24th International Symposium on Automation & Robotics in Construction (ISARC 2007)
Construction Automation Group, I.I.T. Madras

THE DEVELOPMENT OF A GENERAL-PURPOSE AUTONOMOUS ROBOT

Kai-Yuan Gu
Master Student
Department of Civil Engineering
National Taiwan University
No. 1, Roosevelt Road, Sec. 4, Taipei, 106, Taiwan
ttx@caece.net

Shih-Chung Kang
Assistant Professor
Department of Civil Engineering
National Taiwan University
No. 1, Roosevelt Road, Sec. 4, Taipei, 106, Taiwan
sckang@ntu.edu.tw

Ping-Hung Lin
Master Student
Department of Civil Engineering
National Taiwan University
No. 1, Roosevelt Road, Sec. 4, Taipei, 106, Taiwan
b92501030@ntu.edu.tw

Jia-Ruey Chang
Associate Professor
Department of Civil Engineering
MingHsin University of Science & Technology
No. 1, Hsin-Hsing Road, Hsin-Chu, 304, Taiwan
jrchang@must.edu.tw

Mao-Nan Chen
Deputy Director
Division of Transportation Engineering
Institute of Transportation
Ministry of Transportation and Communications
No. 240, Tun-Hwa North Road, Taipei, 105, Taiwan
chenmn@iot.gov.tw

ABSTRACT

This paper presents the ongoing research into the design and development of a general-purpose autonomous robot. The major components of the robot are four ultrasonic sensors, one electronic compass, three microprocessors, a laptop computer, and three motors for three wheels. Software architecture was developed to process the information obtained from the sensors, plan the motion of the robot, and send the commands to the motors. This allows the robot to operate independently of human control thus fulfilling the definition of an autonomous robot. A preliminary test was conducted in the field. The results indicate that the robot is capable of sensing static and dynamic obstacles in the environment and avoiding contact with them during motion.

KEYWORDS

Robot, Construction Automation, Autonomous Robot

1. INTRODUCTION

Automation and robotics have generated much interest in the construction community over the last two decades. The major reasons include high accident rates at construction sites, inconsistent product quality, and the amount of labor wasted in repetitive construction tasks, such as earthmoving or road inspections. Many researchers, such as Skibniewski and Russell [1] and Maynard et al. [2], have developed robots for various construction applications. These robots are referred to as single-task robots; and, generally result in productivity and efficiency improvements.

One disadvantage of such single-task robots is their low cost effectiveness. Because the robots are designed for solving a specific problem in a pre-defined scenario, the researchers not only need to assemble the hardware (including sensors and actuators) for each application of the robot, but also need to develop algorithms to synchronize sensor inputs with the robot's motion. This usually involves a trial-and-error process and requires a large effort. Unfortunately, most of a robot's hardware and software architectures are incompatible with other robots, so the development efforts cannot be leveraged.

In this research we aim to develop an autonomous robot that has a flexible hardware and software architecture. An autonomous robot is an intelligent machine capable of performing tasks without explicit human involvement. The robot developed in this research will facilitate various research topics regarding construction automation in the future.

2. HARDWARE COMPONENTS

Since we aimed to develop a general-purpose robot, we designed hardware architecture with a high degree of flexibility and extensibility. It is composed of three modules: (1) a motion module; (2) a sensing module; and (3) a logic/control module (Figure 1). This architecture allows us to replace or add sensors and actuators in a way such that there is no detrimental influence from other parts.

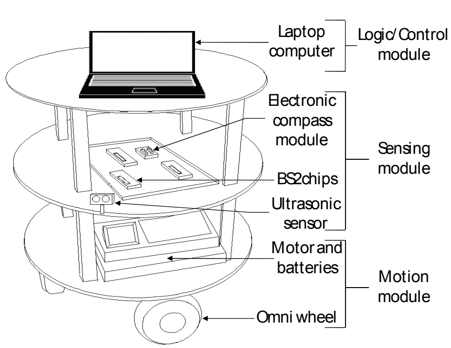

Figure 1 The Hardware Structure of the Developed Robot

The motion module is on the lowest shelf of robot. It includes two batteries, three omni wheels (wheels without lateral friction), a motor for each wheel, and power control devices (Figure 2). The sensing module is responsible for obtaining information from the external environment. All major components, excepting the ultrasonic sensors, are located on the middle shelf of the robot. It includes two BS2 microprocessors [3], one BS2p24 microprocessor [3], an electronic compass, and four ultrasonic sensors. The logic/control module is located on the top shelf of the robot. At present, we are using a laptop computer as the logic/control module. This module analyzes the information obtained from the sensors to make decisions on the robot's motions. Although the laptop computer satisfies our current needs, it could be replaced with one or several more powerful industrial computers; allowing for more complicated and advanced applications.

