120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013

Paper ID #6116

Undefined Obstacle Avoidance and Path Planning

Prof. Akram Hossain, Purdue University, Calumet (Tech)

Akram Hossain is a professor in the department of Engineering Technology and director of the Center for Packaging Machinery Industry at Purdue University Calumet, Hammond, IN. He worked eight years in industry at various capacities. He is working with Purdue University Calumet for the past 24 years. He consults for industry on process control, packaging machinery system control and related disciplines. He is a senior member of IEEE. He served in IEEE/Industry Application Society for fifteen years at various capacities. He served as chair of manufacturing Systems Development Applications Department of IEEE/IAS. He authored more than 25 refereed journal and conference publications. In 2009 he as PI received NSF-CCLI grant entitled A Mechatronics Curriculum and Packaging Automation Laboratory Facility. In 2010 he as Co-PI received NSF-ATE grant entitled Meeting Workforce Needs for Mechatronics Technicians. From 2003 through 2006, he was involved with Argonne National Laboratory, Argonne, IL in developing direct computer control for hydrogen powered automotives. He is also involved in several direct computer control and wireless process control related research projects. His interests are in the area of industrial transducer, industrial process control, modeling and simulation of Mechatronics devices and systems, wireless controls, statistical process control, computer aided design and fabrication of printed circuit board, programmable logic controllers, programmable logic devices and renewable energy related projects.

Undefined Obstacle Avoidance and Path Planning of an Autonomous Mobile Robot in a Two-Dimensional Workspace

Abstract:

Design and development of robotics applications has always been an ever advancing area of study in the field of modern automation technology. The concept of introducing mobile robots in familiarizing engineering technology students with various robotics problem solving approaches, is widely being implemented for educational purpose nowadays. Path Planning and Collision avoidance are two of the most common theories applied for designing and developing advanced autonomous robotics applications. The objective of this study was to develop a graphical programming model for the National Instrument's (NI) Robotic Starter Kit (DaNI) in NI LabView Robotics Platform that incorporates mobile robot path planning and obstacle avoidance concepts. The LabView program enables the mobile robot to travel from a starting point to a user desired destination point, avoiding undefined obstacles on its route. This paper encompasses the scope to educate engineering technology students with the robotics "Sense, Think and Act" approach where the robot senses for random obstacles on its path via an ultrasonic sensor, makes a decision based on a non-colliding threshold distance in order to execute collision avoidance routine and returns to the process of reaching the predefined destination point. The motion of the robot is controlled by two pairs of mechanically geared wheels driven by two DC Servo Motors.

Introduction:

Robotics has become one of the essential segments of modern automation systems. Autonomous mobile robots are noticeably being used in various industrial and non-industrial applications nowadays. The purpose of this project was to study the National Instrument's Robotic Starter Kit 1.0 (DaNI) and to develop a graphical programming model in LabVIEW Robotics Module 2011 for combining mobile robot path planning and collision avoidance concepts. The LabVIEW program enables the mobile robot to travel from a user defined starting point A to a destination point B through avoiding undefined obstacle on its route. The LabVIEW program is based on the "Sense, Think and Act"[1] concept where the robot senses for obstacles in its route by an ultrasonic sensor, makes a decision to execute collision avoidance routine and returns to the process of reaching the destination point. The motion of the robot is controlled by two pairs of mechanically geared wheels driven by two DC Servo Motors. The figure below shows an example of the path of travel for the mobile robot:

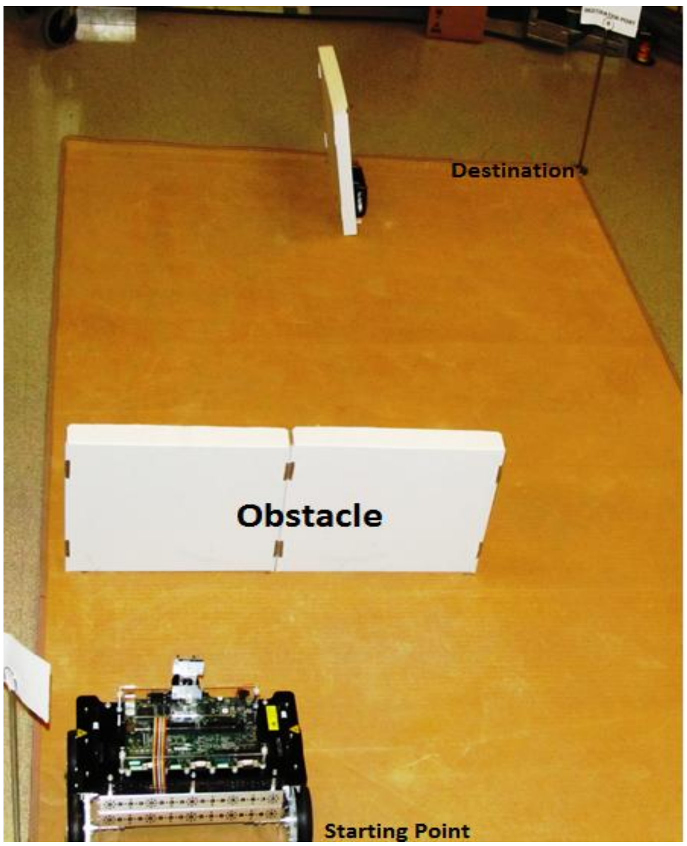


Figure 1: The Path of Travel for the Mobile Robot.

