ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

TOWARDS A LOW-COST AUTONOMOUS OBJECT MANIPULATION DEVICE

P. Chand, M. Shazil and E. Kumar
School of Engineering and Physics, University of the South Pacific, Suva, Fiji
E-Mail: chand_pc@usp.ac.fj

ABSTRACT

This paper describes the development of a low-cost autonomous mobile robot for transporting distributed objects in a manufacturing application. A VEX mobile robot kit is utilized to design and construct a prototype robot that is employed in a model object transfer task. For autonomous operation, the behaviour-based programming method is employed to build two complex behaviours that are sequentially executed. The performance of the designed robot is evaluated by measuring the time taken to transport up to five objects to the goal location. Experimental results indicate that the prototype robot can successfully transport objects to the goal and re-grasp unsuccessfully manipulated objects.

Keywords: mobile robot, autonomous, manipulation, behaviour-based control.

1. INTRODUCTION

Autonomous robotic devices are widely employed in manufacturing systems. The mechatronics research group at the University of the South Pacific (USP) has developed several devices such as a drilling machine [1], a pick and place robot [2] and an automated guided vehicle [3]. At present, an in-house milling machine is also under development. These devices can be employed in a customised manufacturing system for teaching purposes that is also being developed at present. In our application, multiple items that build up in a temporary storage area need to be transferred to a final storage location.

Manual labour may be necessary after automation concludes in some manufacturing applications. For instance, automation can end in the packaging area for tobacco processing. However, items may still need to be transferred to a storage area where inventory processing occurs. Transfer to the storage area can be slower than the packaging process, especially when there are multiple packaging line exits but only few entry points to the storage area. Thus, an intermediate area where items are temporarily stored may exist between packaging and storage. These items can be either manually or autonomously transferred to the entrance of the storage room for inventory purposes. Autonomous transfer is preferred to reduce labour costs. Guided vehicles may not be suitable if the items are dispersed in the environment.

Eliminating human input from transportation tasks in a manufacturing process requires mobile devices that comprise hardware and software elements. The hardware is usually a mobile robotic platform with sensors, actuators, coupled with communication and processing devices [4]. Commercially available mobile robotic platforms with object manipulation capabilities are generally prohibitively expensive. A review of commercially available platforms is presented in [5]. Medium sized robots such as the MR5 [6], RoboProbe Model 2 platform [7] and MURV-100 [8] are mainly used for hazardous tasks like bomb disposal. The cost of these robotic platforms varies between US$58 K and US$185 K depending on the options chosen.

Other potential commercial platforms with manipulation capabilities include the PowerBot, Pioneer P3-DX and Pioneer P3-AT robots from MobileRobots Inc. [9]. The cost of the PowerBot varies from US$25 K to $85 K depending on the options chosen. The Pioneer robots are cheaper with a price tag of US$6.3 K to US$12.4 K depending on the type of manipulator selected.

However, all the commercially available robotic platforms evaluated are beyond the budget constraints of the system under development. An alternative option is to develop a mobile robotic platform with mobile robot kit sets such as Lego [10] and VEX [11]. The VEX robot kit was selected since the mechatronics research group had previously purchased one for approximately US$300. Additional components for sensing and manipulation cost approximately US$215 and US$55 respectively.

Software control architecture provides the framework for intelligent control of the developed mobile robot. Three main categories of mobile robot control architectures include: reactive, deliberative, and hybrid systems [12]. Behaviour-based systems [13] can be classified under reactive control systems and is one of the dominant approaches in mobile robot control. Inspired by biological systems, behaviour-based systems employ distributed and parallel primitive behaviours that are coordinated to produce an intelligent robot. Hence, the behaviour-based method is employed to intelligently control the developed mobile robot.

