from δ_f to satisfy the final state $q_f'=(1, q_0, 0)\in F'$. This label sequence represents the sequence of the extensor states of a gait. Because there are a number of possible state transition functions allowed for state q_a' at any step, the forward gait generator A can be represented by a directed tree with the root node as $q_0'=(0, q_0, 1)$ and all the pendant nodes as $q_f'=(1, q_0, 0)$. Every directed path from the root node to a pendant node is a valid gait (Figure 4). Using the exhaustive search[11] on the graph, we are able to find all possible gaits with the same initial condition. Figure 6 illustrates the one-stride gait generation of a tree-extensor robot. (Auxiliary actuators are not shown.) Five different gaits can be produced with identical initial conditions. Note that the number of steps in each of the gait is the same. We can also show that it will take $n+1$ state transitions to complete one single-stride gait for an n-extensor inchworm robot regardless of their initial conditions.[8] In normal condition, using any one of the gaits originated from the same initial condition will be able to move the robot. Thus, we say that the complexity of the single-stride gait is of $O(n)$ time, where n is the number of extensors in the inchworm robot.

To move the robot backward, one can simply reset the front and rear auxiliary extensors with 1 and 0 respectively. The following strategy is a complete description of the backward motion.

<Backward motion gaits>

- **Initialization**

Given an inchworm robot with extensor state q_0, set the initial augmented state $q'_0=(1, q_0, 0)$.

- **State transition function**

Let $q_a'=(x_0^a, x_1^a, \ldots, x_n^a)$ and $q_b'=(x_0^b, x_1^b, \ldots, x_n^b)$. The state transition $q_b'=\delta_b(q_a', i)$ is defined as:

For any $i \in \Sigma=\{0, 1, \ldots, n\}$,

If $(x_i^a, x_{i+1}^a)=(1,0)$
Then $(x_i^b, x_{i+1}^b)=(0, 1)$ and $x_k^b=x_k^a$ for $k\neq i, i+1$
else $q'_b=q'_a$

- **Final state**

One complete gait cycle will be terminated at $q'_f=(0, q_0, 1)$.

This one-stride gait generation strategy also allows the gaits to be changed during any phase of the movement. One can interrupt the robot at any state and make the current state as the new initial state and reset the front and rear auxiliary extensors with 1 and 0 respectively. A new forward gait can be generated based on this new initial state using the identical transition function. Likewise, the robot can move backward immediately by resetting the front and rear auxiliary extensors to 0 and 1, respectively, and using the backward transition function to generate the backward gait. Since the interrupt can be issued at any instance of the motion, the gait can be changed right on the spot of operation.

5.3 *Fault-tolerant gaits*

The single-stride gait has a robust fault-tolerant feature. If any extensor in the robot failed, the robot can still move

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press

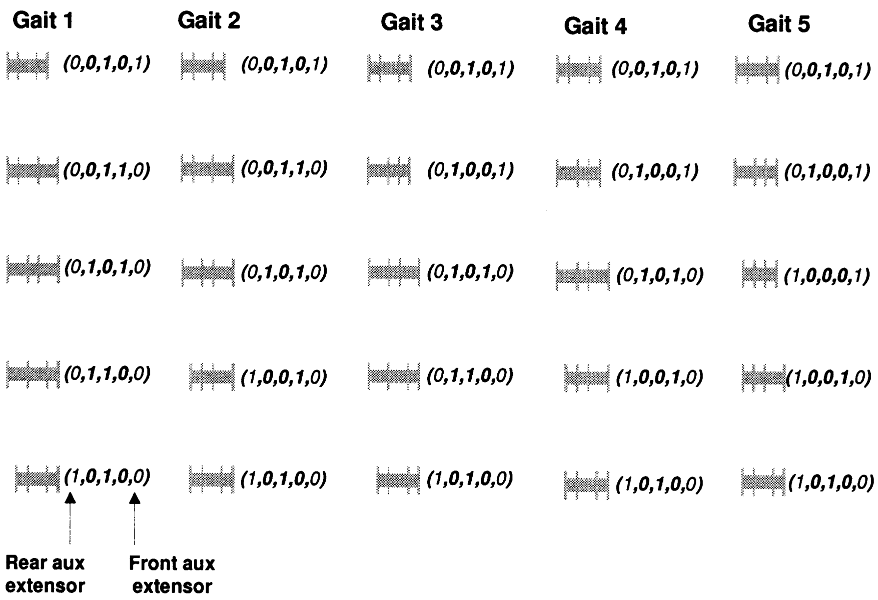

Fig. 6. Single-stride Gaits with identical I.C.

