2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

A Reconfigurable Spherical Robot

Noppadol Chadil Marong Phadoongsidhi Kawee Suwannasit Poramate Manoonpong and Pudit (Sathaporn) Laksanacharoen

Abstract— **This paper presents a reconfigurable spherical robot. The reconfigurable spherical robot can be reconfigured into a form of two interconnected hemispheres with three legs equipped with three omni-directional wheels. A stable reconfiguration control algorithm is constructed to change the robot from spherical shape to two halves of interconnected hemispheres and three legged-wheeled expansions. This work also constructs a transformation controller for the robot which uses an accelerometer to sense its orientation. The performance analysis shows that our reconfigurable robot prototype can transform from spherical shape (dormant mode) into two interconnected hemispheres where the three leg-wheels are projected out of the shells (transformed mode) and vice versa. After the transformation into the three leg-wheel configuration, the robot can autonomously move in L-shaped and U-shaped areas as well as narrowing pathways.**

I. Introduction

Recent studies show that spherical shape robots have been widely developed by many robotic researchers. A spherical shape can be also benefited in keeping mechanical components and electronic circuits inside a compact volume. The shell can also perform rolling motions for going fast and smooth on flat area.

There are a number of studies on spherical robots. Bing and his team designed a spherical hopping robot [1]. Zheng and his team developed an omni-directional spherical robot with telescopic manipulator [2]. Guanghui et. al developed a spherical robot motion based on principles of conservation of angular momentum [3]. Halme et al designed a spherical robot that has an internal single wheel inside drive unit [4].

In addition, a number of roboticists have become interested in hybrid leg-wheel robots [5] [6]. To combine the advantages of legged-wheeled locomotion and spherical robots, we developed the legged-wheeled mechanism inside the spherical shell for multi-locomotion modes.

In previous work we have developed a simulation of a reconfigurable robot with three omnidirectional wheels mounted on legs [7], [8]. It combines the concept of using legs, wheels, and rolling spheres for the multi-locomotion modes. The conceptual design of this robot was that it will be initially packed and deployed in a spherical configuration.

N. Chadil is with Electrical Engineering Dept, King Mongkut's University of Technology North Bangkok, Bangkok, 10800, Thailand nchadil@gmail.com

M. Phadoongsidhi is with Computer Engineering Dept, King Mongkut's University of Technology Thonburi, 126 Pracha Uthit Road, Bangmot, Thungkhru, Bangkok, 10140, Thailand marong.p@gmail.com

P. Manoonpong is with Bernstein Center for Computational Neuroscience (BCCN) University of Göttingen, D-37077 Göttingen, Germany poramate@manoonpong.com

P. Laksanacharoen and K. Suwannasit is with Mechanical and Aerospace Engineering Dept, King Mongkut's University of Technology North Bangkok, Bangkok, 10800, Thailand STL@kmutnb.ac.th

Due to its closed-spherical shape, the robot offers ease in transportation and deployment; for instance, a number of these robots can be packed and deployed together from an aircraft. Cushioning materials can be added to the shell for soft landing. After landing, the robot may roll for some distance. Afterwards it can transform into two inter-connected hemispheres with three legs extended for locomotion using wheels and/or legs for autonomous exploration.

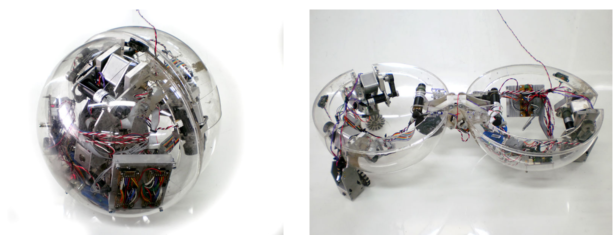


Fig. 1. A Reconfigurable Spherical Robot

II. Mechanical Design

The mechanical design of the reconfigurable spherical robot is constructed with the concept of a spherical shape to be used for rolling on flat ground. The robot can change from a spherical shape into two interconnected hemispheres. The robot can then deploy its three legs.

There are two mechanisms involved in the transformation process: (1) a hinged joint at the middle for transformation of the spherical shape into two interconnected hemispheres was shown in Fig 2, and (2) deployment of the three legs, as shown in Fig 3.

