International Journal of Latest Research in Science and Technology ISSN (Online):2278-5299
Volume 2, Issue 5: Page No.42-44 ,September-October 2013
https://www.mnkpublication.com/journal/ijlrst/index.php

SPECIFICATION OF BEHAVIOR ALGORITHMS FOR THE ROBOT NXT USING LeJOS

[1]Vicente Paul Saldivar-Alonso, [2]Edgar Tello-Leal, [3]Jorge Alejandro Carreon-Gutierrez, [4]Tania Yukary Guerrero-Melendez
[1,2,3,4] Faculty of Engineering and Science Autonomous University of Tamaulipas Victoria, Tamaulipas, Mexico

Abstract- The mobile robotics and autonomous systems are topics of great interest in several research areas. The advances in this field have allowed some sectors of society can use robotic devices or autonomous systems. In industry, robots are used for welding, transporting, assembling and painting pieces. In medicine, sophisticated robots help to conduct complex surgeries. In the military sector, robots are used to explore hostile areas. Therefore, in this paper we propose a behavior algorithm defined with LeJOS programming language, for implementation in the LEGO Mindstorms NXT 2.0 robots. This behavior algorithm uses the ultrasonic sensor to detect objects in the route of the robot, and then calculate the distance between the robot and the object detected. Then, with the recollected data, the robot performs a scan on a predefined angle in search of objects adjacent the identified object. The robot, by running the behavior algorithm, will determine the dimensions of the detected objects and the available spaces between objects (obstacles) to determine the best route to follow to reach a target destination.

Keywords – Behavior algorithm, LeJOS, robot, LEGO Mindstorms NXT.

INTRODUCTION

Research on autonomous displacement in robots has been a topic of great interest in recent years. The requirement to provide an effective method on the robot that allows it to establish routes and paths based on their environment to effectively and efficiently. This is a research topic due to its complexity, caused by the high number of parameters that require consideration to determinate the best route, and as a consequence, make the best decision.

An important aspect is the definition and selection of the method used by the robot, in order to recognize their environment [1]. Then, analyze collected data and based on this data to make a decision. Specifically, which route will execute the robot when presented obstacles (objects) that block to reach its target destination [2]. These actions are the first step to perform complex tasks.

In this paper we propose the specification of a behavioral algorithm that enables the robot to make a decision its path to follow, and select a new route based on overcoming obstacles detected by the ultrasonic sensor, in a range of coverage over the path traced. In the algorithm a method is defined to detect possible obstacles or objects, and determine the dimension of detected objects. Furthermore, using data collected by the ultrasonic sensor, the robot analyzes the possibility to continue over the same path or create a deviation in the current route. Then, the robot make the best analysis based on behavior algorithm to establish the route and continue with its target path.

MATERIALS AND METHODS

The definition of behavior algorithms for robots is a strategy that allows to observe the results immediately. In this work, the behavior algorithms defined have been tested on the LEGO Mindstorms NXT 2.0 robot. The main technical features are that composed of a brick or mini-computer with an Atmel 32-bits processor ARM7 48MHz, a flash memory of 256KB and a 64KB RAM storage capacity, with 4 input ports that are used to connect the different sensors (in this case an ultrasonic sensor), and 3 output ports which are used to interconnect the servo-motors [3].

The sensors provide information about the environment in which the robot is located, i.e. the information of objects near the robot. On the robot construction, an ultrasonic sensor was installed. This sensor measures the distance between the sensor and an object [4]. The ultrasonic sensor sends out high-frequency sound waves and measures the time it takes for them to be reflected, which allows it to determine the object's distance from the sensor [5].

The information collected by the ultrasonic sensor is interpreted by a software application developed using object-oriented language LeJOS [6]. This programming language, supports recursive functionalities, vectors and multidimensional arrays, synchronization, threads manipulation, exceptions, among others functionalities [7]. In order to development the proposed software application we use the IDE Eclipse [8], in order to take advantage of a well-known development environment and the Eclipse platform extension mechanisms [9].