NI Robotic Starter Kit 1.0:

National Instrument's Robotic Starter Kit 1.0 is an active learning module that helps to build robotics fundamentals such as sensor characterization, motor control, kinematics, path planning, data acquisition and other industrial robotics concepts. NI Robotic Starter Kit 1.0 includes Pitsco TETRIX Frame, four wheels driven by two Servo Motors and a Single-Board RIO (reconfigurable IO) Processor. The hardware components of a NI Robotic Starter Kit 1.0 are presented by the following block diagram[2]:

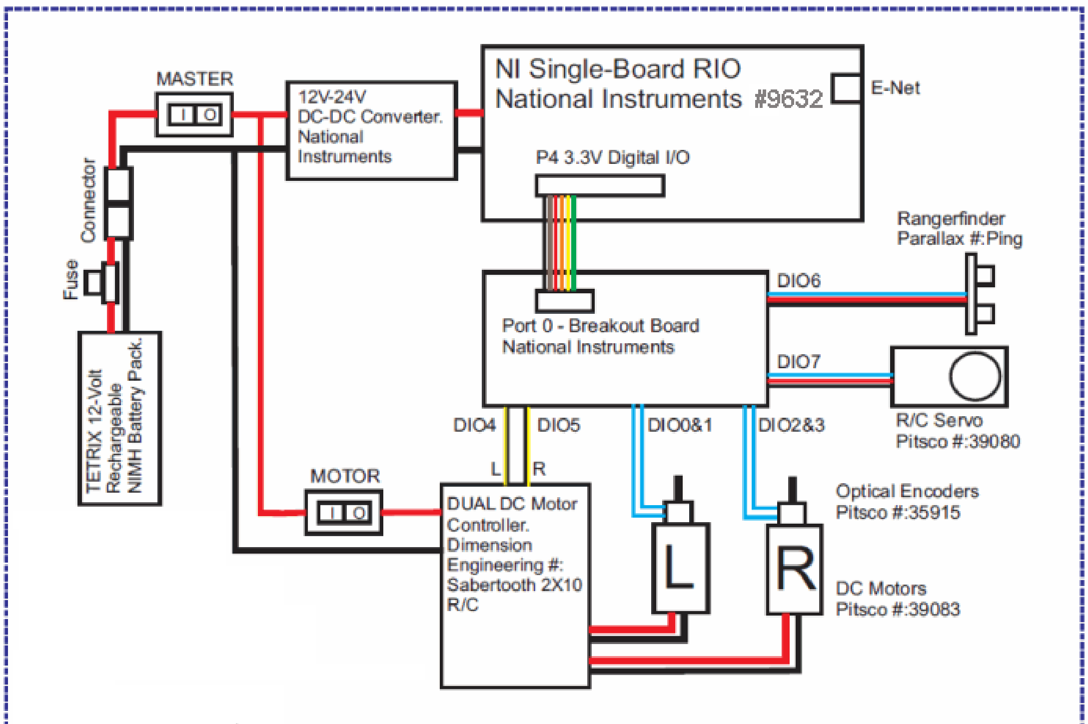


Figure 2: NI Robotic Starter Kit 1.0 Hardware Components.

National Instrument's Single Board Reconfigurable IO comprises of a real-time processor, fpga (field programmable gate array) module and analog and digital I/O in one board. The fpga module examines all the logical conditions from the analog and digital inputs and provides outputs. Depending on the outputs the real-time processor processes the output data and defines robot's kinematics. The Single Board RIO components are presented in the figure below[2]:

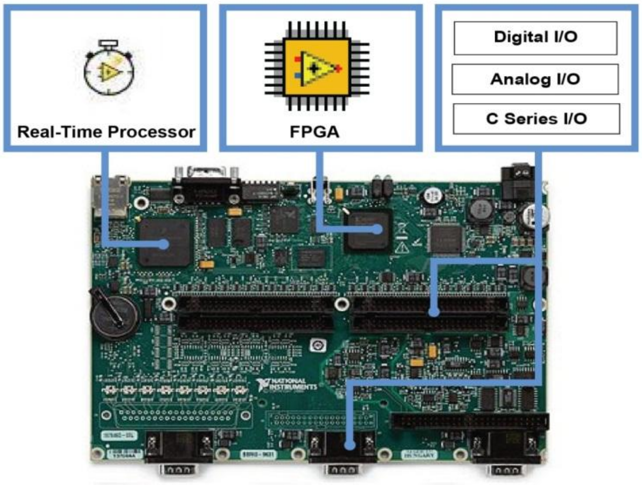


Figure 3: Single-Board Reconfigurable IO Components.

The ultrasonic sensor integrated with the Robotic Starter Kit acquires data about obstacles by transmitting a short pulse of ultrasonic energy (typically for 200µs with 40kHz)[1]. The sensor then stops transmitting energy and waits for a reflected signal from the obstacle in front of it. Once the sensor receives the transmitted signal it provides an output pulse to the real-time processor. Below the ultrasonic sensor with transmitted and reflected energy is shown:

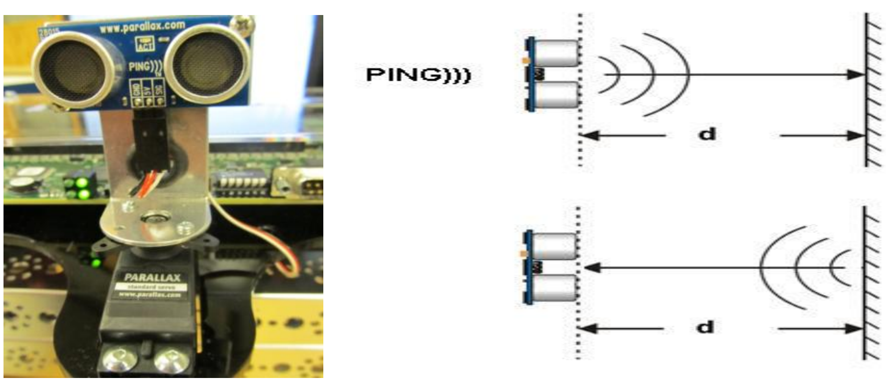


Figure 4: Ping))) Ultrasonic Sensor.

