2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Modular Robot Locomotion based on a Distributed Fuzzy Controller: The Combination of ModRED's Basic Module Motions

José Baca[1], Prithviraj Dasgupta[1], S.G.M. Hossain[2], and Carl Nelson[2]

***Abstract*— We describe a distributed and autonomous technique for dynamic gait adaptation for a chain-type, modular self-reconfigurable robot (MSR) using a fuzzy logic based, closed-loop controller. To maneuver itself, each module of the MSR is provided with a set of basic or fundamental gaits within a gait control table(GCT). A relevant problem in locomotion of a chain-type MSR is how to coordinate the gait of the individual modules with each other so that the desired locomotion of the MSR can be achieved. To address this problem, our proposed controller maps the inputs from the sensors of each module to an appropriate gait for the module determined from the goal and position of the module in the configuration, using a fuzzy technique. An inertial measurement unit (IMU) is used to close the loop between the goal and the module. We have verified the operation of our controller on a simulated 3-D model of an MSR called ModRED within the Webots robot simulator and also implemented it on the physical ModRED MSR. Our results illustrate that our controller can successfully adapt ModRED's locomotion by dynamically combining basic gaits from the individual modules in the configuration, regardless of the number of modules in the configuration and in the presence of noisy sensor inputs.**

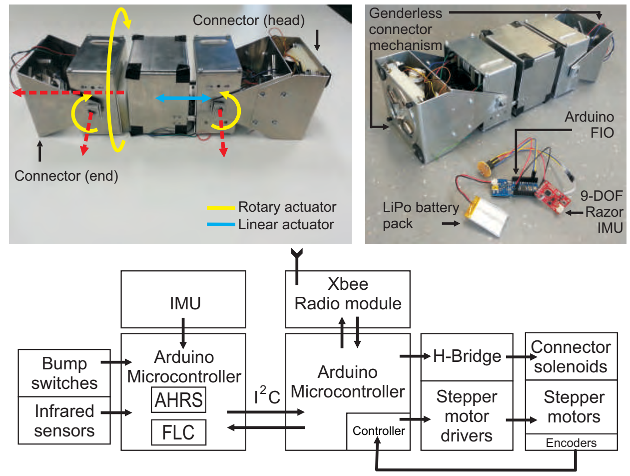

Fig. 1. Mechanical design and control architecture of a ModRED module.

I. INTRODUCTION

Modular self-reconfigurable robots (MSRs) are robotic systems composed of individual modules that can be connected with each other to form different shapes or configurations. This characteristic allows an MSR to be highly dexterous by dynamically adapting its shape and/or locomotion, so that it can continue maneuvering in an environment even if it gets impeded by obstacles [1]. MSRs' high dexterity offers significant advantages during robotic missions in unstructured environments such as the Lunar or Martian surface where the MSR should adapt its shape/locomotion autonomously to continue its maneuver and operations. A fundamental problem in MSR locomotion is to coordinate the movement of the individual modules so that the desired movement of the MSR can be achieved successfully. This problem is non-trivial as each module has to perform different yet coordinated movements at each time step to effect the MSR's locomotion. Also, it is not very straightforward to design model-based controllers for MSR's as they can form simple and complex kinematic structures with many degrees of freedom. Previous researchers have addressed this problem by using different techniques such as inter-module message passing and learning-based mechanisms [2], [3]. Complementary to these approaches, in this paper, we describe a lightweight, goal-driven technique for locomotion of a chain-type MSR in fixed configuration using a fuzzy logic based controller (FLC).

