

Planning Behaviors of a Modular Robot: an Approach Applying a Randomized Planner to Coherent Structure

Eiichi Yoshida, Haruhisa Kurokawa, Akiya Kamimura,
Kohji Tomita, Shigeru Kokaji
Intelligent Systems Institute, National Institute of
Advanced Industrial Science and Technology (AIST),
1-2 Namiki, Tsukuba, Ibaraki 305-8564 Japan
Email: module-robot@m.aist.go.jp

Satoshi Murata
Interdisciplinary Graduate School
of Science and Engineering,
Tokyo Institute of Technology,
4259 Nagatsuta-cho, Midori-ku,
Yokohama, Kanagawa 226-8502 Japan

Abstract—A Method for behavior planning is presented for a modular robot that applies a randomized planner to coherent structure of the robot. To cope with difficulty in planning of many degrees of freedom (DOFs) of a modular robot, coherent structure is introduced in terms of control system and robot configuration. As the control system, a simple phase synchronization mechanism is introduced that can control the robot with many DOFs with reduced number of parameters. Together with symmetrical configuration, this control system generates various dynamic motions. In order to plan the behaviors of the modular robot determined by the parameters of the control system, we adopt a randomized planner called rapidly exploring random trees (RRTs). This can benefit from a number of advantages of RRT planner, including simple implementation, uniform search, and applicability to a dynamic system with differential constraints. By exploring parameter space of the coherent control system, behaviors including dynamic motions can be planned. We have applied the proposed planner to M-TRAN modular robot to demonstrate the effectiveness of the proposed method through preliminary simulation results.

I. INTRODUCTION

Inspired by biological systems composed of cells and accelerated by recent progress in mechatronics, self-reconfigurable modular robots have been studied intensively in recent years. Initiated by earlier work [1], [2], [3], their flexibility, versatility, and fault-tolerance start attracting quite many engineers and scientists [4], [5], [6].

They are considered to be suitable especially for robots that can operate in unknown or unstructured environments, sometimes hostile to humans, through their adaptability. One of the examples of its application is a rescue robot. If external sensors are available, the robot can go into a narrow space, move around, and search for survivors in rubble by adapting its configuration to unstructured environments. Other examples include a planetary exploration robot and a plant inspection robot that are also required to go through environments with many obstacles. Those applications are becoming realistic as many types of three-dimensional (3-D) self-reconfigurable robots have been developed recently and their reliability and self-containedness have made a remarkable progress [7], [8].

In this paper we focus on planning behaviors of such modular robots, to decide what action those robots with many degrees of freedom (DOFs) take according to the

external environment. Research into the software of self-reconfigurable robots can be classified into two main domains. One is discrete reconfiguration planning to generate an appropriate sequence of configuration changes so that it can transform the robot from one configuration to another [9]–[14]. The other is continuous control that allows the robot to generate meaningful dynamic motions for the application [7], [15].

Although the reconfiguration planning has been intensively addressed, dynamic behavior planning is still less investigated except some work [7], [16], [24]. For the purpose of planning behaviors including dynamic motions with reduced parameters, we propose a method applying a randomized planning to “coherent” structures of a modular robot in terms of synchronized motions or structure symmetry.

As a control system, we adopt a simple synchronization mechanism in which state values are synchronized through differential equations. The parameters in this mechanism include several parameters such as phase difference and oscillatory amplification to determine the robots’ behavior. However, when choosing those parameters, search space is still very large even if they are not numerous. We use, therefore, a randomized planning method called rapidly-exploring random tree (RRT) [18] to realize effective search in those parameter space.

The organization of this paper is as follows. A simple phase synchronization mechanism is introduced in Section II and then it is applied to a modular robot M-TRAN in Section III. Planning method based on a randomized planner is presented in Section IV. Simulation results are shown to demonstrate the effectiveness of the proposed planning method in Section V before concluding the paper.

II. COHERENT STRUCTURE: PHASE SYNCHRONIZATION AND SYMMETRIC CONFIGURATION

One of methods for enabling a robot with many DOFs to generate meaningful motions is introducing coherency in its control system as well as its configuration. There are several ways to realize coherent control system. Neural network that generates oscillatory signals is one of well-known approaches, like central pattern generator (CPG)

[19], [20]. In this paper, for the simplicity of implementation, we adopt a simple phase synchronization mechanism.