2. TEST ENVIRONMENT CONFIGURATION

Figure-1 illustrates the developed mobile robot in its test environment. The workplace is a 2.4 m × 2.4 m flat rectilinear black surface. Walls approximately 15 cm in height are mounted along the edges of the workspace to contain the robot. A light source at the bottom right corner of the workspace denotes the goal location where objects must be transported.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

The objects are white in colour in order to distinguish them from the workspace surface. A wheeled mobile robot with various sensors and a manipulator is employed to transport the objects (section 3). To successfully locate and transport the objects to the goal, a behaviour-based control strategy comprising two complex behaviours is employed (section 4).

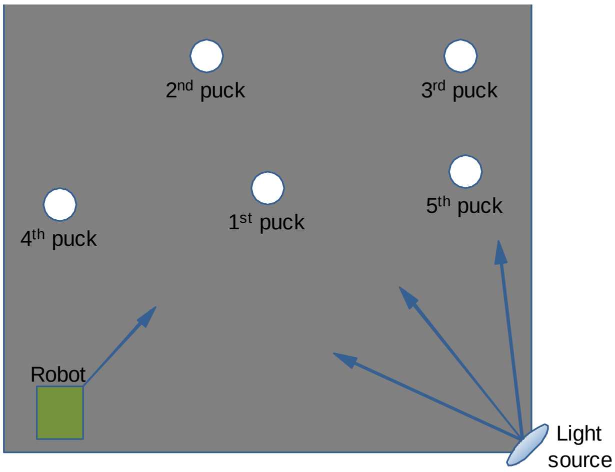

Figure-1. Robot and objects (pucks) in the test environment

3. HARDWARE

Figure-2 illustrates the robot hardware that has been designed for the object transportation task. It consists of a robot base with a range of actuation and sensing devices (section 3.1), and an onboard microcontroller (section 3.2).

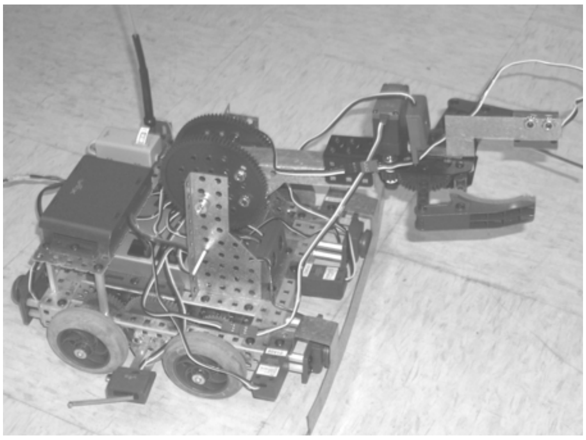

Figure-2. Robot hardware.

3.1 Actuation and sensing

The designed robot is based on the Squarebot [14] which has been adapted to incorporate a manipulator. A key feature of the Squarebot is that it permits good integration of additional components to its base platform. The Squarebot base consists of four wheels, two motors, a microcontroller, an RF receiver and a power supply. Relevant technical specifications of the base are outlined in Table-1. The selected gear ratio enables the robot to travel fast without compromising object handling ability. Additionally, the four wheel differential drive provides stable manoeuvrability and traction.

Mounted on the base of the robot is a manipulator which has been adapted from the VEXplorer [15] claw and wrist kits. The claw is capable of grasping objects up to 8.5 cm wide. Vertical motion restricted to 60° is achieved using the wrist kit. An arm extension to connect the wrist and claw has been fabricated in-house using sheet metal. The wrist and claw are both driven by servo motors. Overall, the manipulator weighs approximately 0.75 kg.

Table-1. Robot base specifications.

Name	Value
Size (L×W)	0.25 m × 0.25 m
Drive type	4 wheel differential (two motors)
Power supply	7.5 V DC
Maximum motor speed	100 rpm
Gear ratio	60:36
Maximum wheel speed	6.67 m/sec
Stall torque	1.46 Nm

A variety of sensors are also attached to the base of the mobile robot (Table-2).