Fuzzy logic has proven to be a convenient tool for decision making in complex scenarios while effectively handling real world uncertainty and knowledge representation [4]. Fuzzy logic-based control is well suited for use with the major limitations that characterize most MSRs such as inexpensive and noisy sensors, low-resolution analog-to-digital converters, and 4-bit or 8-bit micro-controllers with limited computational capability. Additionally, fuzzy control is easily extensible - it can be used to add new behaviors on an MSR by adding an extra layer of intelligence through fuzzy rules to improve an MSR's existing control technique. The design of our approach considers five main aspects; the limited resources of a module (*e.g.,* limitation of sensors, computation capability, size, etc.), the position of the module in a chain-type configuration, the type of locomotion that the robot has to perform (*e.g.,* inchworm, differential drive, snake, etc.), the combination of basic motions from a single module that manifests the desired movement of the chain-type configuration, and, finally, the autonomy of each module's controller, *i.e.,* each module in the configuration decides its appropriate motion on its own. Our proposed FLC operates within the micro-controller of each module; it uses the sensory inputs from the module, the module's

*This work was done as part of the ModRED project which is supported by NASA EPSCoR grant no. NNH11ZHA003C

[1]J. Baca is a post-doctoral researcher and P. Dasgupta is an associate professor with the Computer Science Dept., Univ. of Nebraska at Omaha {jbacagarcia,pdasgupta}@unomaha.edu

[2]S.G.M. Hossain is a Ph.D. student and C. Nelson is an associate professor with the Mechanical and Materials Eng. Dept., Univ. of Nebraska - Lincoln. smgmamur@yahoo.com, cnelson5@unl.edu

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

connections with other modules and the MSR's current goal to generate the motion of the module. This motion is selected from a library of gait control tables (GCTs) representing basic motions of a single module. We have used an inertial measurement unit (IMU) as the sensor that closes the loop between the goal, the MSR, and the environment. We have verified the operation of our controller on a simulated model of an MSR called ModRED (Fig. 1) within the Webots robot simulator and also on the physical ModRED MSR. Our experimental results show that our proposed controller can be implemented successfully to adapt the locomotion of a chain-type MSR, within the limited computational resources of the individual MSR modules.

II. RELATED WORK

Over the past two decades MSRs have matured from proof-of-concept models to accurately simulated and physically implemented systems that demonstrate their fundamental capabilities [5], [6], [7]. Recently, several researchers have focused on techniques for adapting the gait of an MSR. In one of the earliest works on this topic, Kamimura *et al.* [8] described a central pattern generator that adapts the gait of the robot in a fixed configuration. More recently, Christensen *et al.* [9], have described a reinforcement learning technique for learning best actions on a single module and for learning set points in gait control tables for multi-module configurations. The efficacy of the learning technique is demonstrated by adapting the gait to continue locomotion following module failures on simulated models of ATRON and M-TRAN modular robots. Another learning technique called surprise-based learning [2] can adapt to component failures on the Superbot modular robot using a rule-based learning technique. In [10], the authors use a genetic algorithm within a gene-regulatory network framework to learn parameters corresponding to efficient locomotion using different gaits that are susceptible to failure on the simulated ATRON robot. Park and Yim [3] have proposed a gait adaptation technique where the modules in the CKBot system use a message passing algorithm implemented via IR-based inter-module communication and arrive at a consensus on the overall gait for the robot using a majority-based vote. A consensus-based framework for collective locomotion of modules has also been proposed in [11] and verified for locomotion of a 2D robot. Recently, fuzzy control has been used on a modular reconfigurable serial manipulator for an industrial application [12], [13], [14]. Complementary to previous approaches, the main contribution of this work is the use of fuzzy logic as closed-looped and distributed controller for dynamic gait selection on MSRs. It considers the position of the module in a chain-type configuration, the current environmental conditions and operational status perceived through the robot's sensors. In addition, by combining basic GCTs, the locomotion of complex configurations can be achieved efficiently.

III. MODULAR ROBOT: MODRED

ModRED (*Modular Robot for Exploration and Discovery*) is a homogeneous modular robot system that is suitable for efficient maneuver over unstructured surfaces such as extra-terrestrial environments [15]. Each of the ModRED modules has 4 DOF - 3 rotational and 1 prismatic. The 4 independent DOF are characterized by specific ranges to meet the requirements for locomotion and reconfiguration capabilities.