RESULTS AND DISCUSSION

The robot's behavior is composed of different states and actions to be executed depending on the context or the world where the robot is located. The information allow trigger an

Publication History
Manuscript Received : 25 October 2013
Manuscript Accepted : 28 October 2013
Revision Received : 29 October 2013
Manuscript Published : 31 October 2013

action is collected by the sensors, wherein the data represents the objects (and their characteristics) which surround the robot NXT in that context. These actions enable switching from one state to another state.

Figure 1 shows the states defined in the algorithm that executes the NXT robot to determine the best path through the detection of objects, dimensions, calculating the available space between the objects and creates a new route for the robot.

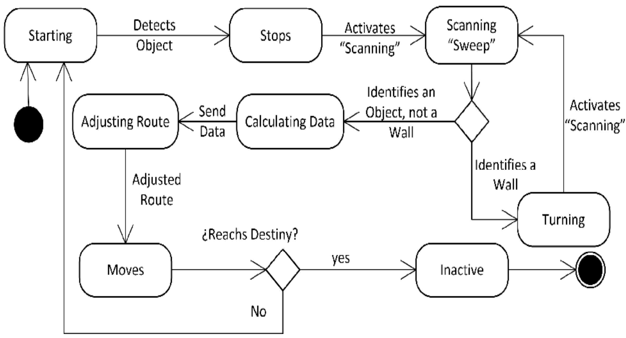

Fig. 1. Defined state diagram for the NXT robot behavior.

The defined behavior algorithm is composed of a set of actions, which are described below using pseudo code. In the first action, is executed predefined path in the robot, with the activated sensors allow the robot to detect its context. This process is performed exploring and continuously scanned in a straight line the robot vision, through the use of the ultrasonic sensor.

The ultrasonic sensor operates continuously, which allows monitoring the distance of any object located within the range of coverage. Then, when a possible obstacle (object) it detected, a command to stop is sent to robot's servomotors. In Figure 2 illustrates the pseudo code method that is called when an obstacle is detected.

```
algorithm object_detected()
    list_obstacles
    robot_size
    stops()
    list_measures = realice_scan()
        threshold =  size of the measures * detection range
         for all the measure elements do
              sum the elements
         end for
         if the sum of the elements is bigger than threshold
              list_obstacles = generate_objects(list_measures,
                                   threshold)
              calculate_movement(list_obstacles,robot_size)
         else
              turns()
         end if
```

Fig. 2. Defined pseudo code to detect an obstacle.

The second action of the behavior algorithm allows to measure an object using the ultrasonic sensor, to make possible that action, the sensor executes a horizontal scanning on a 90 degrees to left, and then, to the right of the point of detection, as illustrated in Figure 3.

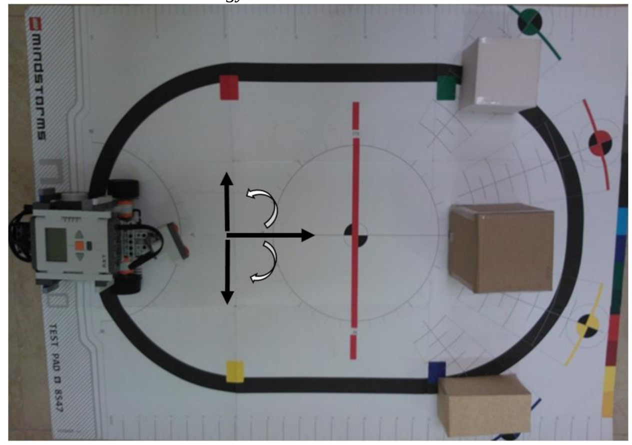

Fig. 3. The NXT robot executing the proposed behavior algorithm.

From this operation, a list of ***n*** elements is created (A_1, A_2, A_3, ... A_n). This list stores distance information and rotation degrees, provided by the motor tachometer. The one that make possible the movement of the robot. This establishes the approximate size of the object and its midpoint. In addition, the rotation angle data collected allows make adjustments on the robot path. Figure 4 shows an excerpt from pseudo code in which describe the action mentioned.

```
algorithm generate_objects(list_measures,threshold)
    list objects
    i = 1
    for all the list_measure elements
          if the elements i < threshold
               startobject = value on the element i
               j = i + 1
              follow = true
              while follow
                    if the element j > = threshold
                    endobject = value on the j element
                          i = j + 1
                          follow = false
                    end if
                           increase j
               end while
               object = generates_obstacles(startobject,endobject,
                         startangle,endangle)
               adds object to the list
            end if
       end for
return object_list
```

Fig. 4. Pseudo code that allow calculates the size of an object detected by the robot ultrasonic sensor.