A. Phase Synchronization Mechanism as Control System

To illustrate the mechanism, let us explain the simplest example, composed of only two connected elements 1, 2. The phase variable θ_i denotes the internal state of each element. In this example, both elements try to have constant velocity ω in such a way that θ_1 of element 1 is always greater than θ_2 of element 2 by phase difference ϕ . This is described by differential equations:

$$\begin{aligned}\dot{\theta}_1 &= \omega - k(\theta_1 - \theta_2 - \phi) \\ \dot{\theta}_2 &= \omega - k(\theta_2 - \theta_1 + \phi)\end{aligned}\quad (1)$$

where k is a positive gain coefficient. By solving this equation, we can verify that the difference $\theta_1(t) - \theta_2(t)$ converges to ϕ with positive k .

This mechanism can be extended to n elements:

$$\dot{\theta}_i = \omega - k \sum_{j=0}^n (\theta_i - \theta_j - \phi_{ij}) \quad (2)$$

where $\phi_{ij} = -\phi_{ji}$. If there are not loops in the element connections, the phase difference $\theta_i(t) - \theta_j(t)$ between element i and j converges to the given value ϕ_{ij} [21]. This synchronization mechanism has also an advantage of distributed implementation into modular robots.

In this paper we only address the cases without loops for simplicity. For the cases including loops, the readers are referred to a method that deals with this problem [22].

We will apply this simple mechanism to generate various dynamic motions of modular robots. Since the output of (2) is linearly increasing along the time, we use a sinusoidal function to generate an oscillatory movement like

$$a_i(t) = \beta_i + \alpha_i \sin \theta_i(t) \quad (3)$$

where α_i and β_i denotes the amplification and offset of oscillation respectively.

B. Symmetric Robot Configuration

By assigning the synchronization element to each actuator, a modular robot is expected to generate diverse coordinated actuator outputs for dynamic motion. However, if the control system does not have a good accordance to the robot structure, it is difficult for the robot to generate meaningful motions.

For this reason, another coherency is introduced in the robot structure, namely symmetrical configurations in our case. Even if the control system has coherency, a robot that has some irregular configuration can hardly be controlled. Looking at the nature, many animals indeed have symmetric form to make efficient dynamics motions.

Among many possibilities, symmetric configurations can be used to apply the phase synchronization control mechanism to realize the dynamic motions such as gait patterns. Moreover, not only changing the dynamic motions, but they can choose different configurations according to the application, a four-legged robot or a snake-like robot. This is one of the major advantages of modular robots as mentioned earlier.

III. APPLYING THE COHERENT STRUCTURE TO M-TRAN MODULAR ROBOT

In this section, the introduced coherent structure, in control system and robot configuration, is applied to a modular robot platform M-TRAN.

A. Self-Reconfigurable Modular Robot M-TRAN

The modular robot M-TRAN (Modular TRANSformer) has been developed in AIST. Its major feature is capability of realizing both self-reconfiguration and dynamic motions in three dimensions by a collection of identical modules. The module has a simple bipartite structure, each of which rotates about a parallel axis by geared motors and has three magnetic connecting faces as shown in Fig. 1. Each module is self-contained with embedded controller circuit board and battery. Figure 2 shows the newest hardware model "M-TRAN II." For details on hardware, please see other references [7], [17].

B. Coherent Control and Configuration for M-TRAN

As a symmetric robot structure, we adopt a configuration composed of nine modules as shown in Fig. 3. By assigning the phase synchronization mechanism to each module's actuator with appropriate parameters, the resulting oscillatory output enables the robot to generate efficient locomotion.

The symmetry of the robot structure relieves us of controlling all the parameters individually. Instead, the same parameters can be repeatedly used to symmetrically corresponding actuators and synchronization connections. For the configuration shown in Fig. 3, the parameters can be considerably reduced since there are two symmetry axes. Figure 4 shows how the same parameters are applied symmetrically. In this figure, circles and dotted lines denote oscillatory elements and synchronization connection between them respectively. The arrow is defined the direction such that ϕ_i is the phase difference from outgoing element to incoming one.