Table-2. Sensor details.

Type	Quantity	Power supply	Range
Ultrasonic	1	5 V DC	0.03-3 m
Line Tracker	3	5 V DC	0-5 V (5-25 mm)
Bumper	4	5 V DC	–
Limit Switch	3	5 V DC	–
Light Sensor	1	5 V DC	0-5 V (< 1.8 m)

An ultrasonic sensor positioned at the front of the robot is employed to detect walls prior to collision. Three line tracking sensors located at the front of the robot enable detection of objects in the workspace. These sensors output a 0-5 V signal and are sensitive within 5-25 mm. A bumper sensor has been placed in each of the four corners of the robot's base. These bumper sensors detect collisions with the walls of the workspace. Three limit switches are employed by the robot. One limit switch is mounted on the claw to detect a successfully gripped object. Two limit switches are positioned on the left-rear and right-rear of the robot. These switches assist the robot to steer away from the wall when side contact is made. The primary function of the light sensor mounted at the front of the robot is to detect the light source (location where objects are to be transported). With a detection range of up to 1.8 m, a 0-5 V analogue output signal represents illumination.

3.2 Microcontroller

Microcontrollers embed a processor core, memory and programmable input/output ports on a single chip. This enables them to function as a computer system to control the robot. Table-3 lists the key features of the VEX robot microcontroller [16]. The controller contains two PIC18F8520 microprocessors and multiple input/output ports. The first "master" microprocessor is employed to remotely control the robot. A second "user" processor is available for software development to autonomously control the robot. Sixteen ports are configurable as analogue or digital input/output. Eight PWM outputs are also available for motors or servos.

Table-3. Key microcontroller specifications.

Name	Value
Power supply	7.2 V DC
Processor type	PIC18F8520
Flash memory	32 KB
Input/Output (I/O)	16 analogue and digital combined
Interrupts	6
Motor outputs	8 PWM outputs

Figure-3 shows the connection of hardware components to the microcontroller. The bumper sensors, limit switches, line tracking sensors and light sensor are connected to the sixteen analogue and digital input/output ports. The four motors that drive the robot and its manipulator (wrist and claw) are connected to the motor outputs. To generate an interrupt signal when the robot is close to the wall, the ultrasonic sensor is connected to the interrupt port.