A. Control System Architecture

As an autonomous system, modular robots must be capable of sensing their environment and acting on this information for task completion purposes. Each of the ModRED modules is equipped with necessary electronics to give them such autonomy, as shown in Figure 1. Each module performs the computation and control tasks using two Arduino Fio (ATmega328P) microcontrollers. For powering the overall system including all the sensors, actuators and microcontrollers, a rechargeable lithium-polymer battery pack is used. Each module is equipped with an XBee modem directly connected to the Arduino board to enable wireless communication among the modules. For navigation through unstructured environments, the module is equipped with one 9-DOF Razor Inertial Measurement Unit (IMU) which incorporates three sensor types; an ITG-3200 (triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5843 (triple-axis magnetometer). The fusion of data from these sensors allows to estimate the orientation of the module due to the Attitude Heading Reference System (AHRS). In addition to this, an array of infrared (IR) sensors (with a range of 40mm-300mm) is used for proximity sensing and local localization strategies. For obstacle detection purposes, and to ensure successful docking, bump switches are incorporated in the front/rear faces of the docking brackets.

B. Gait Control Tables from a Single Module

To maneuver across unstructured terrains, ModRED's module offers unique locomotion types due to its kinematics design. Complex locomotion types can be created by combining basic motions from a single module. For simplicity in representing each of the gait control tables (GCTs), in Figure 2 a triangle is placed at the head and end parts of the module, representing the rotational DOF. A vertical line in the center of the module represents the contraction of the translational DOF. Two parallel vertical lines represent the extension of the translational DOF. For instance, GCT 1 is the neutral position of the module, and GCT 2 and GCT 3 make the module rotate to its left and right, respectively. GCT 4 uses the rotational and translational DOF to generate an inchworm-type locomotion. Similarly, GCT 5 reproduces the same inchworm-type locomotion when the module is facing up. GCT6 and GCT 7 make the module rotate its head down and up. GCT8 and GCT9 make the module rotate its end down and up. Finally, GCT10 uses the translational DOF to generate extension and contraction of the module. In this work, we consider that the above GCTs produce the basic

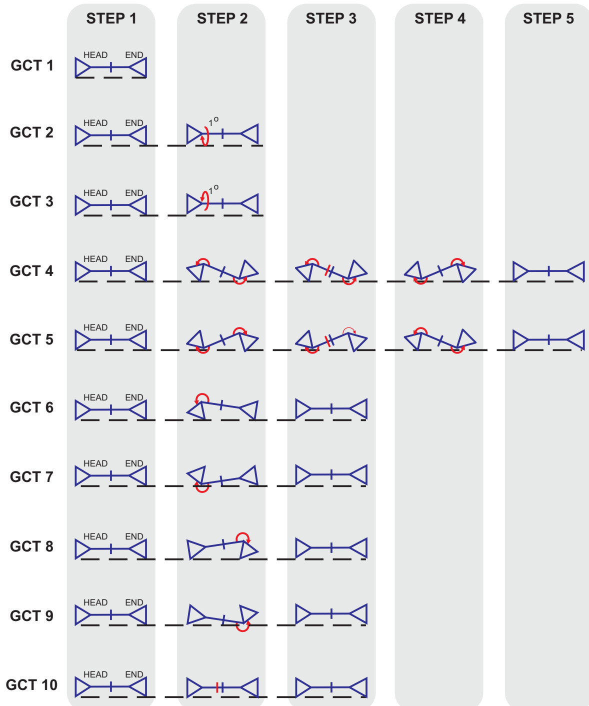

Fig. 2. Basic GCTs for locomotion for a single module explained using step by step actuations of the four DOFs.

locomotion types for a single module. When more modules are added, the desired locomotion is achieved with the help of this basic set of GCTs.

C. Module's position in a chain-type configuration

Since MSRs have the ability to dock and undock modules, in a chain-type configuration the position of the module can be classified into three main cases, as shown in Figure 3.

The first case where the module is not attached to other modules, hence, there are no restrictions to use the set of basic GCTs; a second case where the module is attached to a second module (it has to consider if its head- or end-connector is attached to the second module); a third case where the module is placed between two modules (here, the head- and end-parts' motion is limited by weight and motion-type of the overall configuration).

