Navigation system for an automatic guided vehicle

E M Ngandu1, N Luwes2 and K Kusakana3

1,2 Department of Electrical, Electronics, and Computer Engineering.
3 Central University of Technology, Free State, Bloemfontein, South Africa.
Email: engandu@cut.ac.za1, nluwes@cut.ac.za2, kkusakana@cut.ac.za3

Abstract. An automated guided vehicle (AGV) is a robot that could use wires or indicators on the floor to navigate itself in an environment. These AGV’s use different types of guidance system depending on the application for which they are intended. Literature indicates that continues development have been done to improve those systems. In this project, the author is developing a cost effective and reliable multi-path following guidance system. It makes use of the NI Robotic Starter kit that is fitted with 4 digital infrared sensors to follow multiple paths made of a black tape on a white surface. The author programmed the AGV using a state machine where the middle 2 sensors are used to follow the line and the outer 2 sensors are used to control which path to follow. The results obtain proved the effectiveness of the developed guidance system as the AGV is effectively able to follow multiple paths while reducing the number of sensors needed to accomplish it.

1. Introduction
An automated guided vehicle (AGV) is a robot that could use markers or wires in the floor to navigate itself in a particular environment. It may also use vision, magnets, or lasers for navigation depending on its application. They are frequently employed in industrial applications to move materials from one place to another in a manufacturing facility or a storeroom [1]. Automated guided vehicles (AGVs) increase the productivity and reduce costs by facilitating the automation of a manufacturing plant or storeroom. Automated Guided Vehicles have a wide variety of applications in which they are mostly used to transport many different types of material including pallets, rolls, racks, carts, and containers [2] [3].

2. Hardware and Software review

The development of the system proposed in this article requires a review of the hardware and software that can be used in this project.

2.1. AGVs Guidance system
The aim of AGVs guidance system is to keep the AGV on a predefined path. One of the main advantages of AGV is that it is simple to adjust by making use of the guidance system for altering the guide path at low cost compare to conveyors, chains, etc.
An additional benefit is that guide tracks are flexible to allow intersection on different paths because usually, a guide path does not obstruct another system. Various types of guidance systems can be selected by considering the type of AGV selected, its application, requirement, and environmental limitation [4]. The different kinds are as followed: Wire-guided, Guide tape, Inertial, Infrared, Laser, Teaching type [4].

2.2. Infrared sensor
An infrared sensor is a device that is used to detect infrared radiation that falls on it. Depending on the application, there are various kinds of IR sensors that can be built [5]. The following are some example of different types of IR sensor and their use:
- Proximity sensors: Used in Edge Avoiding Robots and Touch Screen phones.
- Contrast sensors: Used in Line Follower Robots.
- Obstruction counters/sensors: Used for counting goods and in Burglar Alarms.
2.3. **NI LabVIEW Robotics Starter Kit**

NI LabVIEW Robotics Starter Kit is a fully assembled mobile robot base starter kit with an ultrasonic sensor, encoders, DC motors, and a 12V battery. It has a controller based on NI Single-Board RIO with real-time decision making, FPGA-based I/O processing and analogue and digital I/O on a single board. It can easily connect to a variety of robotic sensors and actuators [6].

2.4. **C Sharp (C#)**

C# is a simple, modern, object-oriented, and type-safe programming language that combines the high efficiency of fast application development languages with the raw power of C and C++ [9]. The language provides support for software engineering principles like strong type examination, array bounds testing, detection of attempts to utilize uninitialized variables, and automatic garbage collection [9].

2.5. **LabVIEW**

LabVIEW is a highly creative development environment for producing custom applications that interact with real-world data or signals in fields such as science and engineering. The net result of using a tool such as LabVIEW is that advanced quality projects can be accomplished in less time with a smaller number of people involved. LabVIEW is exceptional because it makes this wide variety of tools accessible in a single environment, making certain that compatibility is as simple as drawing wires between functions.
LabVIEW itself is a software development environment that comprises several components, a number of which are required for any type of test, measurement, or control application [10] [11]. This project makes use of a Guide tape guidance system to design a manufacturing environment. In this environment, the AGV is required to follow a black tape using digital line following infrared sensor. The manufacturing environment is divided into 3 separate area (2 assembly lines locations and a storeroom) that are connected using the black tape as shown in Figure 3.