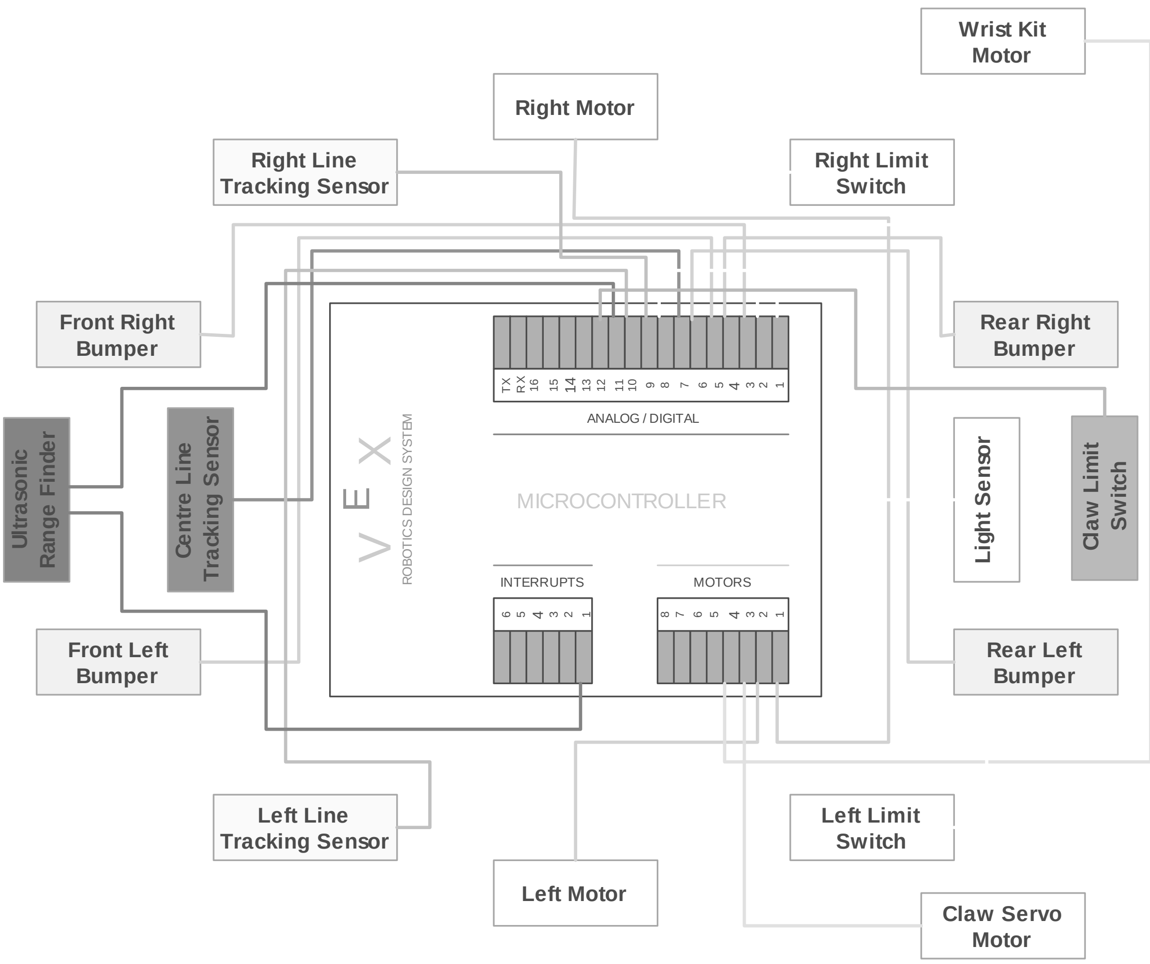

Figure-3. Microcontroller–hardware connections.

4. SOFTWARE

Easy C, a specifically designed software for the VEX controller, has been used to program the robot. The software has two modes of programming: block programming and code programming. Code programming allows greater flexibility and is utilized for the software control.

A high-level overview of the software control for object transportation is shown in Figure-4. Two complex behaviours, search for object and transport object to goal are sequentially executed to achieve the task objectives.

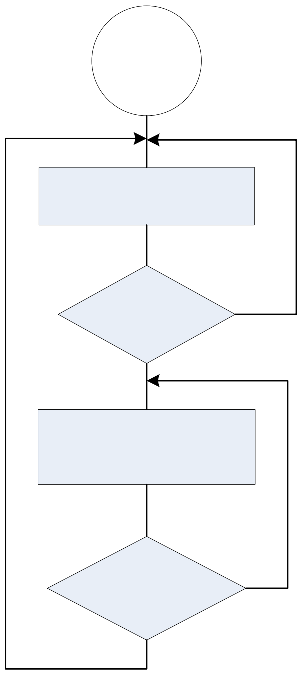

Figure-4. Software control overview.

Six lower level behaviours are combined to produce the two complex behaviours (Figure-5). The move behaviour enables the robot to move in a straight line. To avoid obstacles the avoid wall and avoid other objects behaviours are utilized. When the avoid wall behaviour is triggered from the ultrasonic sensors, the robot turns through a randomly selected angle to avoid the wall. The avoid other objects behaviour is employed when the robot is carrying an object to the goal. To detect objects, the detect object behaviour utilizes the line tracking sensor inputs. When an object is detected, the pick object behaviour controls the manipulator to grip the object. After the object has been successfully lifted, the search for goal direction behaviour adjusts the orientation of the robot for movement towards the light source.