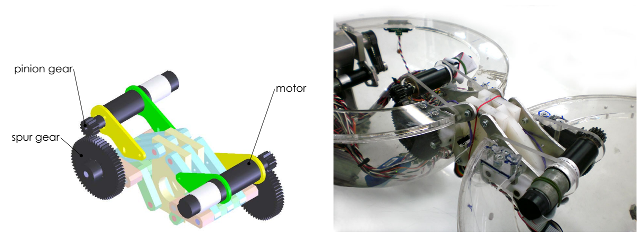


Fig. 2. Hinged joint

The hinged joint at the middle is driven by two MAXON DC motors. Each side of the hemisphere is driven by one motor. The number of teeth for the pinion gear is 15 and 60 for the spur gear.

The spherical shell has a 300 mm diameter and a 5 mm wall thickness. The robot legs are 183 mm long and the omni-directional wheels have a diameter of 50 mm. The

978-1-61284-380-3/11/$26.00 ©2011 IEEE

current prototype robot is tethered to an external power source. The controller is fitted inside the spherical shell. The total weight of the robot is approximately 5 kg.

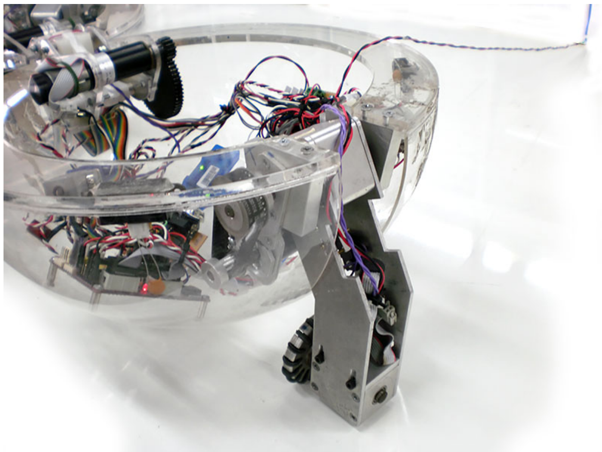


Fig. 3. Leg Mechanism

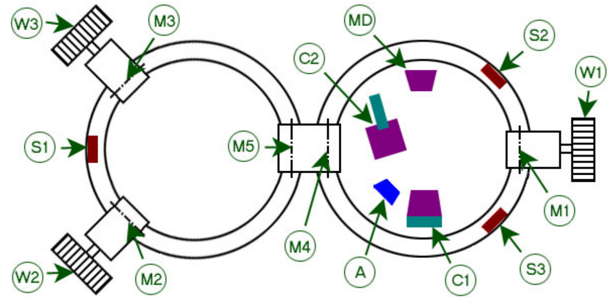
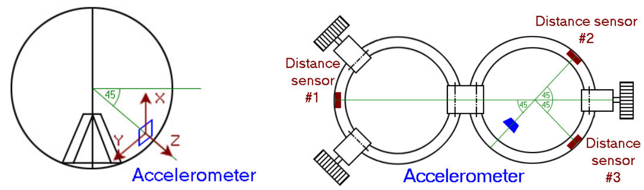


Fig. 4. Control Hardware Components



(a) (b)

Fig. 5. Sensors Installation

III. Robot Control System

For the purpose of efficient deployment and autonomous exploration, our spherical robot consists of two primary configurations. Initially upon deployment, the robot is assumed to be in a *dormant* configuration (i.e., spherical mode). To carry out its mission, the robot will then transform into the roaming configuration. Specifically, the robot begins splitting into two interconnected hemispheres while each of the three legs, in a particular sequence, expand from the shell to give lift and support to the robot body.

Once the roaming configuration is assumed, autonomous exploration may ensue until the mission is completed. At this point, a transformation back to a dormant configuration is possible by leg contraction and closure of the two hemispheres. To enable these actions, we have devised a robust and elaborate system of electronics and control software as described in this section.

A. Hardware Components

The installation of hardware control devices inside each hemisphere of the robot, as shown in Fig. 4, consists of microcontrollers, actuators/sensors, and wireless communication elements. We describe each group of components.

1) Actuators and Sensors: Motion and transformation of the robot are realized by a set of 8 motors. Five MAXON RE-max 21 (6 watt 24 V) DC motors are used. The motor M1-M3 manage the robot's leg deployment, and motors M4 and M5 are used for expansion and contraction.

Three MAXON RE13 (3 watt 24V) DC motors (W1-W3) drive the omni-directional wheels attached to the legs to make roaming possible. A motor drive unit (MD), controlled by a microcontroller, is responsible for feeding electrical current to all the motors.