3. METHODS

The system’s layout shown in Figure 3 is of the introductory section that shows an AGV that collects parts from the storeroom and delivers them to the assembly lines. The author has selected the NI LabVIEW Robotics Starter Kit to be used has the AGV. It will be fitted with line follower sensors and programmed to follow a black line on a white surface. The AGV will be autonomously capable of sensing its environment and make decisions depending on the commands received from the master control application. The process followed by the worker is divided into four different steps:

- Communication
- Sensing
- Decision
- Action

Table 1. Behaviour of a two switch on the AGV

| Switch 1 | Switch 2 | Path follow by the AGV from |
|----------|----------|-----------------------------|
| Off      | Off      | Storeroom to Assembly line 1 |
| Off      | On       | Storeroom to Assembly line 2 |
| On       | Off      | Assembly line 1 to Storeroom |
| On       | On       | Assembly line 2 to Storeroom |

3.1. Communication

The AGV is in constant communication with the master control application. It sends information such as its whereabouts and receive instructions such as what it has to do and where to go. The communication is done through a local area network in which the entire system is connected. The master control application is a simple TCP/IP client/server application written in C#. Its purpose is to send commands...
to the AGV with instructions on what path to follow. The NI LabVIEW Robotics Starter Kit has an Ethernet port that allows it to perform TCP/IP communication over a network of NI (National Instrument) device. Communication is established between the AGV (NI LabVIEW Robotics Starter Kit) and a computer network.

3.2. Sensing
The AGV uses infrared sensors to follow a black line on a white surface. The sensors used are digital and give an output of 1 when they are sensing the white surface and 0 when they are sensing on the black line. These sensors have a very low range in which they can operate successfully. For this reason, they will be placed at the bottom of the AGV with a distance of 10mm from the surface that they have to read. This project made use of 4 sensors and each sensor plays a specific role in the decision that the AGV make while following the line.

3.2.1. Connecting sensor to AGV
This project made use of parallel connector 5 on the NI rio board with the 4 sensors connected as shown in Figure 5.
Each sensor has 3 points that need to be connected to the AGV:
- VCC: the voltage supply of the sensor. If not connected the sensor will not work
- GND: the ground on the sensor circuit.
- Output: this is a digital bit that will allow the AGV to monitor the status of the sensor.

Figure 6 is a picture of the physical AGV showing the sensors connected to the AGV.

| Table 2. Sensors Connections |
|-----------------------------|
| Label | Pins on sensor | Connection to P5 | Pin number on P5 |
|------|----------------|-----------------|-----------------|
| R1   | VCC            | 5V              | 50              |
|      | GND            | D GND           | 48              |
|      | OUTPUT         | Port3/DIO2     | 17              |
| R2   | VCC            | 5V              | 50              |
|      | GND            | D GND           | 48              |
|      | OUTPUT         | Port3/DIO3     | 19              |
| L1   | VCC            | 5V              | 50              |
|      | GND            | D GND           | 48              |
|      | OUTPUT         | Port3/DIO1     | 15              |
| L2   | VCC            | 5V              | 50              |
|      | GND            | D GND           | 48              |
|      | OUTPUT         | Port3/DIO0     | 13              |
3.3. Decision and Action

This section describes how the commands from the masters in combination with the inputs from the sensors are translated into decisions and actions that the AGV has to perform. The first step of this section is to highlight all the possible actions that the AGV can perform. The second step will be the decisions that the AGV can take based on the commands from the masters and the input of the sensor.

3.3.1. Step 1: Action

Five different type of actions can be performed by the AGV and all these actions depend on the status of the motors. stop, move forward, move backward, turn right, turn left.