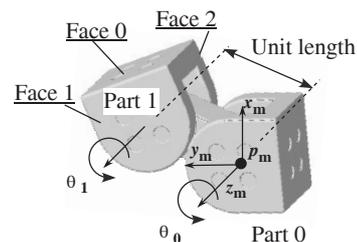


Fig. 1. A module of M-TRAN.

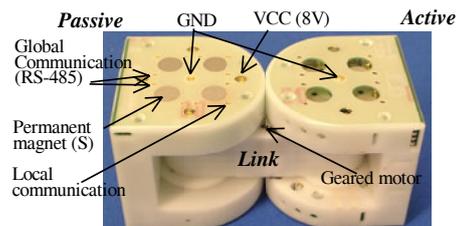


Fig. 2. A hardware module of M-TRAN II.

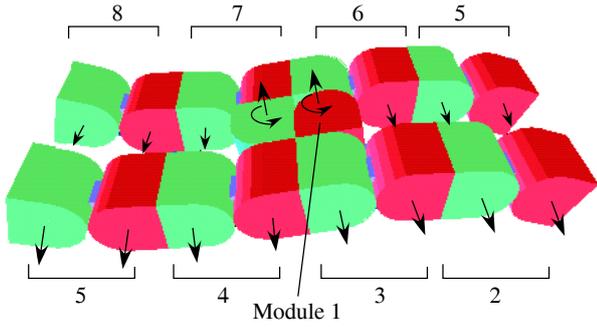


Fig. 3. A coherent configuration of M-TRAN modules

There exist some exceptions of the symmetrical parameter assignment, regarding Module 1 that is in the center of the structure. Its amplification and offset of the oscillation α_i , β_i ($i = 5, 6$) and phase difference ϕ_5 are need to be controlled separately. Another exception is the value ϕ_4 for the connection to Module 1, which is applied in different direction from the symmetric connection.

The modular robot with this configuration can realize different structures for locomotion using this parameter assignment. One is a flat configuration whose corresponding motion snake-like wave motion (Fig. 5). Another motion is four-leg configuration that requires certain gait pattern to move, using parameters illustrated in Fig. 6. Motion direction can be changed by setting non-zero values to α_5 , α_6 and setting offsets β_5 , β_6 that allow the Module 1 to generate rotational motion along the vertical axis as shown in Figs. 7 and 8. In general, the opposite direction of motion can be generated by reversing ϕ_i values. In all the examples, the gain k is set to a constant value 10 and we assume that the actuators are controlled by velocity.

Note that a single synchronization scheme can realize those very different locomotion modes by changing parameters based on the proposed coherent structure. In the next section, we will describe how those behaviors can be planned using a randomized method.

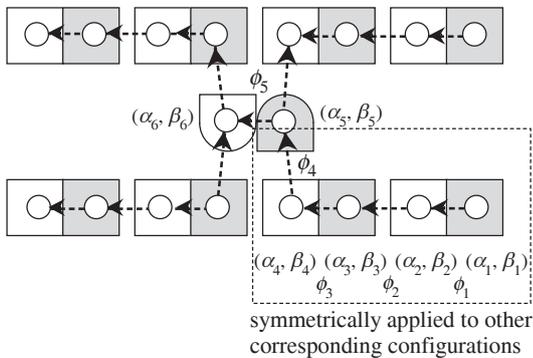


Fig. 4. Assignment of phase synchronization parameters to symmetric robot structure

IV. BEHAVIOR PLANNING USING A RANDOMIZED METHOD

There are several ways to derive behaviors determined by a number of parameters. A heuristic planning method for static motion is proposed to enable a legged robot to move rough terrain [23]. A hormone-based behavior control is another alternative [16]. Støy et al. proposed a control model for chain-type modular robots using coupled oscillators [24]. As a more general scheme, gradient method has been proposed to optimize those phase synchronization mechanism [25].

Kamimura et. al use a genetic algorithm (GA) to obtain CPG parameters for locomotion [7]. This pattern generation method is more dedicated to real-time control and adaptation according to external stimuli, where many parameters of CPG model should be regulated. In the planning phase, however, a simplified control model is more preferable than direct usage of complex model of CPG since it is important to reduce the number of parameters to explore.