The execution of the lower level behaviours is coordinated using priority-based arbitration to achieve complex behaviour. In this approach, certain behaviours have priority over others depending on the situation perceived by the robot's sensors. For example, if a wall is detected while the robot is moving in a straight line searching for an object, the movement will be subsumed by the avoid wall behaviour. The search for object complex behaviour requires arbitration of the move, avoid wall and detect object behaviours. To achieve the requirements of the transport to goal behaviour, the move, avoid wall, avoid other objects, pick object and search for goal direction behaviours are arbitrated.

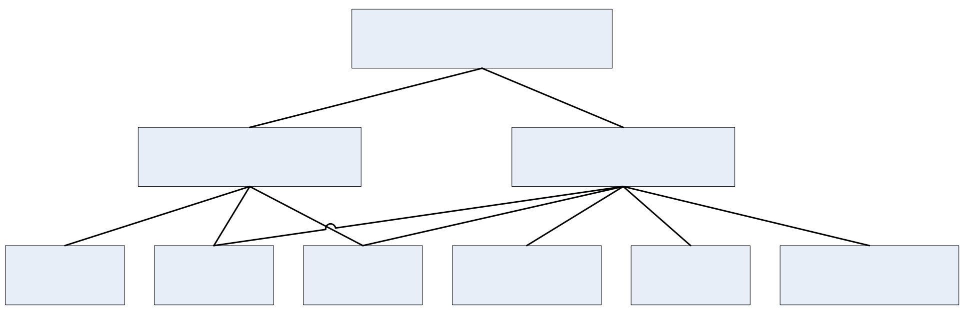

Figure-5. Behaviour hierarchy.

Start

ARPN Journal of Engineering and Applied Sciences

5. EXPERIMENTS

A series of tests have been conducted to evaluate object transportation time. Objects were placed at various positions within the robot's workspace. In each experiment, the robot was positioned in the left-adjacent corner to the light source. Room lighting was switched off during the experiments to reduce ambient light interference.

For single object transportation tests, the object was positioned in the corner opposite to the robot's start position. The objects were randomly positioned for tests requiring multiple objects to be transported. In the multiple object experiments, the retrieval of up to five objects has been tested. Each test configuration was repeated ten times. Table-4 details the time taken to transport up to five objects to the goal location. The second column of Table-4 shows the mean and standard deviation times if all objects are successfully gripped and delivered to the goal in the first attempt. Similarly, the third and fourth columns detail the mean and standard deviation times when all objects are successfully gripped and transported in the second and third attempts, respectively.

Table-4. Time taken to transport objects.

Objects Transported	**1st Attempt Successful** (sec)	**2nd Attempt Successful** (sec)	**3rd Attempt Successful** (sec)
1	56.4 ± 6.8	71.4 ± 3.4	76.7 ± 2.2
2	105.1 ± 7.7	147.3 ± 3.7	155.2 ± 3.2
3	159.4 ± 3.1	184.2 ± 5.1	219.9 ± 3.7
4	250.0 ± 4.9	264.4 ± 4.5	276.5 ± 3.6
5	275.6 ± 4.7	283.6 ± 3.9	301.9 ± 4.6

The second column of Table-4 indicates that time variation is approximately linear when transporting one to three objects. However, there is a significant time increase when transporting four objects to the goal. This indicates interference due to the presence of other objects in the workspace. The third and fourth columns of Table-4 suggest a gradual increase in time as more objects are transported. The time taken to transport multiple objects is generally lower than the integer multiple of single object transportation time. This can be expected since the object was positioned in the most distant corner from the robot in the single object experiments.

Table-5 details the average additional time taken to transport a dropped object. It takes longer to recover a dropped object when less than three objects are transported. A possible explanation for this is that the objects were positioned at a greater distance from the light source and required additional time for robot orientation correction. However, in the four and five object tests some objects were positioned closer to the goal which resulted in faster goal direction detection.