The AGV is programmed in NI LabVIEW. LabVIEW developers have provided an entire section reserved for robotic. That part of the software allows users to develop applications for a robotic starter kit. They provided users with some examples for some of the core function that the robotic starter kit can perform.

3.3.2. Step 2: Decision

The AGV is placed into an environment as shown in Error! Reference source not found., where it is required to follow the black line on a white surface from one point to the other without getting lost while doing it. From Figure 7, the AGV is set to follow one of the following paths:

- From A to B
- From A to C
- From B to A
- From C to A

Note that there is no path set for the AGV to go form B to C or from C to B because each point represents the location of components of the entire system.

- Point A: is the location of the storeroom
- Point B: is the location of the first Assembly line
- Point C: is the location of the second Assembly line

With that in mind, there is no need for the AGV to travel from one assembly line to the other. The decision process of the AGV mostly depends on the inputs of the sensors that are used to follow the line. The commands form the masters just tells the AGV where it must go.

Observe Figure 8, the AGV is placed on the line with 2 sensors on the left of the line and 2 sensors on the right of the line.
To follow the line, the AGV makes use of only 2 sensors (R1 and L1). These sensors have a default value of 1 and changes to 0 when the sensor passes over the black line. The other 2 sensors are used as check point counter. The check points next to the black line are utilized by the AGV to determine its location compared to where it is going.

Figure 9 highlights the 5 different check points that are next to the black line. The AGV receives commands from the masters over the wireless network in the form of a Byte. That byte is then converted into a number that the AGV uses to interpret the master’s command. The possible commands from the masters are as followed:

- 0: stop all motions
- 1: Go from A to B
- 2: Go from A to C
- 3: Go from B to A
- 4: Go from C to A

The AGV program is a case structured program in which each command from the master triggers a specific case that needs to be executed.

| R1 value | L1 value | Decision and action       |
|----------|----------|---------------------------|
| 0        | 0        | Stop movement             |
| 0        | 1        | Turn right                |
| 1        | 0        | Turn left                 |
| 1        | 1        | Move forward              |
Each one of the cases shown in Error! Reference source not found., is a subroutine of its own that controls the AGV allowing it to follow a specific path. For example: in CASE 1, the AGV will follow the Path from A to B. Referring to Figure 9, it has been shown that there are 5 different check points. Note the case in which the AGV follow the path from A to C and using L2 to count the check points, observe that there will be a point where L2 will cross the line that goes to B and will count it as a check point as well. The actions perform by the AGV differs at different check points, for example: in case 1 the AGV must go from A to B. At the start, the check point count is equal to zero. Therefore, the AGV uses the sensors R1 and L1 to follow the line.

- From the moment that L2 reads the first check point, the check point count will be incremented by 1 and will do so at every single check on it path.
- When the counter is equal to 1, the AGV knows that it is at the point where the path to B and C crosses and it has to keep left to go to B. Therefore, all the data coming from the sensors at the right side of the line are ignored.
- After keeping left, the AGV will come across another check point and the counter will increment again bringing its value to 2. At this stage the AGV knows that it is on its way to B and it will follow the line using both sensors (R1 and L1).

At each destination point of the line (A, B, and C), there is a T shape that the AGV uses to know that it has reached its destination. When the AGV reads the T at any of these points (A, B, and C), it will rotate to the left and reposition itself to the line. This happens every time when the check point count reaches a value of 3 and 4 in case 1, 2, and 4, for case 3 it does it for the values of 4 and 5. Every time that the AGV arrives at its destination and finishes to reposition itself on the line, it stops all motions, sends a message to the master, and wait for further commands to execute. Figure 11 shows the flowchart of Case 1, but the same principle and codes are applied to all the other cases in the AGV control program.
4. Results
This section focuses on the test that will determine whether the AGV operates correctly. The author also describes and analyses the results obtained are each test.

4.1. Test
The tests are setup to evaluate the main functions of the worker (aka the AGV). The worker should effectively follow the line from the storeroom to the masters (aka the assembly line) and back. It should receive commands over the network from the masters and execute them accordingly.