In this paper, focusing on planning dynamic behavior using a simple control model based on coherent structure, a randomized planning method called rapidly-exploring random trees (RRTs) is introduced. This randomized planner allows incremental and reactive planning based on a simple algorithm. To apply this randomized method, we assume that the robot has knowledge about local environment and its goal through external sensor capacity.

A. Rapidly-Exploring Random Trees (RRTs)

RRTs have been proposed by LaValle and Kuffner as a randomized motion planning method [18]. The idea is to incrementally construct search trees that attempt to explore the state space rapidly and uniformly. It has been proven that this method is probabilistic complete; namely desired path will be found eventually as the number of vertex becomes infinity. Since it has such advantages as simplicity of implementation for exploring many DOFs and possibility of including differential constraints, we adopt this method for dynamic behavior planning.

The basic algorithm is shown in Fig. 9, which makes a tree \mathcal{T} grow in the space of configuration q of a moving entity, which can be an object, a vehicle, or a robot. At each step, after generating a random configuration node q_{rand} , the function $\text{EXTEND}(\mathcal{T}, q_{rand})$ is called to extend the tree by using $\text{NEW_CONFIG}(q, q_{near}, q_{new})$ function as illustrated in Fig. 10. $\text{NEW_CONFIG}()$ selects a node q_{near} nearest to q based on given metric, and then generates a new node q_{new} that advances to q to grow the tree \mathcal{T} . In this way, the tree rapidly explores the state space through biased search of large unexplored region.

Several RRT-based motion planners have been proposed according to the problem. For example, RRT-Connect [26] is a bidirectional planner that explores RRTs from initial and goal position in environments, possibly with obstacles. The search can be accelerated using bidirectional planning.

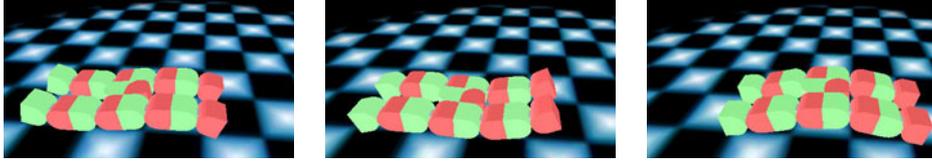


Fig. 5. Snake-like motion: $\alpha_1 \sim \alpha_4 = 10^\circ$, $\alpha_5, \alpha_6 = 0^\circ$, $\beta_1 \sim \beta_4 = 0^\circ$, $\beta_5 = -90^\circ$, $\beta_6 = 90^\circ$, $\phi_1 \sim \phi_3, \phi_5 = 30^\circ$, $\phi_4 = 0^\circ$, $\omega = 180^\circ$

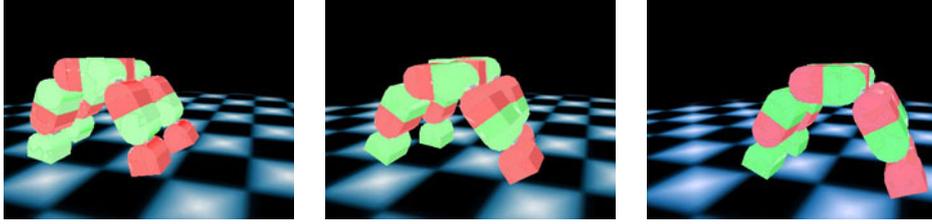


Fig. 6. Legged locomotion: $\alpha_1 \sim \alpha_4 = 5^\circ$, $\alpha_5, \alpha_6 = 0^\circ$, $\beta_1 = 0^\circ, \beta_2 = 10^\circ, \beta_3 = 20^\circ, \beta_4 = 40^\circ, \beta_5 = -90^\circ, \beta_6 = 90^\circ$, $\phi_1, \phi_3, \phi_4 = 90^\circ$, $\phi_2, \phi_5 = -90^\circ$, $\omega = 180^\circ$