Two sensor types are employed to determine the robot's position. To ascertain proper positioning before the robot spherical body is allowed to split in half, we use the data read from an Analog Devices ADXL335 accelerometer (labeled as A in Fig. 4) at the onset of the expansion sequence.

Three Sharp 2Y0A21 long distance sensors (S1-S3) help the robot steer around or backtracking away from any obstacles encountered during the autonomous traversal. Figure 5 shows the installation of the described sensors.

Finally, 10 micro-switches (two on each leg and four at the middle joint) are installed to give the operating sequence to the motors and to prevent them from taking the legs and joint motions out of their designated ranges. Each of the three leg joints and the hinged joint also have a potentiometer attached so that joint angles can be monitored for control purposes.

2) Microcontrollers: Intelligent motion programming for the spherical robot cannot be achieved feasibly without cooperation between the actuators/sensors and the microcontroller unit (MCU) brain of the robot. We utilize two such devices, C1 and C2 in the right hemisphere, shown in Fig. 4, for this purpose.

The dsPIC PIC30F4011 (labeled C1) functions as the controller of all eight motors through the motor drive unit (MD). Specifically, C1 interprets the digital signals generated for the MD via the digital-to-analog signal conversion (DAC) process. For the three wheels, speed and duration of wheel motors rotations are defined by pulse-width modulation (PWM) signals from the MCU. In addition, the microcontroller receives signals from a set of micro-switches at the legs and hinged middle joint to restrain their motion from going out of bounds.

Another microcontroller unit, PIC16F690 (labeled C2) is employed solely for communications with the distance sensors, the four potentiometers at the leg and hinged joints, and the accelerometer.

Microcontroller programming is done using Microchip

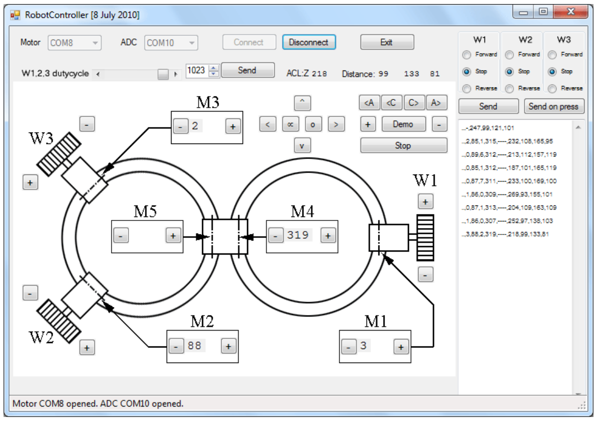


Fig. 6. Control Software Interface

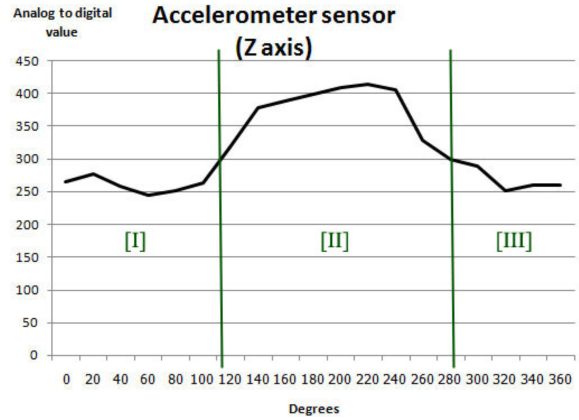


Fig. 7. Regions of Values from Accelerometer

MPLAB v8.46 and C30 development library version v3.23 for the dsPIC PIC30F4011 and CCS PCWH v4.023 for the PIC16F690.

3) Communication Setup: In our prototype, control of the spherical robot, both manual and semi-automated, is performed by a specialized control software running on a personal computer. To achieve this, a Firefly Bluetooth-to-RS232 serial adapter is connected to each of the two micro-controllers for wireless communication with the computer. A practical operating range is approximately 100 meters.

B. Control Software System

During design and development of the spherical robot, it is desirable that we can operate and observe each moving component independently for testing and tuning. A software graphical user interface, as shown in Fig. 6, is therefore developed for easy and efficient control and monitoring. We discuss its development and usage below.