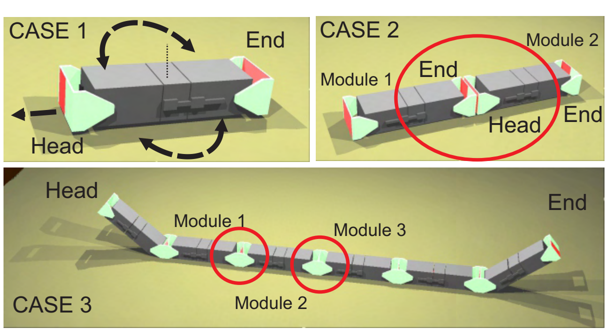

Fig. 3. In a chain-type configuration, the location of the module can be classified in three main cases.

IV. FUZZY CONTROLLER FOR GAIT CONTROL TABLE SELECTION

MSR modules are often confined to small size and weight, which sets hard limits on the on-board resources. The main idea in this work is to efficiently use the limited resources on a single module to develop the fuzzy logic controller (FLC) that enables ModRED to perform its task. The task we consider for ModRED is to reach a goal (direction: orientation and displacement), while maintaining its current configuration but adapting the robot's gait, effected through the gait of its individual modules, dynamically. The proper execution of this task is essential for generic tasks such as self-reconfiguration or displacement in fixed configurations during navigation tasks. Each module of ModRED runs the FLC independently to decide its movements for performing the assigned task. The different components of the FLC are shown in Figure 4.

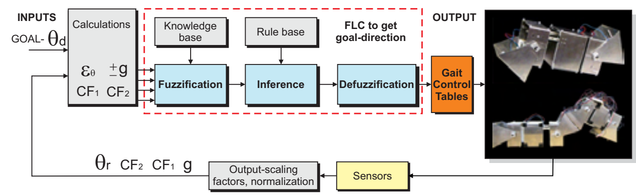

Fig. 4. Fuzzy Logic Controller for MSR with goal-directed behavior.

A. Input and Output Variables

Our FLC uses four inputs obtained from the sensors on the module. For adjusting the orientation of robots in extra-terrestrial environments that are devoid of terrestrial satellite information and earth's gravitational field, inertial measurement units (IMUs) are an attractive choice [16]. In our system, we use the IMU sensor values to calculate a module's absolute orientation θ_r and the direction of gravity $\pm g$ that identifies if a module is facing down or facing up.

The first input to the FLC is the angular error between the robot's desired orientation or *goal direction* and the robot's current heading direction, denoted by $\varepsilon_\theta = \theta_d - \theta_r$. The second input is the gravity direction ($\pm g$). The third and fourth inputs indicate the status of the connector faces of the module *i.e.,* if the connector on the head part of the module (CF_1) or the connector on the end part of the module (CF_2) are connected to another connector. The output signal from the FLC selects the corresponding GCT from Figure 2.

B. Fuzzification

The fuzzification process maps each input parameter to a set of fuzzy symbols corresponding to different ranges of the input parameter's value. The fuzzy sets for our input variables are presented in Table I. The membership functions for the different input parameters are shown in Figure 5 (top three subfigures).

TABLE I

FUZZY SETS FOR INPUT VARIABLES

Parameter	Symbol	Values
Angular Error	ϵ_θ	Z (Zero), BN (Big Negative), N (Negative), BP (Big Positive), P (Positive)
Gravity Direction	g	P (Positive), N (Negative)
Head Connector	CF_1	P (Positive), N (Negative)
End Connector	CF_2	P (Positive), N (Negative)

C. Fuzzy Inference

In this work, a set of twenty-seven rules are designed for chain-type configurations of ModRED with any number of modules, as listed in Figure 6[1]. The rules have been designed considering three important aspects, *i.e.,* each module is controlled individually, the motion-type (inchworm, differential-drive, snake, etc.), and the location of the module within the chain configuration, as shown in Figure 3. The GCTs given in the output of each rule are chosen intuitively based on the desired goal.