Based on the outputs generated by the ultrasonic sensor, the real-time processor provides a pulse width modulation (PWM) signal to the drive motors for the robot. The velocity of the motors can be defined in terms of both rotation per minute (RPM) or radians per second. Figure below shows the servo motor and wheel of a Robotic Starter Kit 1.0:

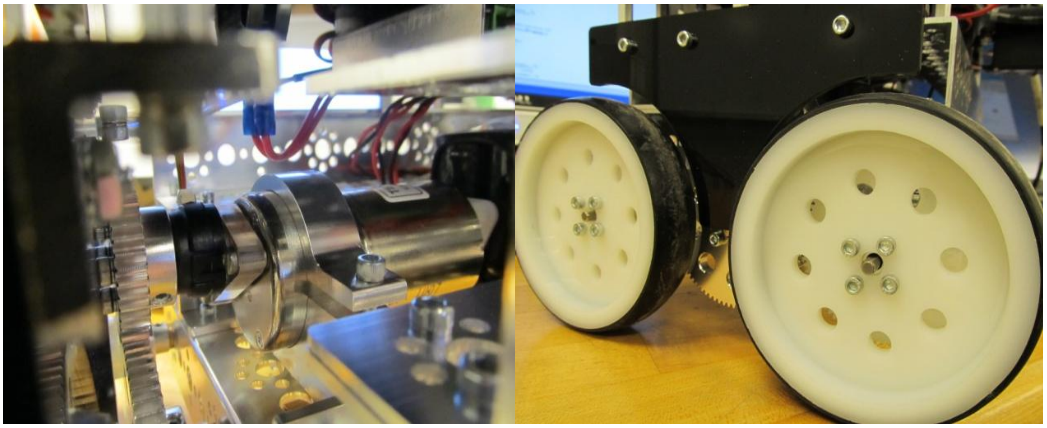


Figure 5: Drive Motors of a Robotic Starter Kit 1.0

NI LabVIEW Robotics Module 2011:

National Instrument's LabVIEW Robotics Module 2011 is a graphical programming language that comprises blocks for ease of understanding. The blocks contain all the necessary input and output parameters required for various types of actions[1]. LabVIEW has two windows: block diagram for writing all the codes in a program and a front panel for displaying all the outputs and for providing user control. In Robotics Module 2011 there are built-in drag and drop blocks available for Robotic Starter Kit 1.0. Robotics →Starter Kit→Starter Kit 1.0, the functions blocks for initializing robot, writing/ reading values for motor control and writing/reading values for ultrasonic sensor control are present. Loops are used for closed loop controls. A picture of the functional blocks for Robotics Module is given below:

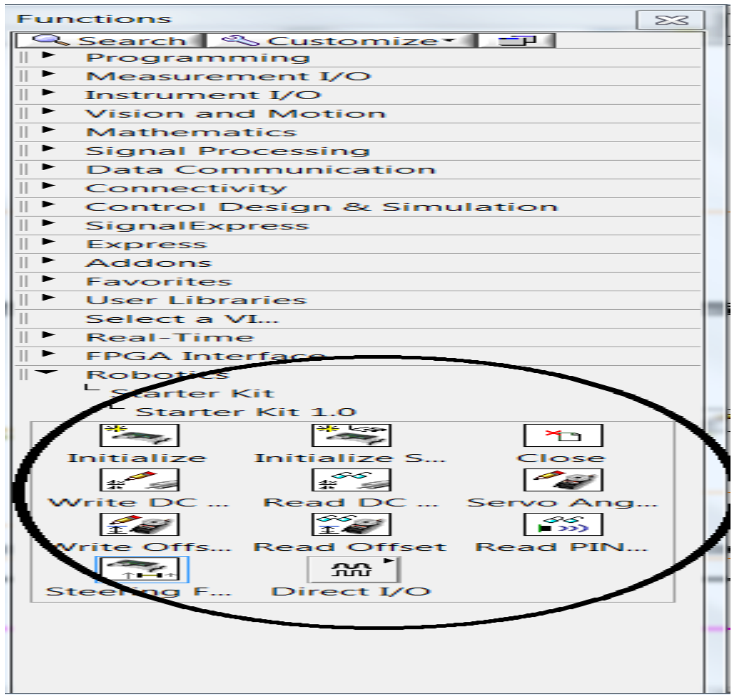


Figure 6: Functional Blocks for NI Robotics Module.

NI LabVIEW Robotic Starter Kit Collision Avoidance:

The ultrasonic sensor of the robot continuously generates outputs whenever any object is present in front of it. The ultrasonic sensor is mounted on a servo motor. The rotational angle (radian) of the servo motor can be defined by the user. The rotation of the motor will be the field of view (FoV) for the sensor. Depending on the time for the reflected signal, the sensor provides the distance (meter) of the object from the robot. In LabVIEW Robotics Module 2011, the following functional blocks are used for reading object distance from the ultrasonic sensor[2].

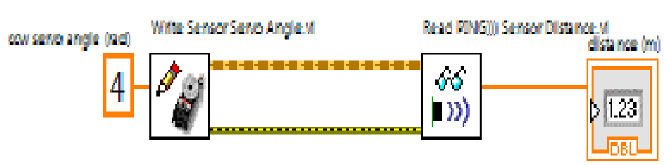


Figure 7: Functional Blocks for Sensor Output.