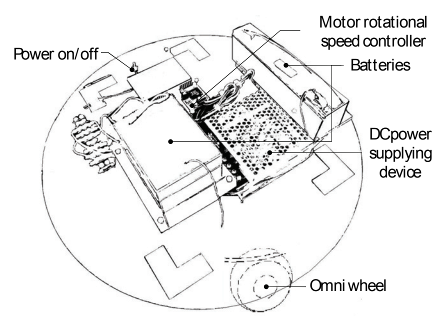

Figure 2 Motion Module

3. SOFTWARE COMPONENTS

To automate the robot, we developed software architecture to control and define the robot's behaviors. The software architecture enables the robot to process sensory inputs, perform cognitive functions and provide signals to output actuators. In other words, the software architecture of a robot defines how the job of generating actions from perceptions are organized [4]. This chapter first overviews the software architecture used in the robot, illustrates the information flow during robot's motion, and explains the implementation of the program to realize an autonomous robot.

3.1 Software Architecture

As shown in Figure 3, the software architecture is separated into three modules: the sensing module, the motion module, and the logic/control module.

The sensing module includes five sensory inputs: four inputs for ultrasonic sensors and one for the electronic compass. The information obtained from the sensors is transferred to the microprocessor in the logic/control module.

The motion module includes three motors, each of which controls a wheel. This module is designed to receive the information sent from logic/control module.

The function of the logic/control module is to integrate the information obtained from the sensors, execute cognitive functions, and then send the information to actuators, which in turn triggers the robot's motion. This module includes three microprocessors, a Bluetooth wireless communication chip, and a laptop computer. One of the microprocessors (BS2 chip) receives the data sent from sensing module. It integrates the data from all sensors and then sends it to the central microprocessor (BS2p24 shown in Figure 3). Since the BS2p24 microprocessor has greater computation performance than the other microprocessors, it performs the function of central communicator between the sensors, the computer and the motors. A Bluetooth chip is employed to wirelessly link the BS2p24 microprocessor with the computer. The BS2p24 microprocessor also needs to send commands to the BS2 microprocessor, which coordinates the three motors.

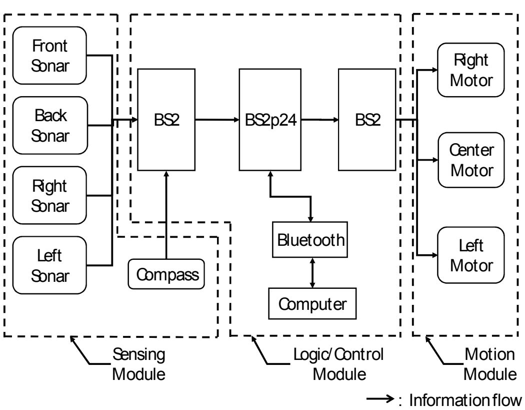

Figure 3 Software Architect of the Robot

3.2 Controller Interface

Controlling the motion of the robot requires simultaneously controlling the three wheels, which would be extremely complex for users. Therefore, a user interface that simplifies the control and allows users to control the robot's motion using a computer input device was developed. We used Microsoft Robotics Studio [5] as a development platform. Figure 4 shows the control interface.

Our major effort in this part of robot was linking the computer interface with the BS2p24 microprocessor via Bluetooth wireless protocol (shown in Figure 3). A web service (a piece of software that makes itself executable over the Internet) is constantly connected to the BS2p24 microprocessor. By connecting with the web service, we are able to use a trackball, as shown in the control interface, to control the robot. For example, we may define that when the user rolls the trackball upward, the intention is to move the robot forward. To achieve the goal, we wrote a computer program to send the commands to the BS2p24 microprocessor. The program interprets these commands to control the speeds of each of the three wheels. The combined of effect of the rotational motion of the three wheels, in combination, drives the robot forward. Similarly, we also define: (1) that rolling the trackball to the right is to turn the robot to the right (rotate clockwise from the top view); (2) that rolling the trackball left is to turn the robot left (rotate counter-clockwise from the top view); and, (3) rolling the trackball downward is to move the robot backwards.

This interface also displays the status of the robot, such as its position and orientation. It also displays any information received from the sensors. In Figure 4, we can see the obstacle detected by laser scanner result in the sub-window shown in obstacle display (right bottom of the window).