D. Defuzzification

For the defuzzification process, we have used the *height method* along with a symmetrical and triangular output function denoted by m_{out} that is shown in the bottom sub-figure of Fig. 5. In the defuzzification process, each rule i that is selected or activated in the fuzzy inference engine is assigned a numeric value given by $(\bar{z})^i = m_{out}^{-1}(GCT^i)$. The height method is then used to do a normalized weighting of each $(\bar{z})^i$ value with the minimum membership function value, $\mu_C(\bar{z}^i)$. Finally, the maximum of these values is mapped to a single GCT using the output membership function, as given by the following equations:

[1]An entry of $P - N$ on the input of a rule indicates that both P and N values have the same effect in firing the rule.

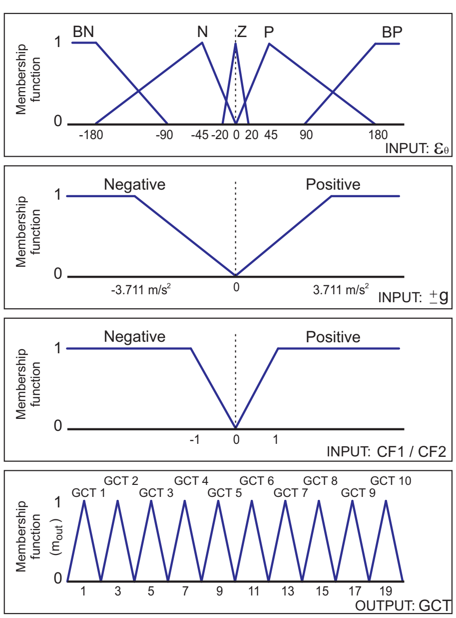

Fig. 5. Membership functions for each of the input and output variables.

$$z^* = \max \frac{\mu_C(\bar{z})^i \times (\bar{z})^i}{\sum_i \mu_C(\bar{z})^i} \quad (1)$$

$$GCT_{out} = m_{out}(z^*) \quad (2)$$

V. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed controller, we performed different experiments with an accurately simulated model of ModRED within the Webots robot simulator and also implemented the controller on the hardware of a ModRED module. We have considered three chain-type configurations of ModRED of varying length - single module, two-module and three-module configurations. Each configuration starts with an arbitrary orientation and the goal direction is introduced into the controller from an external setup formed by an overhead Webcam that tracks an arrow on the ground as an indicator of the goal direction, as shown in Figure 7. This setup also models the effect of noisy input information by having small variations of the arrow's direction captured by a basic computer vision algorithm using the Roborealm software. The intention is to show the controller's behavior is robust to noisy (non-constant) input values. The objective of the controller is to maneuver the modules so that the entire configuration get aligned with the specified goal direction. Videos from our implementation and simulation can be found at http://cmantic.unomaha.edu/projects/modred/index.htm.

RULE	INPUTS				OUTPUT
	ε_θ	g	CF1	CF2	GCT
1	Z	P	N	N	GCT 4
2	Z	P	N	P	GCT 6
3	Z	P - N	P	P	GCT 10
4	Z	P	P	N	GCT 8
5	Z	N	N	P	GCT 7
6	Z	N	N	N	GCT 5
7	Z	N	P	N	GCT 9
8	BN	P - N	P	P	GCT 1
9	BN	P	N	P	GCT 3
10	BN	P	P - N	N	GCT 3
11	BN	N	P - N	N	GCT 2
12	BN	N	N	P	GCT 2
13	N	P - N	P	P	GCT 1
14	N	P	P - N	N	GCT 2
15	N	P	N	P	GCT 2
16	N	N	N	P	GCT 3
17	N	N	P - N	N	GCT 3
18	BP	P - N	P	P	GCT 1
19	BP	P	P - N	N	GCT 2
20	BP	N	N	P	GCT 3
21	BP	N	P - N	N	GCT 3
22	BP	P	N	P	GCT 2
23	P	P - N	P	P	GCT 1
24	P	N	P - N	N	GCT 2
25	P	P	N	P	GCT 3
26	P	P	P - N	N	GCT 3
27	P	N	N	P	GCT 2

Fig. 6. Linguistic rule base for Goal-Directed Behavior showing the inputs and output of the inference engine.