The collision avoidance concept includes a threshold distance (meter). A threshold distance is something that helps the robot to determine whether an object should be considered as an obstacle in its route or not. In other words, a threshold is a non-colliding distance for the robot.

The real-time processor compares the output of the ultrasonic sensor with the threshold distance. If the ultrasonic sensor output distance is above threshold distance then the robot continues travelling forward. Whenever the sensor output goes below the non-colliding distance, the robot performs the collision avoidance routine in order to avoid any collision with the object in front of it. A simple collision avoidance conditions in LabVIEW is given below:

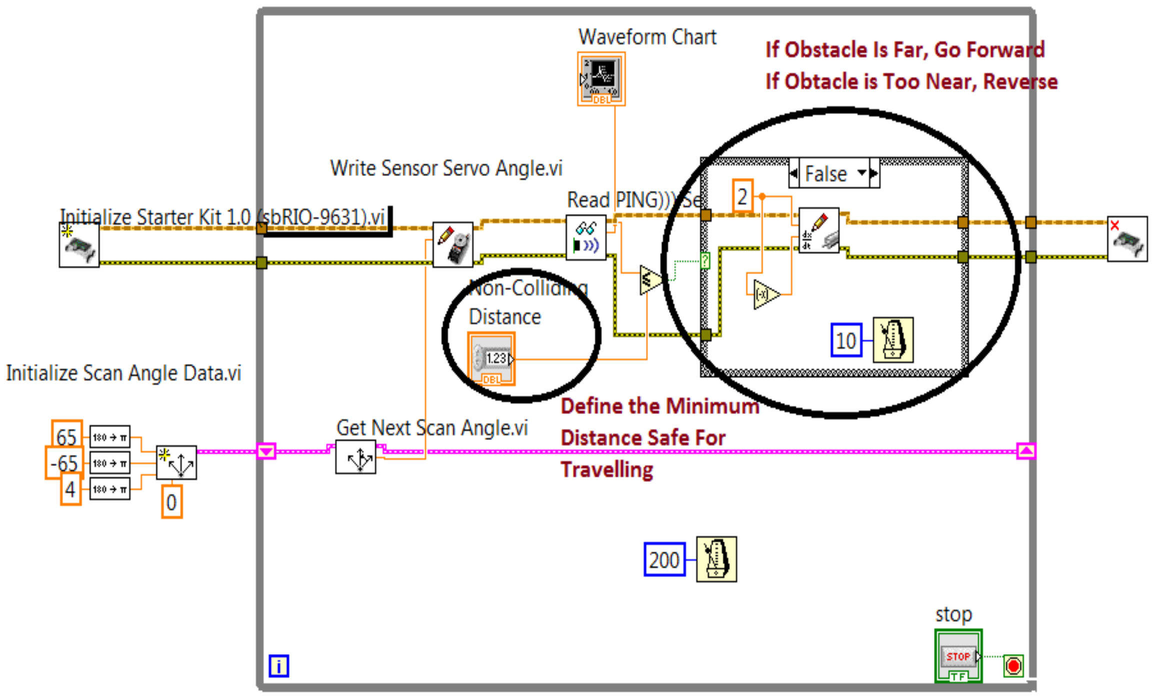


Figure: NI Labview Simple Collision Avoidance Routine

Figure 8: NI Robotic Starter Kit 1.0 Collision Avoidance.

NI LabVIEW Robotic Starter Kit Path Planning:

In the LabVIEW environment, the movement of the NI Robotic Starter Kit 1.0 is performed through defining CCW (counter clock wise) set point velocity to the Servo Motors. Depending on the positive and negative CCW velocity the robot moves in forward, reverse or rotational direction. The Figure below shows the functional block for drive servo motor with combinations of motor velocity for different directions:

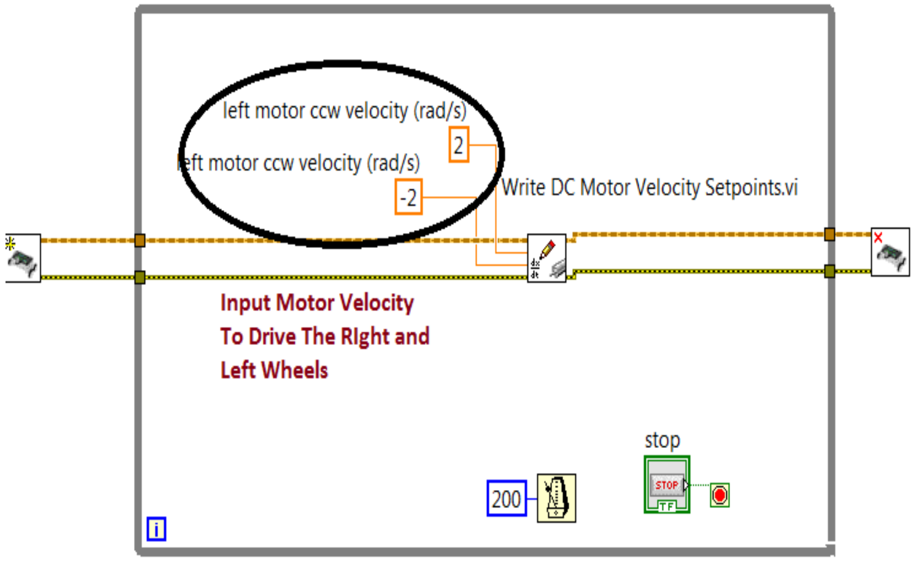


Figure: NI Labview Program For Robot's Motion Control

Motor Velocity	Left	Right
Forward	(+) ve	(-) ve
Reverse	(-) ve	(+) ve
Turn Right	(+) ve	0
Turn Left	0	(-) ve

Figure 9: NI Robotic Starter Kit 1.0 Motor Control.