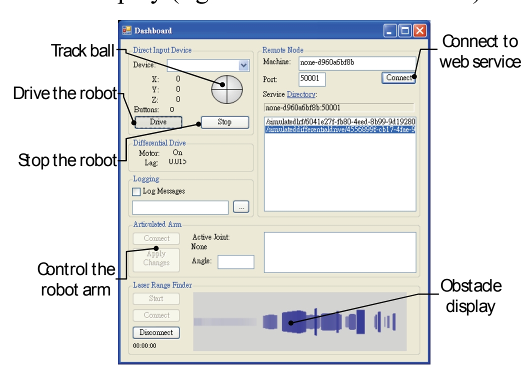

Figure 4 The Robot Control Interface

3.3 Motion Control

Because we are developing a general-purpose robot, we need to control not only the movement of the robot but also rotation of the robot. Thus the robot can be moved to a target position and target orientation. To achieve the goal, we define two wheels as the front wheels. The intersection between the center of the robot and the imagined extension lines of the front wheels was defined as the forward direction. The third wheel functions as a rudder, for rotation.

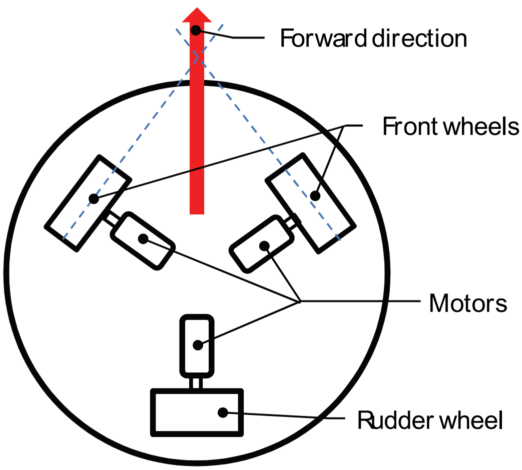

Figure 5 The Wheels and Forward Direction

After defining the forward direction, let's discuss how the robot moves forward, backward, turns right and turns left; all by controlling the wheels. When the robot moves forward, the motor for the left front wheel rotates counterclockwise and the motor for the right front wheel rotates clockwise with equal angular speed (vectors shown in Figure 6 (a)). When the robot moves backwards, the motor for the left front wheel rotates clockwise and the motor for the right front wheel rotates counterclockwise (vectors shown in Figure 6 (b)). Because the omni wheel is frictionless during lateral movements, the rudder wheel remains still during both forward and backward movement. When the robot turns right, the motors for all three wheels simultaneously rotate clockwise with equal angular speed (as shown in Figure 6 (c)). When the robot turns right, the motors for all three wheels simultaneously rotate counterclockwise with equal angular speed (as shown in Figure 6 (d)). If we would like to control the robot arbitrarily, we interpolate the rotational speeds found in the two of the four cases to compute the appropriate angular speeds for each of the three motors.

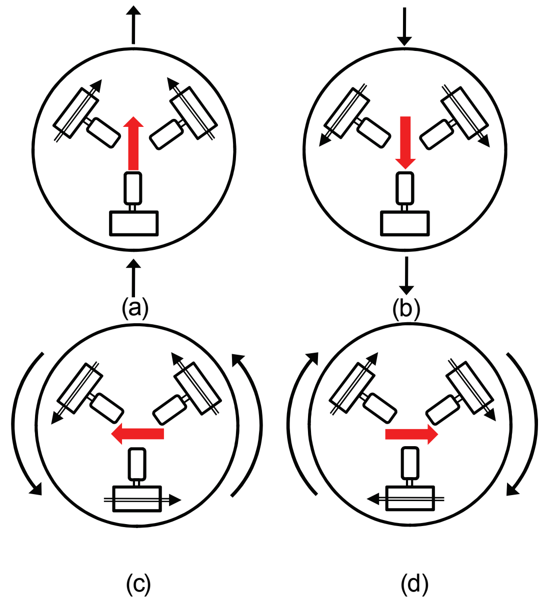

Figure 6 Wheel Controls: (a) Move Forward; (b) Move Backward; (c) Turn Left; (d) Turn Right

4. FIELD TEST

We conducted a field test to verify the integration between the hardware and software. We also wanted to see whether this robot fulfills the definition of an autonomous robot – a robot capable of intelligent behaviors without human controls. During the test, we wrote a computer program and installed it on the computer located on the top shelf of the robot. This computer controls the behavior of the robot and enables it to travel smoothly in an indoor environment.

Two major behaviors are coded for in the computer program. One is random patrol (walk behavior) and the other is obstacle avoidance. The random walk behavior can be achieved by assigning a random distance for the robot to move forward and a random angle for it to rotate. By repeating the forward and rotational motions, the robot will patrol randomly in the environment. Obstacle avoidance is achieved by analyzing the information obtained from sensors coupled to the motion controls.

To seamlessly shift between the two behaviors, Concurrency and Coordination Runtime (CCR) [5] was employed. CCR is the mechanism provided by Microsoft Robotic Studio. It allows programmers to define multiple executing threads with different priorities. It also provides "dispatcher", a mechanism which executes the viable thread with a higher priority. By controlling the priority of each thread, users are able to integrate multiple behaviors in the robot. To integrate the random patrol and obstacle avoidance behaviors, we set obstacle avoidance behavior as having the higher priority. When the sensors detect an obstacle, the obstacle avoidance behavior is activated. Since the sensors are at different locations in the robot, various different obstacle avoidance behaviors are defined.