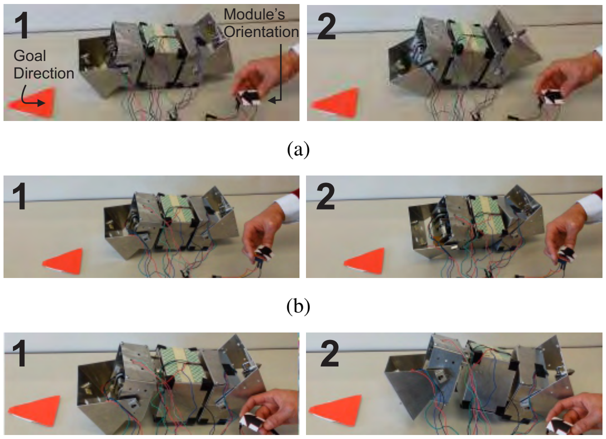

Fig. 7. (a) ModRED's module executing GCT 4 due to zero orientation error. (b) If module's orientation change during locomotion, the FLC immediately compensates the error by selecting GCT 3. (c) Gait adaptation towards the goal by selecting GCT 2.

A. FLC with a Single Module

In the first experiment, to illustrate that the controller is able to respond to dynamic changes in the goal orientation, the goal orientation is changed at certain intervals of time to take the following values: 100°(t=0 sec), 40°(t=45 sec), and -120°(t=100 sec). The sequence of red points indicates the current direction of the single module and its evolution according to the goals, as shown in Figure 8 (top). We observe in Figure 8 (bottom) how the controller is able to effectively select an appropriate gait for the module and adjust the orientation of the robot to match the goal orientation. In the same way, a second more complex experiment is performed to follow the goal direction that is changed at certain intervals of time. Figure 9 displays the controller's response to changing input values. As before, we observe that with the presence of noise in the goal direction, the FLC is able to orientate the robot towards the goal direction.

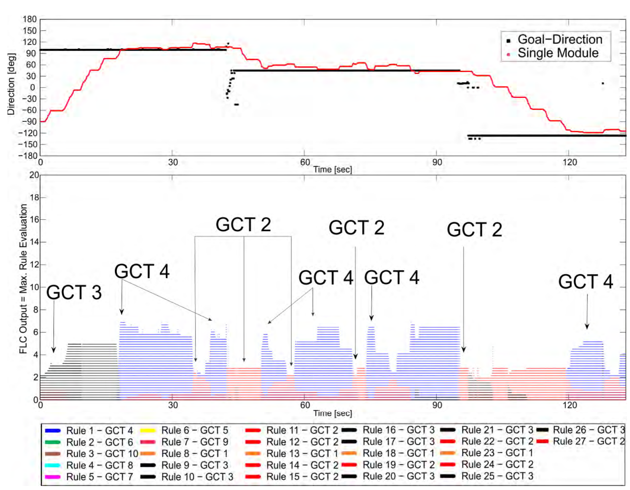

Fig. 8. The controller evaluates each of the rules and activates the adequate GCT according to the goal, module's status, and locomotion type.

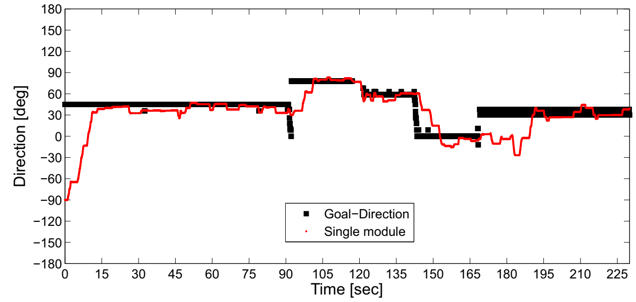

Fig. 9. FLC within a single module adjusting orientation towards different goal directions that are given by an external Webcam-based setup.

B. FLC with Two-module and Three module Chain Configurations

In the third experiment the controller has been implemented into two modules forming a two-module chain configuration in Webots. Each controller receives the goal direction from the external setup and selects the appropriate GCT for each module. Simulation results in Figure 10 shows the change in orientation of this configuration for three different goal-directions.