1) Development Tools and Operating Environment: The multi-threaded graphical control software is developed and tested using Microsoft Visual Basic .NET 2005 running on a personal computer with Microsoft Windows 7 operating system. On our test computer (the Intel Core2Duo P7550 2.26GHz), the control software requires up to 50% CPU usage and 6 MB of memory.

2) Software Interface and Usage: A robot operator may use the control software to interact with the robot as follows:

- Connection between the control software interface and the robot can be established by launching the software and selecting ports to the dsPIC PIC30F4011 and PIC16F690 microcontroller units. Within 10 seconds, if the handshaking is successful, the software starts polling and displaying the sensor reads with a one-second period.
- The motors responsible for wheel and joint motion can be operated independently by pushing the $+$ and $-$ buttons (to have the motors rotate clockwise or counter-clockwise), as located on the control software interface in Fig. 6. Moreover, values sampled from potentiometers at the leg and hinged joints (M1-M3 and M4) are displayed with 0 for a fully extended joint configuration and 1023 for a fully retracted configuration.
- To instruct the robot in the spherical configuration (i.e., the dormant form) to transform into the interconnect hemispheric configuration (roaming configuration), a user may push the button labeled ∞. Conversely, to have the robot transformed back to the dormant, spherical configuration, hit the button labeled $\circ$.
- Once in a roaming configuration, an operator can mobilize the robot in various ways. Here we define a robot *head* as the hemisphere with a single wheel, and its *tail* as the one with two wheels. Requesting the robot to traverse head-first requires an operator to push $>$ (manual) or A$>$ (autonomous) buttons. Likewise, going tail-first can be done with $<$ or $<$A labeled buttons.
- Emergency robot suspension is possible by hitting the space bar or clicking on the Stop button in the software interface. All motors then stop instantly. To break off communications altogether, the user can press the Disconnect button.

IV. Robot Transformation and Roaming Algorithm

A spherical robot at any given time may be in either a dormant configuration state (spherical shape) or in a roaming configuration state (two interconnected hemispherical shape). Here we describe the transformation steps involved in transitioning the robot between the two states (via expansion and contraction sequences). Besides maneuvering the robot in the roaming configuration manually, we have implemented an algorithm so that autonomous roaming in an area with obstacles is possible.

A. Home Position of the Robot in Dormant Form

A robot in dormant configuration state cannot transform to another state unless it is in a home position. Fig. 7 shows range of accelerometer values polled in by the control software. A value falling within region [I] or [III] indicates a robot in the home position, ready for transforming to roaming configuration state.

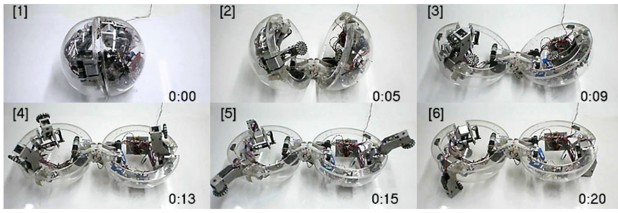


Fig. 8. Expansion Sequence

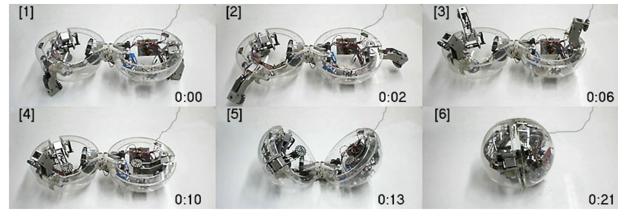


Fig. 9. Contraction Sequence

B. Expansion Sequence

Once in home position, a robot can transform to the roaming configuration state through an expansion sequence. Fig. 8 illustrate the robot going from being fully spherical to fully expanded bi-hemispherical with all the legs lifting up the body, all in 20 seconds. The body begins to split in frame [1], and after 3 seconds leg #2 (driven by M2 in Fig. 4) starts coming out, as shown in frame [2]. When a potentiometer readout at leg #2's joint suggests that it has expanded halfway, the leg expansion pauses while leg #1 and #3 (whose joints are driven by M1 and M3 respectively) starts their own expansion (frame [3]). This continues until the legs are halfway extended (frame [4]) to keep the robot, now fully split, balanced. Then every leg simultaneously proceeds with the expansion all the way until robot body is cleared from the ground (frame [5]-[6]).

C. Autonomous Locomotion

The robot is equipped with three distance sensors to help its locomotion via wheels. We devise a simple algorithm to avoid obstacles while moving on flat ground which, for head-first roaming, can be summarized as follows.