Figure 7 illustrates one of the robot's behaviors in the test. We placed the robot in a hallway of the civil engineering building at National Taiwan University. A tester was assigned as a dynamic obstacle. During the test, the robot first patrolled randomly in the environment (Figure 8 (a)). While it was close to the dynamic obstacle, the tester, the front ultrasonic sensor received an ultrasound response (Figure 7 (b)). Then the robot moved a random distance backwards. Finally it rotated by a random angle (Figures 7 (c) and (d)). When the obstacle could not be detected by any of the sensors, the robot disabled its obstacle avoidance behavior and returned to random patrol behavior.

5. CONCLUSION

In this research, we have built an autonomous robot. The robot's electronic hardware consists of five sensors, three motors, three microprocessors, and a laptop computer. The hardware is separated into three shelves, a motion/sensing module, a control module and a logic module. We provided software architecture that integrates the information flow in the robot. We also developed a user interface to allow users to control the robot with high level commands and at the same time display the information collected by the sensors on the robot. From the results of the preliminary field test, we determined that the robot is capable of performing a random patrol in an unfamiliar environment; and, collision with both static and dynamic obstacles can be avoided.

In the future, we plan to integrate a laser range finder and camera into the robot. We also plan to use the robot for various pavement inspections, such as measuring pavement roughness.

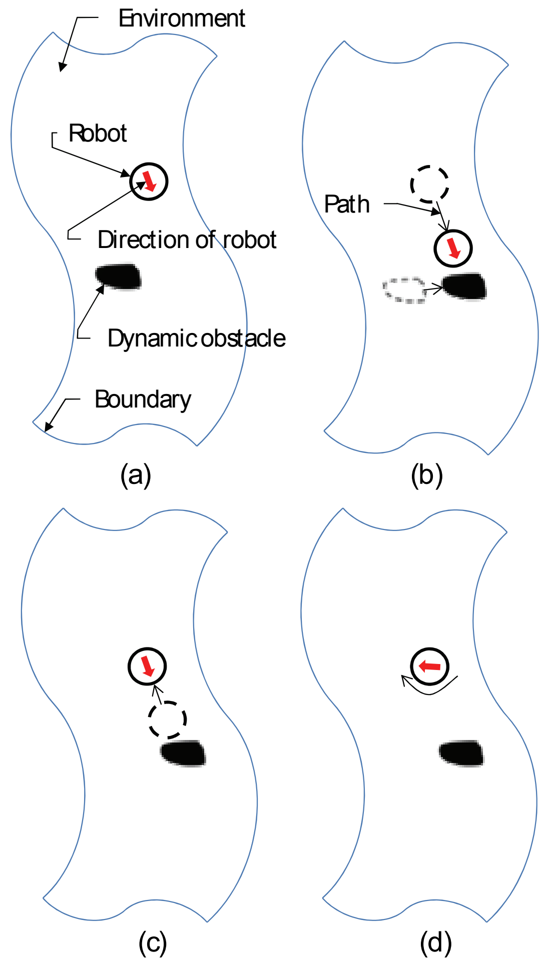

Figure 7 Tests for Autonomous Robot: (a) Random Patrol; (b) Sensing Obstacle; (c) Obstacle Avoidance; (d) Finding Alternatives

6. ACKNOWLEDGMENT

The authors would like to thank the financial support provided by the Technology Development Program for Academia, the Ministry of Economic Affairs, Taiwan, under the project of "The Development of Integrated Intelligent Robotics System" (contract no: 95-EC-17-A-04-S1-054).

7. REFERENCES

[1] Murali, K. & John, B. (1998) Constructing Hydraulic Robot Models Using Memory-based Learning. IEEE Intelligent

Robot Systems Conference, Victoria, British Columbia.

[2] NaitoJunpei, O. N. (2006) Development of a Wearable Robot for Assisting Carpentry Workers. International Symposium on Automation and Robotics in Construction (ISARC) 2006 Session Table, Japan, 523-526

[3] Parallax. (2004) BASIC Stamp Syntax and Reference Manual Version 2.1. Retrieved May 11, 2007, http://www.parallax.com/dl/docs/prod/stamps/basicstampman.pdf

[4] Russell, S. & P. Norvig (2003) Artificial Intelligence A Modern Approach, Upper Saddle River, New Jersey: Prentice-Hall.

[5] Microsoft. (2006) Microsoft Robotics Studio. Retrieved May 9, 2007, http://msdn2.microsoft.com/zh-tw/robotics/default(en-us).aspx