Finally, we consider adjusting the orientation of a three-module chain configuration of ModRED using the proposed FLC. According to the set of rules presented in Section IV-C, this configuration represents any chain configuration formed by three or more modules. We show in Figure 11 four different goal directions that are set at different time intervals and how the configuration adapts its orientation to move towards the goal directions. The three-modules chain configuration uses most of the GCTs within the library to accomplish the entire task. In the case that the robot has to move forward, three different GCTs are used by each of the modules, for instance GCT 6 by the module at the front, GCT 10 by the module placed at the middle of the chain and GCT 8 by the module at the end of the configuration. If the robot has to rotate for instance, to the right, then GCT

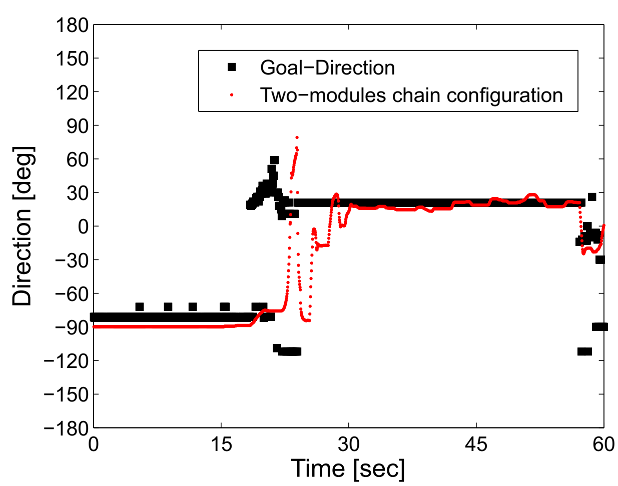

Fig. 10. FLC within a two-module chain configuration adjusting orientation towards different goal directions.

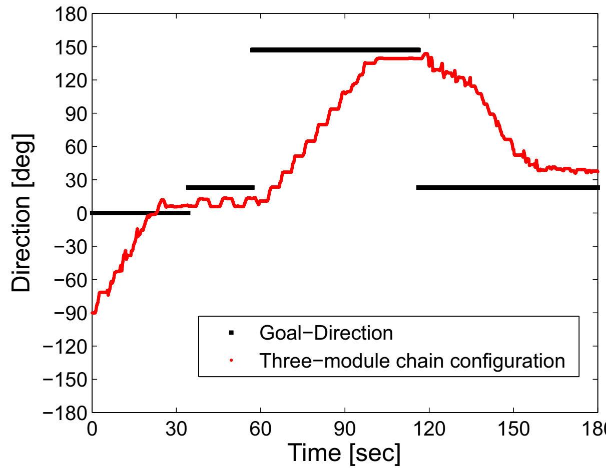

Fig. 11. FLC within a three-module chain configuration adjusting orientation towards different goal directions.

3 is used by the modules placed at the front and at the end of the configuration while GCT 1 is used by the module in middle of the chain.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a goal-driven approach that dynamically adapts the locomotion of a MSR called ModRED. The main contribution of this paper lies in the design and implementation of a distributed FLC that uses the sensor's data from ModRED's modules. It considers the module's position in the configuration to autonomously determine the appropriate gait control table (GCT) for its corresponding module, so that the overall chain-type configuration can continue its locomotion towards the goal.

We have shown our controller works successfully on simulations using Webots and the real implementation in ModRED. The experimental results demonstrate that the approach is able to dynamically adapt ModRED's locomotion and enable the system to maneuver within different chain-type configurations. A good advantage of of this approach is that it can properly work in microcontrollers running at 8MHz using less than 32 kB of flash memory.

The approach can be extended to different configurations (ring-, biped-, quadruped-, spider-type configuration, etc.) by designing a set of rules based on basic motions from a single module for different locomotion types. In addition, the development of new FLCs with different behaviors, using inputs from different types of on-board sensors, leads to the possibility of combining the different behaviors to improve the system's autonomy towards performing more complex tasks or goals. The controller in this paper work can also be combined with higher level path- and task-planning to improve the system's intelligence. In parallel to this research, we are also investigating different techniques for reconfiguration planning of ModRED [17], [18] as well as developing the second generation of ModRED [15].