- While roaming head-first, if the value readouts from either sensor S2 or sensor S3 (as depicted in Fig. 4) go above the set threshold, the robot is deemed to face an obstacle, of which it must try to steer clear.
- First, the robot goes in reverse for 1 second to create some space away from an obstacle. While reversing, sensor S1 at the tail ensures that the robot stops before hitting anything.
- Then the robot rotates away from the obstacle. Values from both S2 and S3 sensors are compared to make a decision whether to rotate clockwise or counter-clockwise. Rotation direction is always made with respect to a particular sensor which is further away from the obstacle than the other one. For example, if S3 is closer to a blockage than S2, then the robot rotates counter-clockwise towards S2.
- The rotation continues until the sensor closer to an obstacle is considered clear of the blockage while S1 sensor prevents the tail from hitting anything.
- Robot now can resume with the roaming head-first.
- In the event that a particular blockage prevents the robot from steering clear of it even these steps are followed, an exception routine is taken. Specifically, if the robot rotates back and forth for 4 consecutive times without being able to move forward, it will instead do a longer, 180-degree rotation (5 seconds) before trying again.

A similar algorithm is applied in the case of Tail-first roaming, although the obstacle avoidance performance is slightly degraded due to a single sensor facing the blockage as opposed to two sensors used for head-first roaming. We discuss roaming experiment results later in the performance analysis section. Note that this simple algorithm is implemented as a state machine in order to preliminary test the capability of the hardware. However, for future work we will apply a more advanced control algorithm such as modular neural control for effective autonomous exploration to help preventing the robot from getting stuck in a corner or a deadlock situation [9].

D. Contraction Sequence

A still robot in a roaming configuration may be transformed back to a dormant configuration by going through the contraction sequence as shown in Fig. 9. This sequence is essentially a mirror of the expansion sequence previously explained. It starts with all legs simultaneously being pulled back until halfway (frame [1]-[3]), when the contraction is pulsed. After the robot is allowed a brief moment to regain stability, leg #3 and #1 resume the contraction to the finish (frame [4]). Finally, leg #2 resumes contraction while the hemispheres shut back to the dormant configuration (frame [5]-[6]).

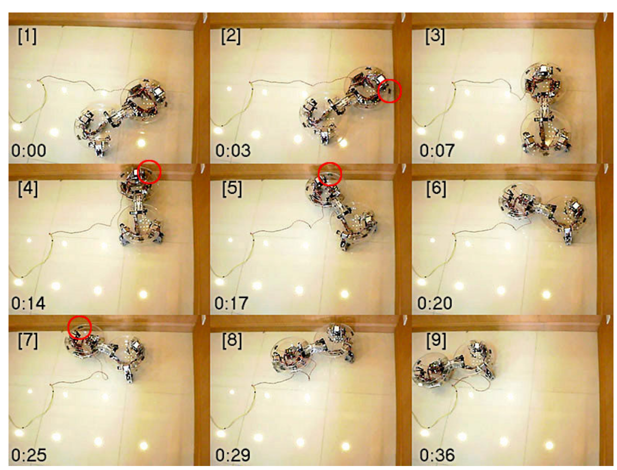


Fig. 10. Roaming in an L-shaped environment

For the current prototype design, we consider only passive rolling while the robot is in the spherical form. However,

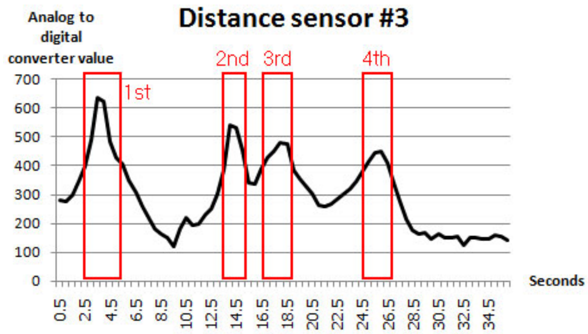


Fig. 11. Sensor output while roaming in an L-shaped environment

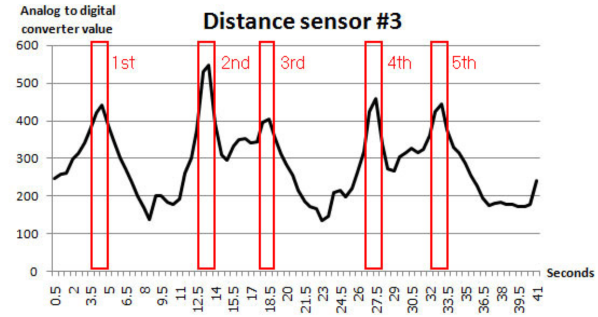


Fig. 13. Sensor output while roaming in a U-shaped environment

active rolling is possible by controlling spinning motion of the three wheels to create centrifugal forces or leg movement to change its center of mass. This will subsequently requires some modifications of the hardware.