around by using the rest of the extensors as long as the grippers are functioning. In this situation, we can treat the robot as an under-actuated system. When using the state transition δ_f or δ_b to determine the next valid move, the state of the failed extensor will be omitted. The fault-tolerant gait generation is shown in Figure 7 (auxiliary actuators not shown). Conditions (a), (b) and (c) have the same initial condition but with different faulty segments. In Condition (a), Extensor 1 is failed, so it is treated as a two-extensor robot. It takes only 3 transitions to complete this gait. If the robot has no faulty extensor [Condition (d)], it will take 4 transitions to complete the gait cycle. In principle, the robot will use fewer steps to complete a gait when it has faulty extensors. Since the faulty extensor is being dragged along during the movement, actual locomotion speed will be affected.

6. MULTI-STRIDE GAIT GENERATION

The inchworm robot will move more than one stride-length relative to its previous location after completing one cycle of a multi-stride gait. To achieve this type of motion, multiple pairs of extensors will be coordinately actuated at the same time, which consumes much more power than the single-stride gait motions. In addition, there are conditions imposed on the number of segments in the robot to have multiple-stride motion.

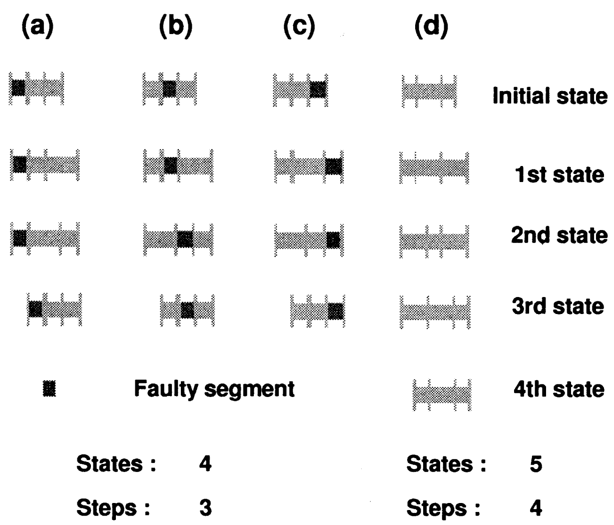

Fig. 7. Fault-tolerant gaits.

6.1 Double-stride gaits

A *double-stride gait* makes the inchworm robot move two-stride length when one gait cycle is completed. Based on the analogy of single-stride gait generation strategy, the double-stride gait requires 4 extensors to be actuated simultaneously (Figure 8). Three intermediate grippers will be released and moved by the stretching of extensors 1, 2 and the contraction of extensors 3, 4. To generate such type of gaits two auxiliary extensors and grippers have to be added to the each end of the robot. With the double-stride gaits, the robot can complete one gait cycle with fewer numbers of state transitions and larger stride motion but at the expense of larger power consumption as compared to the single-stride gaits. Triple-stride gaits and even n-stride gaits can be defined similarly. Nevertheless, to use these types of gaits, the initial states of the extensors must be paired up to enable the subsequent motion that may not be available all the time.

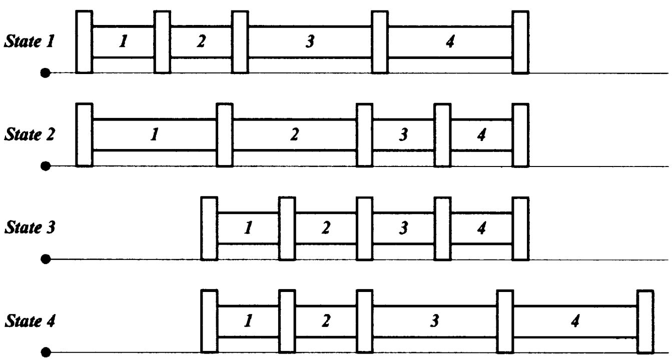

Fig. 8. Double-stride gait.