4.1.1. Line following test setup
The AGV, in this test, is placed in an environment where it has to follow a black line on a white surface. The purpose of this test is to check the line following code that is loaded on the AGV's motherboard to see if it will be able to correctly see and follow the line. The test is conducted as followed:
- Place the AGV on the line with the sensors on either side of the line.
- Turn on the switch. Then wait about 10 to 30 seconds for the program in the memory to initiate.
- Turn on the motors switch to start running the AGV

Figure 12 shows the worker's side by side view. It serves as reference for the location of the switches that control the LabVIEW robotic starter kit.

This test will also determine the optimum speed at which the AGV can properly follow the line without going off track.

4.1.2. Communication and network command
This test is designed to check the communication between the AGV and the computer. A simple application runs on the PC to send commands to the AGV to tell it where to go. With each command received, the AGV performs a specific section of its control code. This allows the author to not only test the network communication but also the AGV's destinations repertory.

4.2. Results
4.2.1. Line following

The result of the line following test was as expected, the AGV followed the line successfully. The accuracy of the line following however depends on the speed at which the AGV is traveling. Referring to the calculations below, the maximum velocity of the AGV was determined.

\[
\text{max } DC \text{ motor velocity} = 152 \text{ rpm} \\
\omega = \frac{152}{60} \times 2\pi = 15.92 \text{ rad/s} \\
r = 5 \text{ cm} \\
v = \omega \cdot r \\
v = 15.92 \times 5 \times 10^{-2} = 0.79 \text{ m/s}
\]
where:
\( r \) is the radius of the wheel
\( \omega \) is the maximum angular velocity the AGV
\( v \) is the maximum linear velocity of the AGV

The AGV control program is set in such a way that one uses the linear speed to go forward or backward depending on the sign of the number set to that variable (+ forward and – backward). The angular velocity is used to control the direction in which the AGV turns (+ right and – left). Figure 13 shows that the line following accuracy decreases when the AGV speed increases. It also indicates that the best speed setting to an accurate line following range between 0.1 to 0.38.

4.2.2. Communication and network command

The AGV and PC wireless communication work successfully, the sample program application running on the PC successfully sent messages or commands to the AGV. Figure 14 is the user interface of the application designed to test the communication and commands from the PC to the AGV. Each button represents a command that is sent to the AGV when it is clicked.

The commands from the user interface above are as followed:

- **M1**: Go from A to B
- **M2**: Go from A to C
- **RM1**: Return to A from B
- **RM2**: Return to A from C
- **Stop**: stop all motion
- **DRIVE**: Drive forward ignoring all sensors inputs
- **BACK**: Drive backward ignoring all sensors inputs
- **LEFT**: Turn or rotate Left (all sensors ignored)
- **RIGHT**: Turn or rotate Right (all sensors ignored)

All of the above commands were successfully received and executed by the AGV.

5. Summary and Conclusion

In this project, the author developed and implemented an Automated Guided Vehicle that uses digital infrared sensors to follow a black line on a white surface. The author started with an introduction to illustrate what AGVs are and how they are often used in manufacturing plants. Afterward, the selected hardware and software were reviewed. The methods followed to assemble and program the AGV were shown and explained. Finally, the author documented the tests and results taken to prove the concepts illustrated throughout this project.