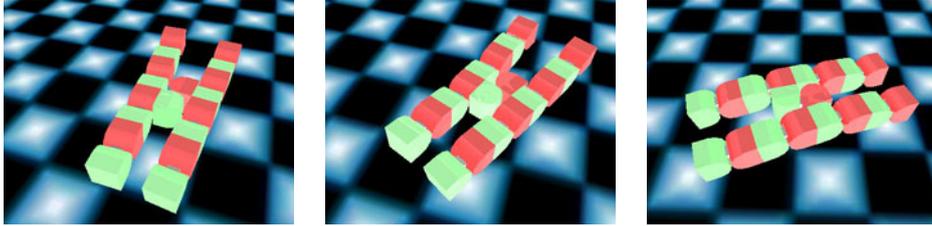


Fig. 7. Snake-like motion changing heading direction: $\alpha_1 \sim \alpha_6 = 5^\circ$, $\beta_5 = -80^\circ, \beta_6 = 80^\circ, \phi_4 = 20^\circ$, and other parameters same as in Fig. 5

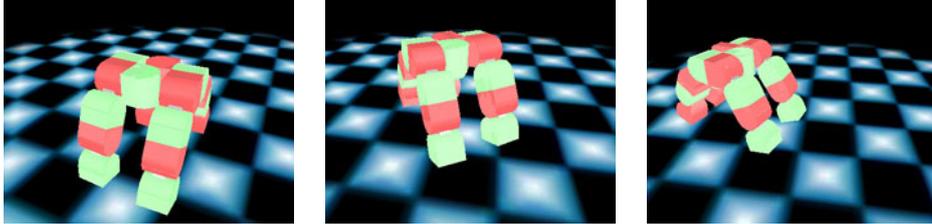


Fig. 8. Legged locomotion changing heading direction: $\alpha_5, \alpha_6 = 5^\circ, \beta_5 = -80^\circ, \beta_6 = 80^\circ$, and other parameters same as in Fig. 6

```

BUILD_RRT( $q_{init}$ )
1   $\mathcal{T}.init(q_{init})$ 
2  for  $k=1$  to  $K$  do
3     $q_{rand} \leftarrow \text{RAND\_CONFIG}()$ ;
4     $\text{EXTEND}(\mathcal{T}, q_{rand})$ ;
5  Return  $\mathcal{T}$ ;

EXTEND( $\mathcal{T}, q$ )
1   $q_{near} \leftarrow \text{NEAREST\_NEIGHBOR}(q, \mathcal{T})$ ;
2  if  $\text{NEW\_CONFIG}(q, q_{near}, q_{new})$  then
3     $\mathcal{T}.add\_vertex(q_{new})$ 
4     $\mathcal{T}.add\_edge(q_{near}, q_{new})$ 
5    if  $q_{near} = q$  then
6      Return Reached;
7    else
8      Return Advanced;
9  Return Trapped;

```

Fig. 9. Basic Algorithm of RRT extension

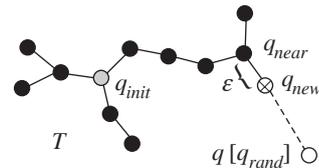


Fig. 10. Function $\text{EXTEND}(\mathcal{T}, q)$

B. Applying RRTs to Modular Robot's Behavior Planning

Now RRT planner is applied to modular robot's behavior planning based on phase synchronization mechanism. As mentioned in III-B, synchronization mechanism can be described using parameters ϕ_{ij} , ω , α_i and β_i for the configuration shown in Fig. 3.

As the robot generates motions according to those parameters, it causes oscillatory motion that brings to the robot to a different position. This is a differential constraint where the relationship between control input and resulting configuration must be specified, since simple interpolation between