A robot's path can be a predefined program by a user by providing different combinations of set point velocities to the drive motors for certain amount of travel time. A simple path planning routine is given below:

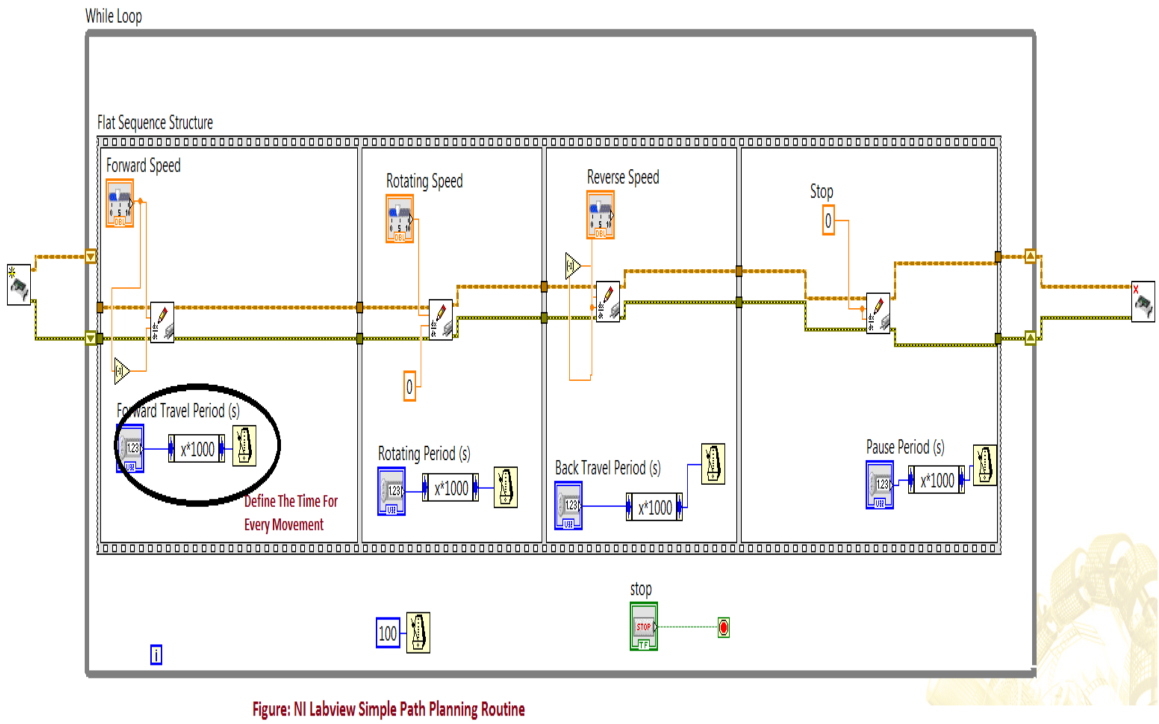


Figure 10: NI Robotic Starter Kit 1.0 Path Planning.

Project Problem Solving Methodology:

The LabVIEW programming approach for this project was mainly based on the required travel time in X direction and Y direction for a certain velocity in order to reach the destination. In order to solve the program for reaching a destination by successfully avoiding undefined obstacle on the way, two types of variable were considered in this research: independent variables and dependent variables. The independent variables in this program are predefined by the user. These are: sensor's FoV (field of view), the non-colliding distance, the travel time in X direction, the travel time in Y direction and travelling speed. Whenever the robot determines an obstacle, the total travel time in both the directions increases as there are some time elapsed due to performing the collision avoidance routine. The travel time in this research is represented by the number of iteration of the while loop. The logical approach for solving this problem is given below:

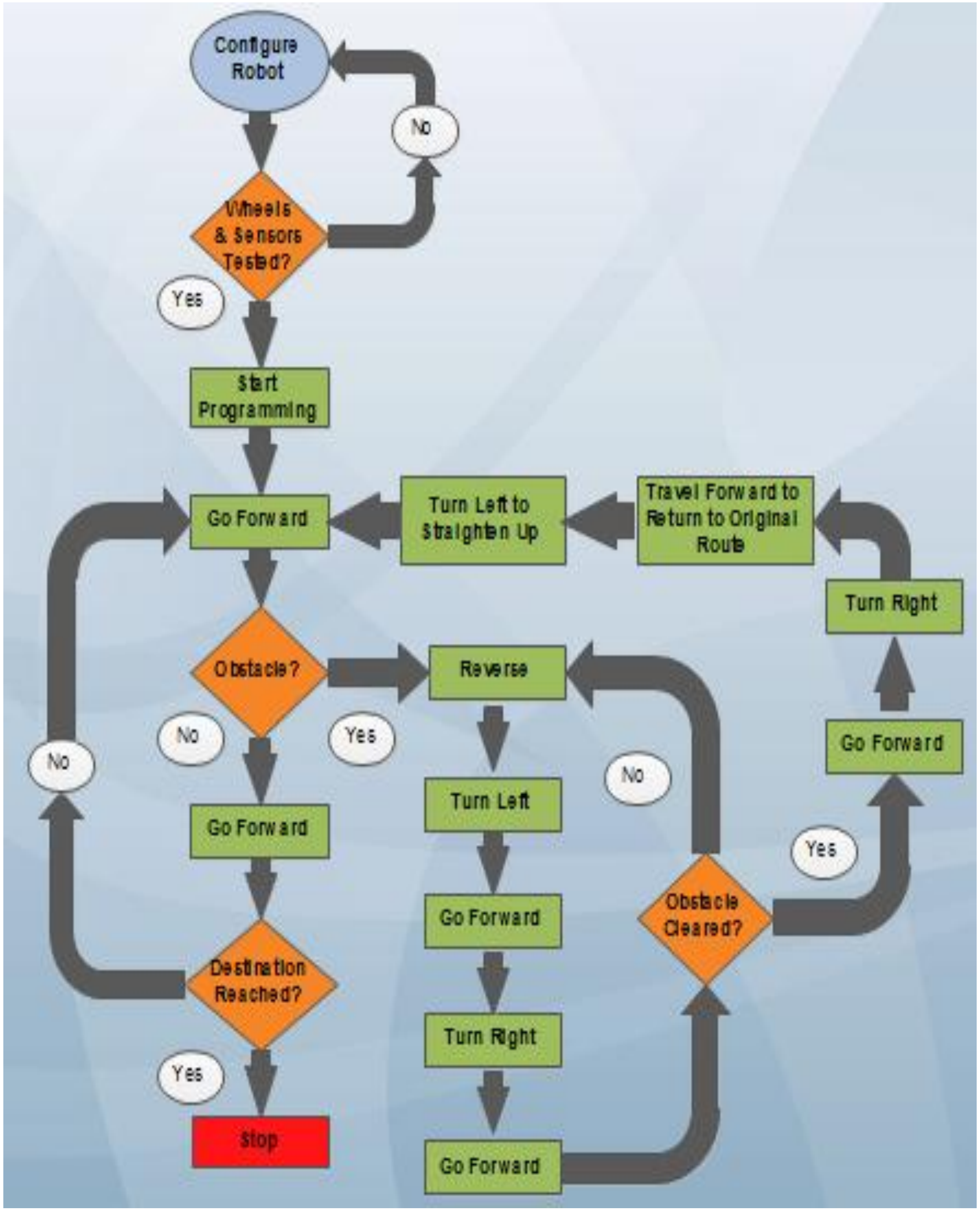


Figure 11: Process Flow Diagram.

In the collision avoidance routine, every time the robot detects an obstacle; it reverses, goes in the right hand direction and then tries to reach its original path of travel. However, it keeps of performing the collision avoidance step until it sees a clearance. All movements in this case are time dependent. The collision avoidance segment of this research is given below:

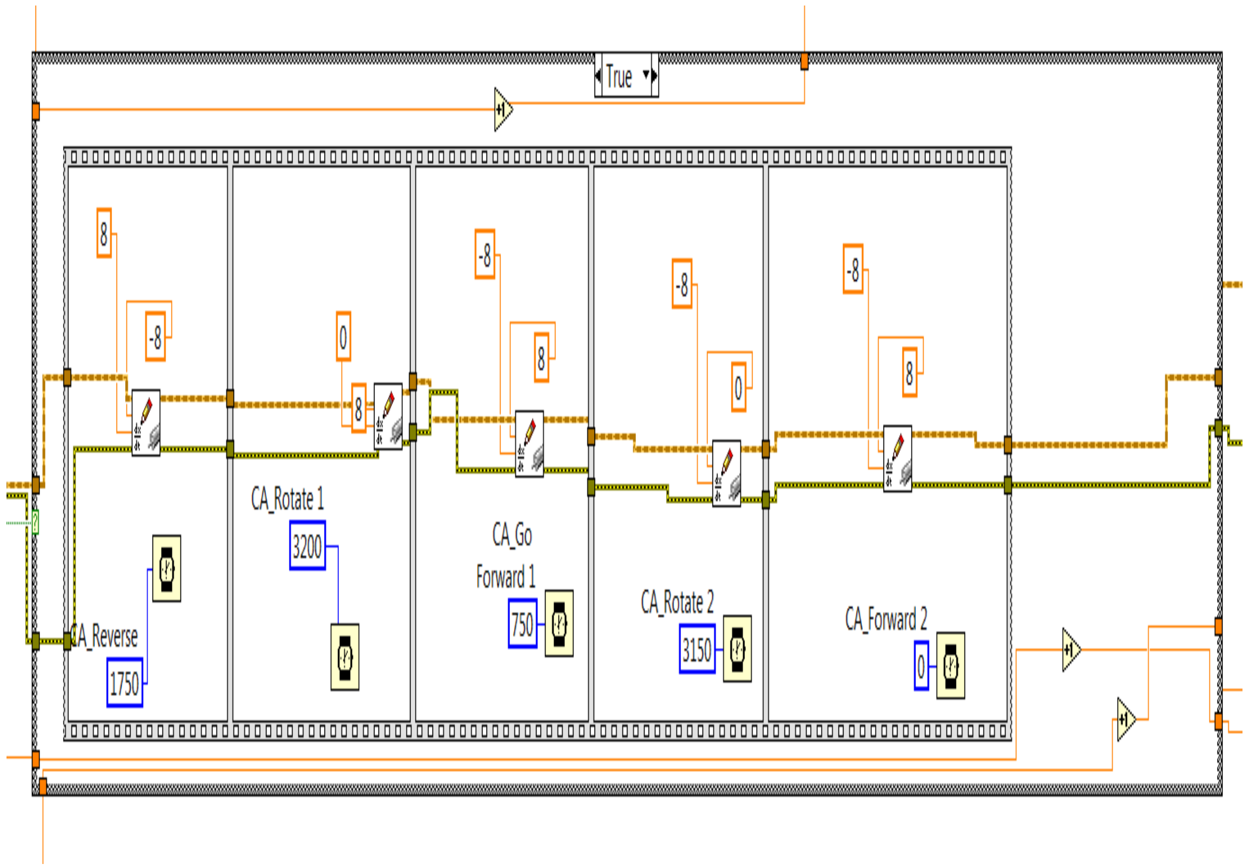
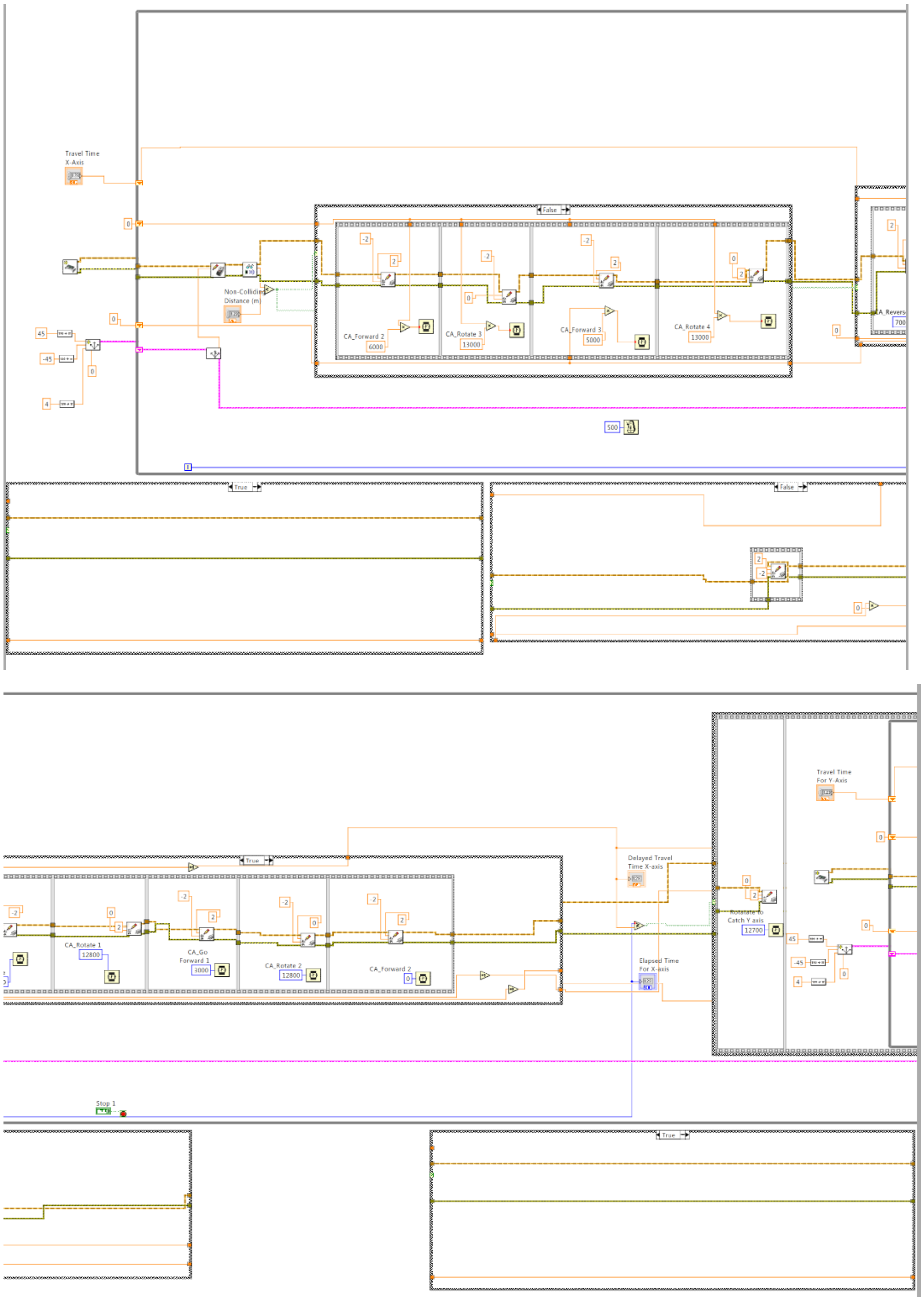


Figure 12: Undefined Obstacle Avoidance Routine.

Two significant considerations were taken into account in case of undefined obstacle during this research. Firstly, In order to continue travelling after an obstacle is avoided through collision-avoidance routine, the program should return to main execution loop with additional delay in travel time (loop iteration). Otherwise, the robot would stop before reaching its destination. This has been performed by considering increment of an additional iteration every time there is a collision detected.

Secondly, when the robot encounters an obstacle while travelling vertically, it obtains additional horizontal motion in order to avoid a collision. After the collision is avoided the robot must travel the same horizontal distance in reverse direction for returning to its original path of travel. This has been achieved by considering a variable outside of the main execution loop which counts up the reverse horizontal iteration from the time of detecting an obstacle till finding a clearance. After the robot overcomes the obstacle, the variable goes back to zero. In LabVIEW this is one smart way of coming out of a loop execution after certain condition.