This project’s AGV only uses 4 infrared line following sensors to follow 2 different paths depending on the commands from the master application. The principles used in the programming method of the AGV could be applied to other types of line follower robots. The number of paths followed could easily be increased without the need to increase the number of sensors used. The number of sensors could be reduced to 3 if all the check points or reference point are placed on the same side of the followed line. The author would like to emphasize the fact that the line following guidance system developed in this project is reliable, simple to implement and cost effective when focused on the price of sensors needed for an AGV to be able to follow multiple paths. However, as shown in the results, the AGV speed will greatly be reduced for it to follow the line effectively. This line following method is recommended for applications where the speed at which the AGV follows the line is not crucial. For greater speed, it is recommended to use an array of at least 8 digital line following sensors or a proportional–integral–derivative (PID controller) method to program your AGV.
6. REFERENCES
[1] J. Niemann, "Development of a reconfigurable assembly system with enhanced control capabilities and virtual commissioning," Bloemfontein: Central University of Technology, Free State, 2013.
[2] P. Rao, Cad/Cam: Principles & Application: the McGraw-Hill Companies, 2006.
[3] S. Automation. (2014, 29 June). AGV Basics. Available: http://www.agvsystems.com/agvs-basics/basics-agvs/
[4] A. Alavudeen and N. Venkateshwaran, Computer integrated manufacturing: PHI Learning Pvt. Ltd., 2008.
[5] V. Aggarwal. (2013, 13 october). maxEmbedded a guide to robotics, embedded electronics and computer vision. Available: http://maxembedded.com/2013/08/how-to-build-an-ir-sensor/
[6] N. Instruments. (2012, 19 june). NI LabVIEW Robotics Starter Kit Robotics Platform for Teaching, Research, and Prototyping. Available: http://sine.ni.com/ds/app/doc/p/id/ds-217/lang/en
[7] N. Instruments. (2015, 19 May). Overview of the LabVIEW Robotics Module. Available: http://www.ni.com/white-paper/11564/en/
[8] N. Instruments. (2013, 19 May). LabVIEW Delivers Embedded Programming to New NI Single-Board RIO Platform. Available: http://www.ni.com/newsletter/50452/en/
[9] A. Hejlsberg, S. Wiltamuth, and P. Golde, C# language specification: Addison-Wesley Longman Publishing Co., Inc., 2003.
[10] N. Instruments. (2013, 19 JUNE). What is LabVIEW? Available: http://www.ni.com/newsletter/51141/en/
[11] J. Travis. (19 june). Introduction to Graphical Programming with LabVIEW. Available: http://www.informit.com/articles/article.aspx?p=662895&seqNum=3
[12] N. Instruments. (2014, September 2015). NI sbRIO-9632/9632XT. Available: Available: http://zone.ni.com/reference/en-XX/help/373197D-01/sbriohelp/ni_9632/.
[13] J. Niemann, "Development of a reconfigurable assembly system with enhanced control capabilities and virtual commissioning," Bloemfontein: Central University of Technology, Free State, 2013.
[14] P. Rao, Cad/Cam: Principles & Application: the McGraw-Hill Companies, 2006.
[15] S. Automation. (2014, 29 June). AGV Basics. Available: http://www.agvsystems.com/agvs-basics/basics-agvs/
[16] A. Alavudeen and N. Venkateshwaran, Computer integrated manufacturing: PHI Learning Pvt. Ltd., 2008.
[17] V. Aggarwal. (2013, 13 october). maxEmbedded a guide to robotics, embedded electronics and computer vision. Available: http://maxembedded.com/2013/08/how-to-build-an-ir-sensor/
[18] N. Instruments. (2012, 19 june). NI LabVIEW Robotics Starter Kit Robotics Platform for Teaching, Research, and Prototyping. Available: http://sine.ni.com/ds/app/doc/p/id/ds-217/lang/en
[19] N. Instruments. (2015, 19 May). Overview of the LabVIEW Robotics Module. Available: http://www.ni.com/white-paper/11564/en/
[20] N. Instruments. (2013, 19 May). LabVIEW Delivers Embedded Programming to New NI Single-Board RIO Platform. Available: http://www.ni.com/newsletter/50452/en/
[21] A. Hejlsberg, S. Wiltamuth, and P. Golde, C# language specification: Addison-Wesley Longman Publishing Co., Inc., 2003.
[22] N. Instruments. (2013, 19 JUNE). What is LabVIEW? Available: http://www.ni.com/newsletter/51141/en/
[23] J. Travis. (19 june). Introduction to Graphical Programming with LabVIEW. Available: http://www.informit.com/articles/article.aspx?p=662895&seqNum=3
[24] N. Instruments. (2014, September 2015). NI sbRIO-9632/9632XT. Available: Available: http://zone.ni.com/reference/en-XX/help/373197D-01/sbriohelp/ni_9632/.