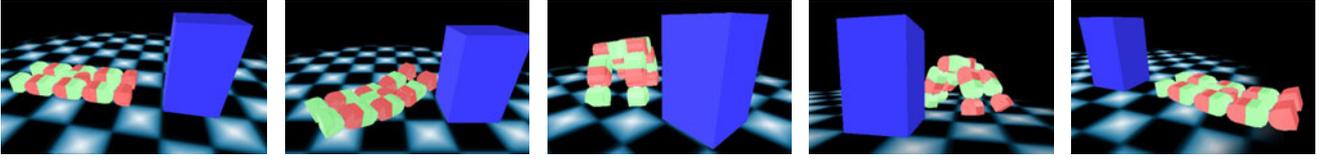


Fig. 11. Simulation results (1): avoiding an obstacle in front

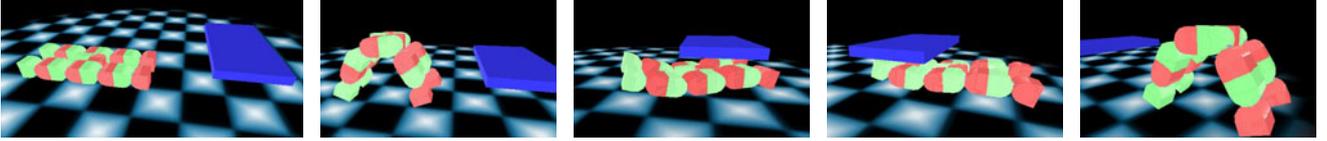


Fig. 12. Simulation results (2): going under an obstacle

configurations does not apply. In the algorithm Fig. 9, the configuration q is replaced by the state x that describes robot's current state [27]. Also, $\text{NEW_CONFIG}(q, q_{near}, q_{new})$ is replaced by $\text{GEN_STATE}(x, x_{near}, x_{new}, \Delta t, Inputs)$ that generates next state x_{new} advancing to given state x from its nearest neighbor x_{near} in the tree, by selecting control input from the possible set $Inputs$.

In our example, the state x includes the position and orientation of the whole robot represented by a reference frame fixed in the Module 1 as well as each module's the output angles, position, orientation, and velocity. To compute the distance between the states x , we adopt the distance between these representative positions and orientations as the metric.

The RRT planner explores in the space of those synchronization parameters as the above $Inputs$. Then the next state x is computed as a result of dynamic motion described by the control system based on the phase synchronization mechanism. Usually, those inputs are selected in such a way that they determine directly the state of robot, like position or velocity of each actuator. However, direct search of those input results in meaningless motions in most cases. In contrast, by applying RRT to those "indirect" parameters through coherency of control system and robot structure, we expect our goal of dynamic behavior planning can be achieved. On the other hand, this causes a disadvantage of heavy computation of next state in $\text{GEN_STATE}()$. Currently we must calculate the next state using a dynamics simulator that is computationally expensive. To accelerate planning, this needs to be substituted simple and fast solver for dynamics and collision detection.

V. SIMULATION RESULTS

In this section we show several preliminary results of simulations conducted based on the behavior planning scheme presented in the previous sections.

The state of the robots are described as (X, Y, Θ) using its position X, Y on the plane and orientation Θ of Module 1. Given its goal state, the planner attempts to find a sequence of synchronization parameters of the modular robot to reach the goal. In the simulations, the input parameter are selected from following sets shown in Table I. Some of resulting motions of those parameters are

TABLE I
INPUT PARAMETER SETS FOR PHASE SYNCHRONIZATION (DEGREES)

amplification	$\alpha_1 \sim \alpha_4$	5, 10, 20
	α_5, α_6	0, 5
offset	$\{\beta_1, \beta_2, \beta_3, \beta_4\}$	$\{0, 0, 0, 0\}, \{0, -5, -5, -5\}, \{0, 10, 20, 40\}$
	$\{\beta_5, \beta_6\}$	$\{-90, 90\}, \{-80, 80\}$
phase difference	$\phi_1 \sim \phi_3, \phi_5$	$\pm 30, \pm 60, \pm 90$
	ϕ_4	$0, \pm 20$
angular velocity	$\{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$	$\{\mp 90, \pm 90, \mp 90, \mp 90, \pm 90\}$
	ω	140, 180, 220

given in Figs. 5 to 8. Although small numbers of values are used to reduce the search space, this simple combination turned out to be sufficient to generate various motions.

The simulations are implemented using Vortex dynamics simulation library [28] and MSL library [29] for RRT planner. Collision detection is implemented in both libraries. According to each input, the next state (X, Y, Θ) after time $\Delta t = 2$ (sec) are calculated by the dynamics simulator. We use relatively large Δt to allow the robot to make oscillatory motion for a certain period. Then the RRT planner explores this state space to reach the goal. Here, goal-biased RRT planner is used as the planner. In the initial state, the robot is at $(0, 0, 0)$ and goal state is $(18, 0, 0)$, namely moving to a given position with the same orientation, on in a plane that ranges from $(-6, -12)$ to $(24, 12)$. The unit length is shown in Fig. 1. Two simulations are conducted where there are obstacles with different shape at different positions.