V. Performance Analysis

A number of analyses have been performed in various environments to demonstrate the performance of the robot while the robot is in the roaming configuration (leg-wheel mode). The speed of the robot is about 6.8 cm/s on flat ground driven by the three omni-directional wheels.

A. Autonomous roaming in an L-shaped environment

Fig 10 shows a series of photos according to the movement of the robot in a corner or L-shaped environment. Four collisions occur approximately at 3 s, 14 s, 17 s, and 25 s, respectively, during the movement which activate the right distance sensor as shown in Fig 11. The total time to get away from this environment is about 36 s.

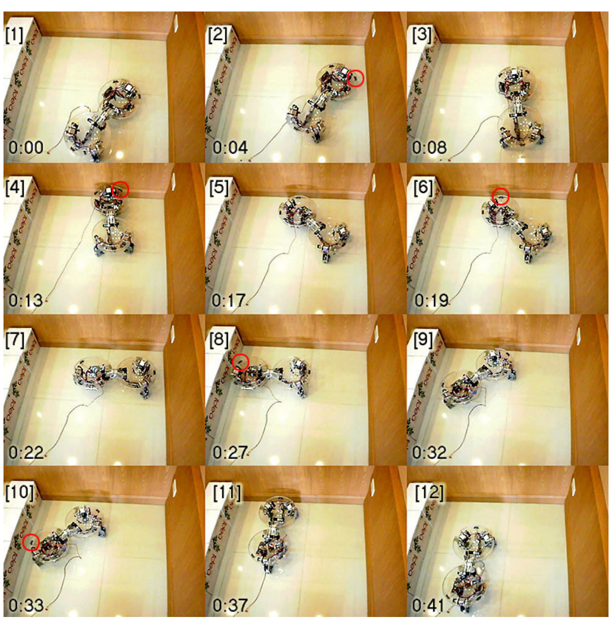


Fig. 12. Roaming in a U-shaped environment

B. Autonomous roaming in an U-shaped environment

Fig 12 shows a series of photos when the robot moves into a dead-end environment or U-shaped environment. A total of five collisions occur. The sensors output signal is shown in Fig 13.

C. Autonomous roaming in a narrowing pathway

Fig. 14 shows the robot moving in a narrowing pathway. The two distance sensors work alternatively to help the robot get across the narrowing pathway, as shown in Fig 15. At the first collision, distance sensor #2 activates at about 3 s, followed by distance sensor #3 at about 13 and 18 s. Finally, the robot moves out of the narrowing pathway successfully. Currently, the robot traversal is calibrated for a minimum pathway width of approximately 40 cm.

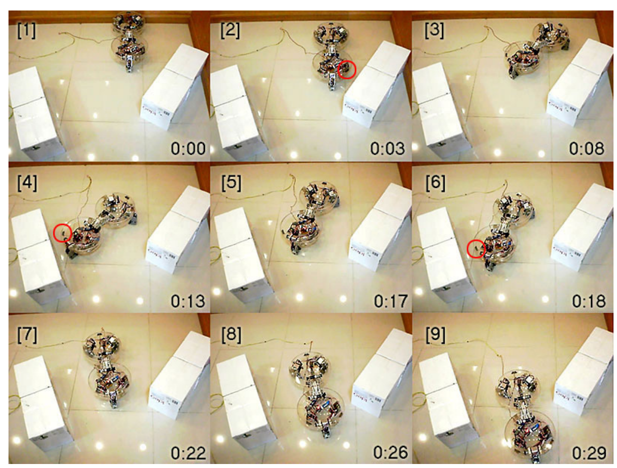


Fig. 14. Roaming in a narrowing pathway

Fig. 16 shows that there is no effect of the installation of the long distance sensors inside the acrylic surface of the sphere.