In the third action, in the case that the ultrasonic sensor readings detected below the threshold ***T***, on both sides, indicating that is not possible to cross through this area. On the other hand, in case that the sensor detects superior readings to the threshold, it assumes that is possible to continue on that route. For which purpose, it uses readings collected before, and sets the location of objects, based on the data set delimited by readings superior the threshold.

Then, is established the midpoint of the object and traces a circumference from itself, this allows to know the approximate size of the object. Then, the algorithm analyze the spaces between the different objects. After carry out the calculation the spacing between objects, is selected larger space to allow the crossing of the robot, as shown in Figure 5.

```
algorithm calculate_movement(list_objects,robot_size)
   i = 1
   temp = 0
   valueMin
   while exists elements on the list
      start = extracts endobject from element i
      end = extracts startobject from element i + 1
      temp = end – start
      if valueMin > end – start
         min = temp
         objectA = element i
         objectB = element j
      end if
      increments i
    end while
    if min > robot_size
       movement(objectA angle,objectB angle)
    else
       turns()
    end if
```

Fig. 5. Pseudo code used to calculate the size of the object and detect the next route for the robot.

CONCLUSIONS

In this work we proposed the definition of a behavior algorithm, as well as its implementation using the programming language LeJOS for Mindstorms NXT 2.0 robots, with functionalities that allow identify objects close to robot, determinate their dimensions, and calculates the space available between them. Then, using the defined behavior algorithm, the NXT robot can resolve the blind spot problem over it. In addition, this algorithm enables to determine the best route, allowing the robot to chose among possible pathways.

The first results show that it is necessary to make some adjustments the structure of the robot for efficient execution of the behavior algorithm, this due to the quality of the data collected by the ultrasonic sensor. In some of the early tests show inconsistencies in choosing the best route. In addition, in some cases, the calculations generated by the algorithm shows that it is not possible to continue on a given route, even if there is enough space to cross and let the robot to set a new path.

This is due to the ultrasonic sensor sends wrong readings as a consequence of the robot movement and the roughness (not flexible) of the data cable that connects the ultrasonic sensor with the brick or minicomputer NXT 2.0. Therefore, in the future work, the purpose will be to find solutions to the detected problems over the behavior algorithm presented in this research work.

REFERENCES

[1] J.F. Kelly. *LEGO Mindstorms NXT-G Programming Guide.* Apress-Springer, New York, USA. 2010.

[2] E. Tello-Leal, T.Y. Guerrero-Melendez, and V.P. Saldivar-Alonso. Revisión de la Plataforma Robótica LEGO Mindstorms para Aplicaciones Educativas y de Investigación. *Revista S&T.* Vol 11(26), pp. 9-27, 2013.

[3] LEGO Group. *LEGO Mindstorms Robot*. LEGO Group. Internet: http://mindstorms.lego.com/en-us/default.aspx, 2012 [August 24, 2012].

[4] D.J. Perdue and L. Valk. *The unofficial LEGO Mindstorms NXT 2.0 Inventor's Guide.* N Starch Press Inc., San Francisco, CA, USA, 2011.

[5] T. Griffin. *The art of LEGO Mindstorms NXT-G Programming.* No Startch Press Inc., San Francisco, CA, USA. 2010.

[6] LeJOS NXJ. *LeJOS Java for LEGO Mindstorms.* Internet: http://lejos.sourceforge.net/index.php, 2012 [October 24, 2012]

[7] J.A. Breña Moral. *Develop LeJOS Programs Step by Step.* JAB Press. Madrid, España. 2009.

[8] Eclipse. *Eclipse Platform*. Internet: http://www.eclipse.org, 2004, [September 09, 2013].

[9] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks. *EMF: Eclipse Modeling Framework.* Addison-Wesley Professional, Boston, MA, USA, 2008.