Figures 11 and 12 show some snapshots of robot motion of the simulation results. In Fig. 11, the robot moves around the obstacle by changing its moving direction to reach the goal position. In the next simulation, first the robot uses the four-leg locomotion to advance in free space. Although it could go around the obstacle, the goal-biased RRT planner found the motion to go under the obstacle with snake-like locomotion before finally restoring the four-leg locomotion to reach the goal. Those preliminary results demonstrated the effectiveness of the proposed method.

Figure 13 shows the explored tree in the state space projected to X-Y plane in the simulations, where the thick

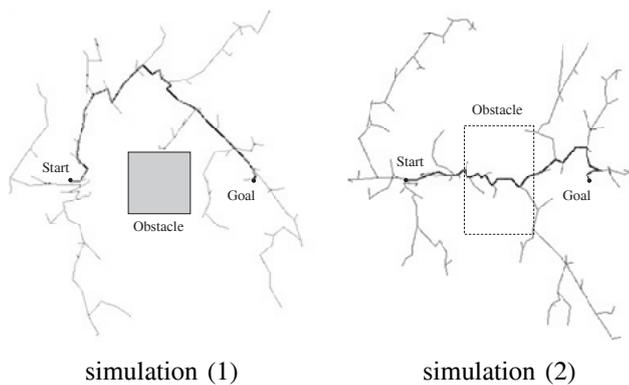


Fig. 13. Explored RRT projected on X-Y plane for simulations

line is the path from the initial position to the goal. Note that the states are actually more smoothly connected than it appears because they are plotted only at every Δt .

The computation took several minutes in both cases using Pentium M processor with 1.4GHz. As mentioned earlier, improvements will be addressed using alternative fast solver for dynamics and collision detection in future developments.

VI. CONCLUSIONS AND FUTURE WORK

We presented a behavior planning of modular robot with coherent structure using a randomized planning method. As a coherent structure, a simple phase synchronization mechanism was adopted for symmetric robot configurations. This control structure enables a modular robot with many DOF to generate various dynamic motions with reduced number of control parameters. A randomized method called rapidly-exploring random tree (RRT) was employed to plan the robot's dynamic behaviors. This method is capable of planning a dynamic system with differential constraints based on simple implementation. The proposed method was implemented on a coherent structure of a modular robot platform M-TRAN. The preliminary simulation results showed the feasibility of the proposed method.

Future work includes such issues as integrating self-reconfiguration process, selection of coherent structures and control parameters, and more efficient implementation. Since the RRT planning scheme can include both discrete planning and differential constraints, the first issue is important in the next stage of development. This is related to the second issue, as several coherent structures can be possible according to the application. Concerning the control parameter, the control input sets are currently defined empirically. To improve the applicability of the method, improvement toward automatic acquisition of those inputs, through learning or evolutionary computation like in [7], will be addressed. This evolutionary motion generation can be combined with the proposed planning method by using evolved motion patterns as behavior elements to be planned. Efficient implementation is also to be addressed so that to reduce the planning time. Likewise, self-learning of dynamic property through interaction with environments is also a challenging issue for implementation in real robots.