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press

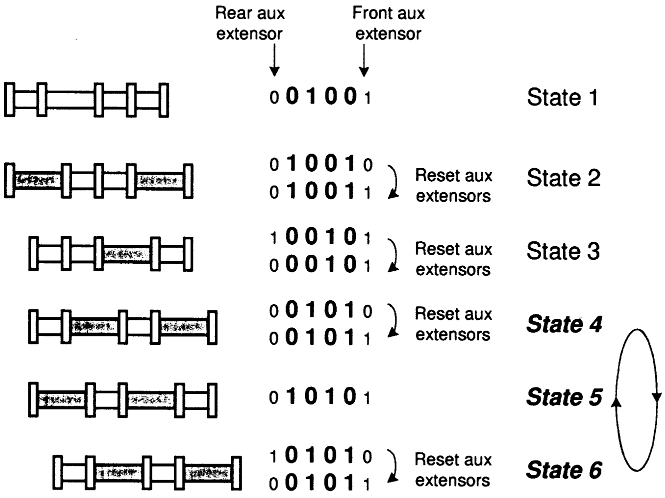

Fig. 9. Standing wave gait generation.

6.2 Standing wave gaits

The other type of multi-stride gaits is the standing wave gait, which requires all the extensors to be actuated at the same time and the neighboring extensors to be activated in opposite way (Figure 9). Therefore, it requires only two state transitions to complete the cycle. Similar to the double-stride gaits, the initial condition usually cannot start the standing wave pattern. Nevertheless, based on the augmented gripper and extensor concept, we propose the following strategy to generate standing wave gaits.

<Standing wave gaits>

- **Initialization**

Given an inchworm robot with extensor state q_0, set the initial augmented state $q'_0=(0, q_0, 1)$.

- **State transition function**

Let $q'_a=(x^a_0, x^a_1, \dots, x^a_n)$ and $q'_b=(x^b_0, x^b_1, \dots, x^b_n)$. The state transition $q'_b=\delta_w(q'_a, i)$ is defined as:

Step 1

For all $i \in \Sigma = \{0, 1, \dots, n\}$,

If $(x^a_i, x^a_{i+1})=(0,1)$

Then $(x^b_i, x^b_{i+1})=(1,0)$ and $x^b_k=x^a_k$ for $k \neq i, i+1$

else $q'_b=q'_a$

Step 2

Set $x^b_0=0$ and $x^b_{n+1}=1$

- **Final state**

The terminal state q'_f has the following property:

$\delta_w(\delta_w(q', a),b)=q'$, where $a, b \in \Sigma$. The sequence of the label (a,b) represents the standing wave gait.

In state transition δ_w, the front and rear auxiliary extensors are reset to 0 and 1 every time. This will ensure the kinematic constraints imposed by the stationary auxiliary grippers are matched for every state. This state transition function also describes a complete standing wave generation process staring from the initial condition, going through a transition phase to activate all the extensors, and finally entering the two-step standing wave gait loop. This behavior is shown in the 4-extensor robot example in Figure 9 (without auxiliary actuators). The extensors changing from 0 to 1 are shaded in the figure for illustration. From state 1 to state 3, the robot is in the transition phase. From state 4 to state 6 and onward, the robot enters the standing wave gait phase. Note that there are always two or three grippers holding to the environment to provide friction force in the standing wave gait loop. The kinematic constraints of locomotion are maintained through out the gait.

7. IMPLEMENTATION

7.1 Computer simulation

To validate the gait generation algorithms, a computer simulated inchworm locomotion program based on the finite automata is developed. The gaits shown in Figures 7 and 8 are generated by the program. It is coded in *Mathematica* running on a PC with Pentium II 233 MHz processor. The computer simulation verified the correctness of all the single-stride and multi-stride gait generation algorithms described in Section 5 and 6. The animated computer simulation results can be found at this website.[12]

7.2 Physical experiments

In addition to computer simulation, we also design and build an experimental platform for the gait generation and planning of the inchworm robot. This is to investigate the dynamic characteristics of various gaits and their interaction with the environment that cannot be observed in the computer simulation. The robot is designed based on modular concept as shown in Figure 10. Each module has a cart-like geometry moving along a horizontal track. There are two solenoid actuators on each of the cart: one for gripping action (emulating the gripper) and the other for extension action (emulating the extensor). The natural state of the gripper is retracted. It can be opened to hold on the wall. Viscous layers are adhered to the walls of the track to