The entire program for the project is given below:


Travel Time
X-Axis
Non-Colliding
Distance (m)
False
CA_Forward 2
6000
CA_Rotate 3
13000
CA_Forward 3
5000
CA_Rotate 4
13000
CA_Reverse
700
500
True
False
True
CA_Rotate 1
12800
CA_Go
Forward 1
3000
CA_Rotate 2
12800
CA_Forward 2
Delayed Travel
Time X-axis
Elapsed Time
For X-axis
Rotatate to
Catch Y axis
12700
Travel Time
For Y-Axis
Stop 1
True

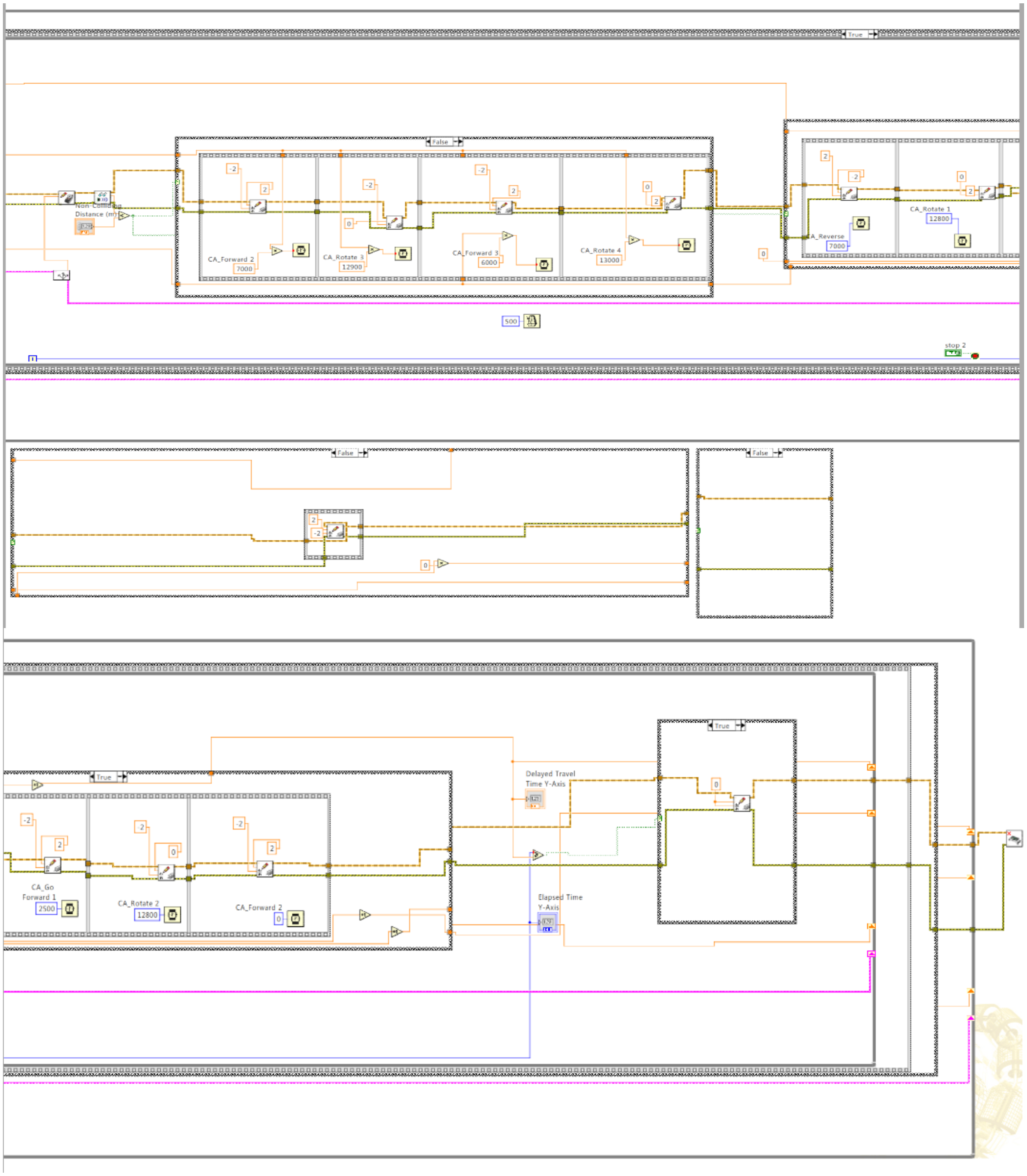


Figure 13: Undefined Collision Avoidance Routine.

Results:

As mentioned earlier, for different velocities the time period (ms) for each rotational movement in the collision avoidance routine varies noticeably. In this research we have performed the motion for 2, 4, 6 and 8 radian per second. The following results were obtained for different rotational time:

Velocity (rad/sec)	First Right Turn (ms)	First Left Turn (ms)	Second Left Turn (ms)	Second Right Turn (ms)
2	12800	12800	13000	13000
4	6400	6300	6500	6500
6	4267	4267	4300	4334
8	2600	3250	3225	3250

Figure 14: Time Period (ms) for Different Rotational Velocity

Conclusion:

The collision avoidance and path planning concepts for autonomous mobile robot focused in this research can play significant role in real life applications. In an industrial cleanroom environment, special clothing of human operators does not completely avoid contamination. In this case a mobile robot can perform the tasks inside a cleanroom reducing the risk of any contamination from outside.

In non-industrial applications such as emergency situations or space mission where it can be risky for humans, mobile robots can be used to collect information and data for rescue or scientific research.

The purpose of this study was to come up with some fundamental programming concepts with LabVIEW robotics programming environment for the mechatronics students.

Reference:

1. www.ni.com
2. **“Mobile Robotics Experiments with DaNI”-** By Dr. Robert King, Colorado School of Mines.