REFERENCES

- [1] T. Fukuda and S. Nakagawa: Approach to the dynamically reconfigurable robotic system, *Journal of Intelligent and Robot Systems*, **1**, 55–72, 1998.
- [2] S. Murata, et al: Self-assembling machine, *Proc. 1994 IEEE Int. Conf. on Robotics and Automation*, 441–448, 1994.
- [3] G. Chirikjian, et al: Evaluating Efficiency of Self-Reconfiguration in a Class of Modular Robots, *J. Robotic Systems*, **12-5**, 317–338, 1996.
- [4] *Autonomous Robots Special issue on self-reconfigurable robots*, **10-1**, 2001.
- [5] *IEEE/ASME Trans. on Mechatronics Special issue on reconfigurable robots*, **7-4**, 2002.
- [6] *Science Shape Shifters Thread a Daunting Path Toward Reality*, 301, 754–756, August, 2003.
- [7] A. Kamimura, et al: Automatic Locomotion Pattern Generation for Modular Robots, *Proc. 2003 IEEE Int. Conf. on Robotics and Automation*, 2003.
- [8] M. Yim, et al: Modular Robots – *IEEE Spectrum*, **39-2**, 33–34, February 2002.
- [9] K. Kotay and D. Rus: “Motion Synthesis for the Self-Reconfigurable Molecule,” *Proc. 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 843–851, 1998.
- [10] E. Yoshida, et al: A Distributed Method for Reconfiguration of 3-D homogeneous structure, *Advanced Robotics*, **13-4**, 363–380, 1999.
- [11] C. Ünsal, et al.: A modular self-reconfigurable bipartite robotic system: Implementation and Motion Planning, *Autonomous Robots*, **10-1**, 23–40, 2001.
- [12] M. Yim et al.: Distributed Control for 3D Metamorphosis, *Autonomous Robots* **10-1**, 41–56, 2001.
- [13] D. Rus and M. Vona: Crystalline Robots: Self-reconfiguration with Compressible Unit Modules, *Autonomous Robots*, **Vol.10-1**, 107–124, 2001.
- [14] E. Yoshida, et al.: A Self-Reconfigurable Modular Robot: Reconfiguration Planning and Experiments, *Int. J. Robotics Research*, **21-10**, 903–916, 2002.
- [15] Y. Zhang et al.: Scalable and Reconfigurable Configurations and Locomotion Gaits for Chain-type Modular Reconfigurable Robots *Proc. 2003 IEEE Int. Conf. on Computational Intelligence in Robotics and Automation (CIRA2003)*, 2003.
- [16] W.-M. Shen, et al.: Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots, *IEEE Transactions on Robotics and Automation*, **18-5**, 700–712, 2002.
- [17] S. Murata, et al.: M-TRAN: Self-reconfigurable Modular Robotic System, *IEEE/ASME Transactions on Mechatronics*, **7-4**, 431–441, 2002.
- [18] S. LaValle and J. Kuffner: Rapidly-Exploring Random Trees: Progress and Prospects, In *Algorithmic and Computational Robotics: New Directions*, 293–308, A K Peters, Wellesley, 2001.
- [19] G. Taga: A model of the Neuro-Musculo-Skeletal System for Human Locomotion II / Real-Time Adaptability under Various Constraints,” *Biolog. Cybern.*, **73**, 113–121, 1995.
- [20] H. Kimura, et al.: Realization of dynamic walking and running of the quadruped using neural oscillator, *Autonomous Robots*, **7-3**, 247–258, 1999.
- [21] H. Yamaguchi, et al.: A distributed control scheme for multiple robotic vehicles to make group formations, *Robotics and Autonomous Systems*, **36**, 125–147, 2001.
- [22] S. Kokaji, et al.: Clock synchronization algorithm for a distributed Autonomous System, *J. Robotics and Mechatronics*, **8-5**, 317–328, 1996.
- [23] C. Eldershaw and M. Yim: Motion planning of legged vehicles in an unstructured environment, *Proc. 2001 Int. Conf. on Robotics and Automation*, 3383–3389, 2001.
- [24] H. Støy, et al: Using Role Based Control to Produce Locomotion in Chain-Type Self-Reconfigurable Robots *IEEE/ASME Trans. on Mechatronics*, **7-4**, 410–417.
- [25] H. Yuasa and M. Ito: Coordination of Many Oscillators and Generation of Locomotory Patterns, *Biol. Cybern.*, **63**, 177–184, 1990.
- [26] J. Kuffner and S. LaValle: RRT-Connect: an efficient approach to single-query path planning, *Proc. 2000 IEEE Int. Conf. on Robotics and Automation*, 995–1001, 2000.
- [27] S. LaValle and J. Kuffner: Randomized kinodynamic planning, *Proc. 1999 IEEE Int. Conf. on Robotics and Automation* 473–479, 1999.
- [28] <http://www.cm-labs.com>
- [29] <http://msl.cs.uiuc.edu/msl>