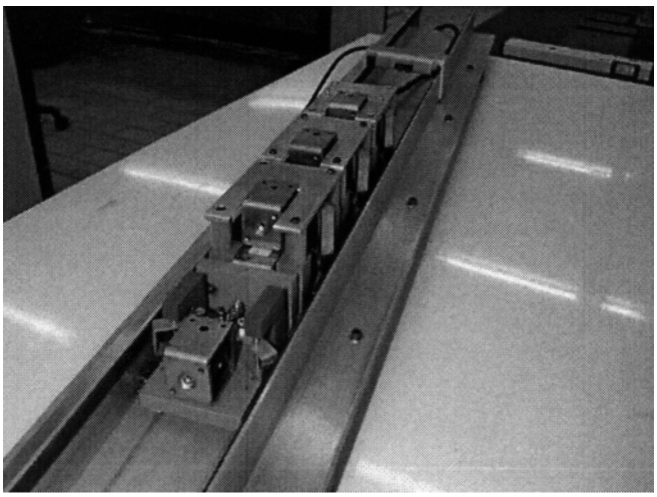

Fig. 10. Inchworm robot gait experiment platform.

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press

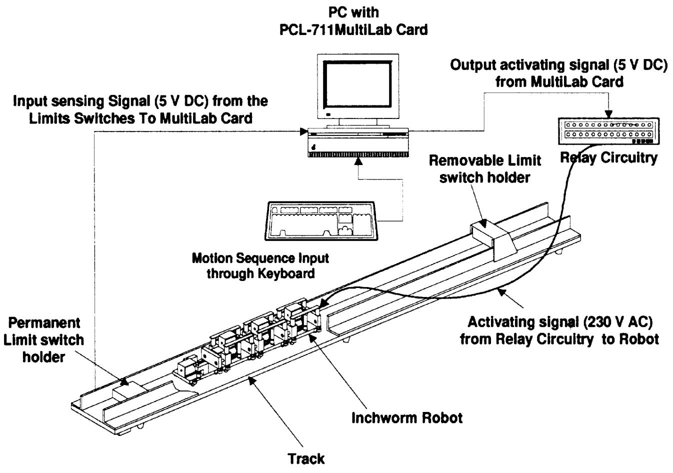

Fig. 11. Inchworm robot system.

provide enough friction. The extensors are naturally retracted as well. Wheels are built under all carts to support the weight of the robot. The overall system is shown in Figure 11. The control of the robot is through a data acquisition interface connected to a PC. After verification from computer simulation, all gait generation algorithms were coded into C language resided on the PC. The schematic diagram of the gait generation function and its relationship with the robot controller is shown in Figure 12. Basically, the gait generator will send a gait command to the robot controller depending on the types of the gaits. The gait command is in fact a bit string of "0" and "1" indicating the "off" and "on" states of the grippers and extensors. As indicated in Section 5 and 6, the gait generation algorithms will only manipulate the gaits represented as bit strings. Therefore, the gait generator can communicate with the robot controller close to their nominal data transmission rate. Also because of the binary string format, the gait generation algorithms based on the finite states are implemented in an effective manner. The real bottleneck is actually the time delay in the robot controller converting from the binary commands to the actual mechanical

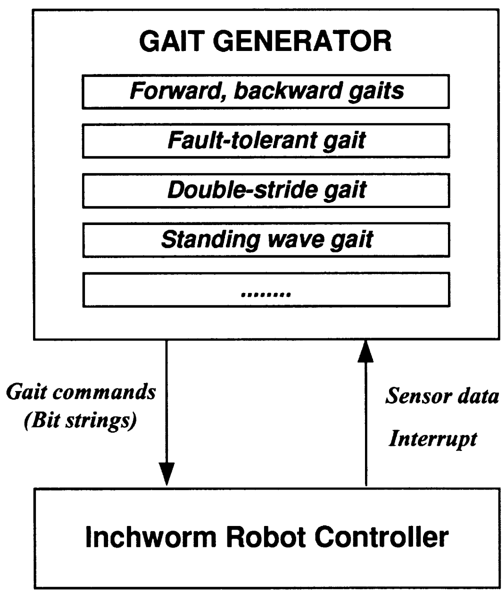

Fig. 12. Schematic gait generator.

movements of the actuators. The experiments on various types of gait generation are demonstrated in the video[11] and can also be accessed through at this website[12] in digital form. From gait generation point of view, the experiment results tally well with the computer simulation. However, as the solenoid actuators are slow in response to the bit string signals, programmed delay are amended to the actual robot control commands. Different gait patterns exhibit different dynamic characteristics as well. Because of the programmed delay and the sufficient gripping force, sing-stride gaits can be treated as quasi-static locomotion. When multi-stride gaits are initiated, the robot may slip due to the insufficient gripping force compared to the impulsive force produced by the extension and contraction of the actuators. Further investigation on the dynamic characteristics of different gaits will be necessary.