As future work, we plan to extend the FLC in this paper by adding wireless coordination techniques [19] and different on-board sensors into ModREDs' modules, and, integrating it with ModRED's reconfiguration mechanism to perform more complex maneuvers and tasks. Moreover, we want to combine these types of FLCs with learning techniques to allow the system to autonomously change its current controller and goal according to the acquired experiences.

REFERENCES

[1] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[2] N. Ranasinghe and W.-M. Shen, "Autonomous adaptation to simultaneous unexpected changes in modular robots," in *Reconfigurable Modular Robotics Workshop at International Conference On Intelligent Robots And Systems*, 2011.

[3] M. Park and M. Yim, "Distributed control and communication fault tolerance for the ckbot," in *Proc. Int. Conf. on Reconfigurable Mechanisms and Robots*, 2009, pp. 682 –688.

[4] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 2nd ed. WILEY, 2004.

[5] K. Stoy, R. Garcia, and A. Lyder, "Mechanical design of odin, an extendable heterogeneous deformable modular robot," in *Proc. Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 883–888.

[6] J. Baca, M. Ferre, and R. Aracil, "A heterogeneous modular robotic design for fast response to a diversity of tasks," *Journal of Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 522 – 531, 2012.

[7] A. Ijspeert, A. Sprowitz, S. Pouya, S. Bonardi, J. V. den Kieboom, R. Mockel, A. Billard, and P. Dillenbourg, "Roombots: Reconfigurable robots for adaptive furniture," *IEEE Computational Intelligence Magazine*, vol. 5, no. 3, pp. 20–32, August 2010.

[8] A. Kamimura, H. Kurokawa, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji, "Automatic locomotion pattern generation for modular robots," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 714–720.

[9] D. Christensen, U. Schultz, and K. Stoy, "A distributed strategy for gait adaptation in modular robots," in *IEEE International Conference on Robotics and Automation*, may 2010, pp. 2765 –2770.

[10] P. Zahadat, D. J. Christensen, U. P. Schultz, S. Katebi, and K. Sty, "Fractal gene regulatory networks for robust locomotion control of modular robots," in *Int. Conf. on Simulation of Adaptive Behavior*, 2010.

[11] C.-H. Yu and R. Nagpal, "Self-adapting modular robotics: A generalized distributed consensus framework," in *IEEE International Conference on Robotics and Automation*, may 2009, pp. 1881 –1888.

[12] W. W. Melek and A. A. Goldenberg, "Neurofuzzy control of modular and reconfigurable robots," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 3, pp. 381 –389, sept. 2003.

[13] M. Zhu and Y. Li, "Decentralized adaptive fuzzy sliding mode control for reconfigurable modular manipulators," *International Journal of Robust and Nonlinear Control*, vol. 20, pp. 472–488, 2010.

[14] M. Biglarbegian, W. Melek, and J. Mendel, "Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: Theory and experiments," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 4, pp. 1371 –1384, april 2011.

[15] S. G. M. Hossain, C. A. Nelson, and P. Dasgupta, "Hardware design and testing of modred a modular self-reconfigurable robot system," in *Proc. ASME/IEEE International Conference on Reconfigurable Mechanisms and Robots*, 2012.

[16] M. Bajracharya, M. Maimone, and D. Helmick, "Autonomy for mars rovers: Past, present, and future," *Computer*, vol. 41, no. 12, pp. 44 –50, dec. 2008.

[17] P. Dasgupta, V. Ufimtsev, C. Nelson, and S. Mamur, "Dynamic reconfiguration in modular robots using graph partitioning-based coalitions." in *Intl. Conf. on Auton. Agents and Multi-Agent Systems*, 2012.

[18] A. Dutta, P. Dasgupta, J. Baca, and C. Nelson, "A fast coalition structure search algorithm for modular robot reconfiguration planning under uncertainty," in *Int. Symposium on Distributed Autonomous Robotic Systems*, 2012.

[19] R. Fitch and R. Lal, "Experiments with a zigbee wireless communication system for self-reconfiguring modular robots," in *Proc. IEEE International Conference on Robotics & Automation*, 2009.