VI. Conclusions

We have successfully built a reconfigurable spherical robot and verified that the simulation model in our previous work can be feasibly constructed [7], [8]. The robot is now fully capable of reconfiguration from spherical shape into

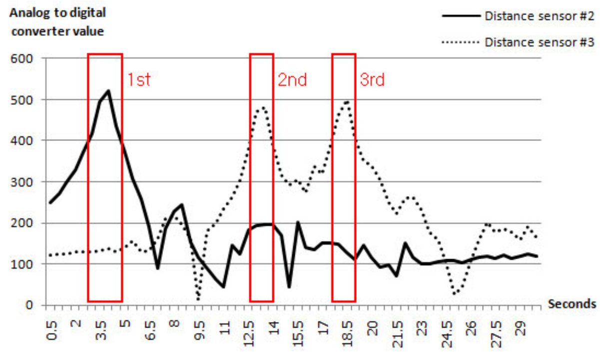


Fig. 15. Sensor output while roaming in a narrowing pathway

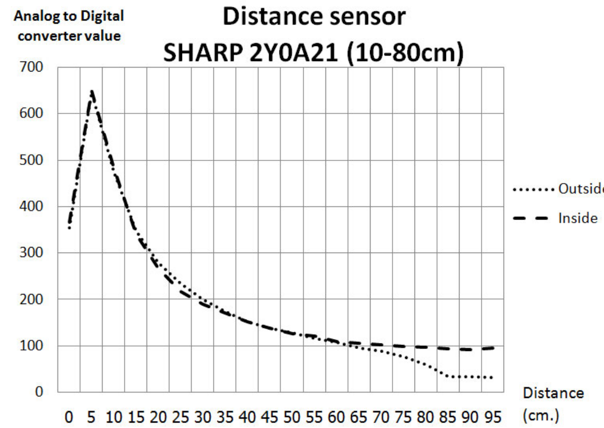


Fig. 16. Distance sensors

a two-interconnected hemispheres with three leg-wheels. The locomotion mode presented in this paper involves the roaming using three omni-directional wheels. For future work, we plan to implement a navigation system and leg movement algorithm to add walking or climbing ability to the robot. This reconfigurable spherical robot is expected to have various types of applications such as search and rescue, disaster mitigation and surveillance for security. In addition to this, we will develop the next version of this robot to achieve active rolling in a spherical form and to have more space in order to install batteries inside the robot.

VII. ACKNOWLEDGMENTS

This research was supported by the Higher Education Commission of Thailand and partially by the Faculty of Engineering, King Mongkut's University of Technology North Bangkok through patent registration.

REFERENCES

[1] B. Li, Q. Deng, and Z. Liu, "A spherical hopping robot for exploration in complex environments," in *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics ROBIO2009*, Guilin, China, December 2009, pp. 402–407.

[2] Z. Yili, S. Hanxu, J. Qingxuan, S. Chenkun, and Z. Kailiang, "An omni-directional rolling spherical robot with telescopic manipulator," in *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium*, Shenzhen, China, February 2009, pp. 1–6.

[3] G. Shu, Q. Zhan, and Y. Cai, "Motion control of spherical robot based on conservation of angular momentum," in *Proc. of the 2009 IEEE International Conference on Mechantronics and Automation ICMA2009*, Changchun, China, August 2009, pp. 599–604.

[4] A. Halme, T. Schonberg, and Y. Wang, "Motion control of a spherical mobile robot," in *Proceedings of the 4th International Workshop on Advanced Motion Control AMC1996*, Tsu, Japan, 1996, pp. 100–106.

[5] G. Endo and S. Hirose, "Study on roller-walker-adaptation of characteristics of the propulsion by a leg trajectory," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1532–1537.

[6] M. Lauria, Y. Piguet, and R. Siegwart, "Octopus: An autonomous wheeled climbing robot," in *Proceedings of the 5th International Conference on Climbing and Walking Robots*, 2002.

[7] S. Laksanacharoen and P. Jearanaisilawong, "Design of a three-legged reconfigurable spherical shape robot," in *Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM2009*, July 2009, pp. 1730–1733.

[8] ——, "Dynamic simulation of a reconfigurable spherical robot," in *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics ROBIO2008*, February 2009, pp. 2156–2160.

[9] P. Manoonpong, F. Pasemann, and H. Roth, "Modular reactive neuralcontrol for biologically-inspired walking machines," in *The International Journal of Robotics Research*, vol. 26, no. 3, 2007, pp. 301–331.