Table-5. Average additional time taken to transport a dropped object.

Objects transported	**Average extra time per object** (sec)
1	10.1 ± 3.8
2	12.5 ± 2.2
3	10.1 ± 1.0
4	3.3 ± 0.4
5	2.6 ± 0.6

6. CONCLUSIONS

A low-cost autonomous mobile robot has been developed using a VEX mobile robot kit. The base platform of the robot has been adapted to incorporate a manipulator for object handling. A variety of sensors provide input to a behaviour-based control algorithm that facilitates object transportation to a goal location. Based on initial results, the constructed robot is able to successfully locate and transport items to a goal location. Furthermore, the robot has the ability to re-grasp dropped objects. Further testing with alternative sensor configurations may improve object transportation.

The object transportation task can be extended by incorporating vision-based sensors. This can enable improved sensing of the goal location. A slightly larger version of the mobile robot base and manipulator will be fabricated to test the autonomous robot in larger environments. Another extension to this work will be to incorporate multiple robots for transportation.

REFERENCES

[1] G. C. Onwubolu, S. Aborhey, R. Singh, M. Prasad, H. Reddy, S. Kumar and S. Singh. 2002. Development of a pc-based computer numerical control drilling machine Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture. vol. 216, No. B11. pp. 1509-1515.

[2] R. V. Sharan and G. C. Onwubolu. 2006. Development of a vision-based pick-and-place robot. In: Proceedings of the International Conference on Autonomous Robots and Agents. pp. 473-478.

[3] R. Singh and G. C. Onwubolu. 2005. An integrated design towards the implementation of an automatic guided vehicle. In: Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems.

[4] R. Siegwart and Nourbakhsh. 2004. Introduction to Autonomous Mobile Robots. MA: The MIT Press.

[5] D. A. Carnegie, A. D. Payne and P. Chand. 2005. The design of a pair of identical mobile robots to investigate cooperative behaviours in Cutting Edge Robotics-Section V Multi-Robot Systems. Austria: International Journal of Advanced Robotic Systems. pp. 377-396.

[6] Engineering Services, Inc. 2009. Mobile Robots. [Available online]. http://www.esit.com/mobileRobots.php.

[7] Robo Probe Technologies, Inc. 2009. Model 2-Platform with Claw-Arm. [Available online]. http://www.roboprobe.com/catalog.aspx?productid=2.

[8] Defenders Network, Inc. 2009. MURV-100 Robot System. [Available online]. http://www.defend-net.com/robots/index.html.

[9] Mobile Robots, Inc. 2009. Mobile Robots Research and University Robots. [Available online]. http://www.activrobots.com/.

[10] D. J. Perdue. 2007. The Unofficial LEGO MINDSTORMS NXT Inventor's Guide. San Francisco CA: No Starch Press.

[11] Innovation First, Inc. 2008. VEX 2008 Inventor's Guide. [Available online]. http://www.vexrobotics.com/docs/inventors-guide/main-2008/vex-inventors-guide-07022008.pdf.

[12] R. Murphy. 2000. Introduction to AI Robotics. Cambridge MA: MIT Press.

[13] R. C. Arkin. 1998. Behavior-Based Robotics. Cambridge Massachusetts: The MIT Press.

[14] Carnegie Mellon Robotics Academy. 2008. Squarebot 2.0 Building Instructions. [Available online]. http://www.robotc.net/content/vex_quick/media/build/docs/squarebot.pdf.

[15] Innovation First, Inc. 2009. VEX Robotics Design System-VEXplorer. [Available online]. http://www.vexrobotics.com/vex-robotics-vexplorer.shtml.

[16] Innovation First, Inc. 2009. VEX Robotics Design System-Microcontroller Specifications. [Available online]. http://www.vexrobotics.com/vex-robotics-microcontroller.shtml.