8. CONCLUSION

This article is our modest attempt to systematically investigate the gait generation problem for an n-segment inchworm-like robot. We model the inchworm robot as a finite automaton. The gait is a sequence of state transitions that follows the mechanics of inchworm locomotion. The gait generation problem becomes an exhaustive search on the directed graph defined by the automaton. The generation algorithms for various gaits are investigated, namely, single-stride gaits with backward motion and fault-tolerant features, and multi-stride gaits with standing wave feature. These algorithms can be applied to the control of various inchworm robot systems for inspection and material delivery purpose. Computer simulation of the gait generation is developed. An experimental inchworm robot platform is designed and constructed for further investigation of the dynamic characteristics of various gaits and their interaction with the environment. Further study on real-time on-line gait generation and planning will be carried out in the near future.

Acknowledgement

The authors would like to thank Mr. Edwin Ho of the School of Mechanical and Production Engineering, Nanyang Technological University for providing the design drawings of the experimental inchworm robot platform. This research is sponsored by National Science and Technology Board (NSTB) and Ministry of Education, Singapore under Academic Research Project JT/ARC 7-97.

References

1. M.C. Carrozza, L. Lencioni, B. Magnani, P. Dario, D. Reynaerts, M.G. Trivella and A. Pietrabissa, "A Microrobot for Colonoscopy", *Proceedings 7th Int. Sym. on Micro Machine and Human Science*, Nagoya, Japan (1996) pp. 223–228.
2. A.B. Slatkin and J.W. Burdick, "The Development of a Robotic Endoscope", *Proceedings IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Yokohama, Japan (1995) pp. 162–171.
3. T. Fukuda, H. Hosokal and M. Uemura, "Rubber Gas Actuator Driven by Hydrogen Storage Alloy for In-Pipe

Inspection Mobile Robot with Flexible Structure", *Proceedings IEEE Conf. Robotics Automation*, Cincinnati, OH, USA (1989) pp. 1847–1852.
4. G.S. Chirikjian and J.B. Burdick, "The Kinematics of Hyper-Redundant Robot Locomotion", *IEEE Trans. Robotics & Automation*, **11**, No. 6, 781–793, (1995).
5. G. Poi, C. Scarabeo and B. Allotta, "Traveling Wave Locomotion Hyper-Redundant Mobile Robot", *Proc. IEEE Robotics Automation*, Leuven, Belgium (1998) pp. 418–423.
6. S.D. Kelly and R.M. Murray, "Geometric Phases and Robotic Locomotion", *J. Robotic Systems*, **12**, No. 6, 417–431 (1995).
7. R.M. Murray, et al., "Trajectory Generation for Mechanical Systems with Application to Robotic Locomotion", *Robotics: The Algorithmic Perspective: WAFR 1998* (P. K. Agarwal, et al, edited.) (A. K. Peters Ltd., MA, USA, 1999) pp. 81–90.
8. Y. Gao, "Gait Generation and Planning for Inchworm-like Robot Locomotion", *Technical Report* (School of Mechanical & Production Engineering, Nanyang Technological University, 1998).
9. J.E. Hopcroft & J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. (Addison-Wesley, 1979).
10. T. Cormen, C. Leiserson and R. Rivest, *Introduction to Algorithms*. (Cambridge, MA, MIT Press, 1990).
11. I.M. Chen, S.H. Yeo, H.L. Ho and Y. Gao, "Gait Generation for Multi-Segment Inchworm-Like Robot Locomotion", (Video proceedings), *IEEE Int. Conf. Robotics Automation*, Detroit, MI, USA, (1999) p. 17.
12. "Simulation and experiment of inchworm robot locomotion", *Website URL:* http://www.ntu.edu.sg/mpe/Research/Groups/Mod_Robotics/index.htm

https://doi.org/10.1017/S0263574700003271 Published online by Cambridge University Press