UoA Undergraduate Mechatronics Research Journal

Volume 4 • December 2011

A Publication of Mechatronics Engineering Group,
Department of Mechanical Engineering,
The University of Auckland, New Zealand

ISSN 1171-9494

UoA Undergraduate Mechatronics Research Journal

UoA Undergraduate Mechatronics Research Journal is published annually by the Mechatronics Engineering Group of the Mechanical Engineering Department, The University of Auckland, 20, Symonds Street, Auckland, New Zealand.

Preface

Welcome to the Undergraduate Research Journal of the Mechatronics Engineering Group at the University of Auckland's Department of Mechanical Engineering. The journal offers undergraduates the chance to present a summary of their achievements and is published in December each year. This fourth volume has only 2 articles selected from the B. Eng (Hons) Mechatronics Engineering final year projects in 2011. This year's contribution from the students is a little disappointing with very few manuscripts received. We hope to receive more contributions in the future.

We hope you enjoy reading about the accomplishments of our students. This journal also provides an overview of the scope and capability of the Mechatronics Engineering at the University of Auckland. We like to thank those involved in supporting the publication of this journal.

Kean C. Aw
Editor-in-Chief

Dec 2011

Table of Contents

Autonomous Guided Vehicle Capable of Laser-Guided Obstacle Avoidance 1
M. Chidambaram and K. Z. Yong

Requirements for Stable Path Tracking in a GPS Guided AGV 7
M.L. Cheuk, L.H.T. Nguyen

Autonomous Guided Vehicle Capable of Laser-Guided Obstacle Avoidance

M. Chidambaram and K. Z. Yong
The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract

This project further developed an autonomous guided vehicle (AGV) focusing on the obstacle avoidance. Obstacle avoidance and motion control algorithms have been developed and implemented on an AGV. This AGV has been equipped with a laser range finder as the primary sensor for obstacle avoidance. The algorithms use the data gathered about the environment to identify and move towards an optimum free path. The obstacle avoidance capabilities of the system were tested in several different path setups, each focused on different configurations; to evaluate the smoothness of the trajectory, turning capability and multiple object scenarios.

The system performed well in all cases with the exception of scenarios that contained multiple objects within close proximity and is believed due to slow processing rate and limited sensing field. In conclusion, it can be seen that the implemented algorithms provide the autonomous guided vehicle with smooth, controlled obstacle avoidance capabilities in most situations.

Keywords: AGV, obstacle avoidance, laser range finder, motion control

1. Introduction

The Autonomous Guided Vehicle (AGV) is a system which can navigate an environment independent of a driver. They are commonly used in several applications, such as warehouses, factories, etc since they can improve efficiency, reduce costs and replace human resources in more dangerous or difficult situations.

To achieve autonomy, the AGV must be able to sense the surrounding environment, process the information it has gathered into meaningful data, which can be acted upon and actuate the system accordingly. One of the important aspects that must be addressed by this control system is the avoidance of objects on the desired path trajectory. The development of a low-cost, efficient and robust solution to this problem is the main objective of this project.

The main task of accomplishing this objective was the development of obstacle avoidance and motion control algorithm. The Obstacle Avoidance Algorithm (OAA) utilizes the environmental data acquired by a Laser Range Finder (LRF) to determine the optimum path of travel. The Motion Control Algorithm (MCA) guides the AGV towards the desired pathway in a smooth and controlled manner.

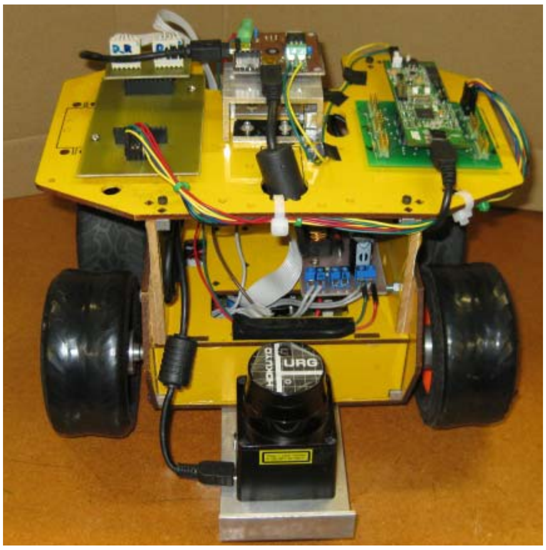

Figure 1. AGV system

2. AGV System Hardware

The current AGV system as shown in Figure 1 consists of the following three main sections. First, to sense the surrounding the AGV has been equipped with a LRF [1]. Second, to move around the environment, each rear wheels is powered by a DC motor. Last, an AVR AT90usb1287 [2] microcontroller has been integrated with the sensors and actuators to provide control. Also several printed circuit boards have been designed to interface with different components and provide a means of power distribution throughout the AGV.

AGV System Software

The software that controls the AGV system in real-time consists of three components is shown in Figure 2. The red section considers the interface between the LRF and the microcontroller via USB communication. The orange section includes the process necessary for obstacle avoidance and the identification of non-obstructed paths. The green section involves moving the AGV to the desired path in a smooth and controlled fashion.

USB Interface

The LRF is interfaced with the on-board microcontroller via an USB. This communication has been set up by a software driver specially modified for this application. For successful operation, the driver must accomplish two tasks. Firstly, the peripheral (LRF) must be identified and a connection must be enumerated. Secondly, the host (microcontroller) must send the correct command then retrieve and decode the returned data.

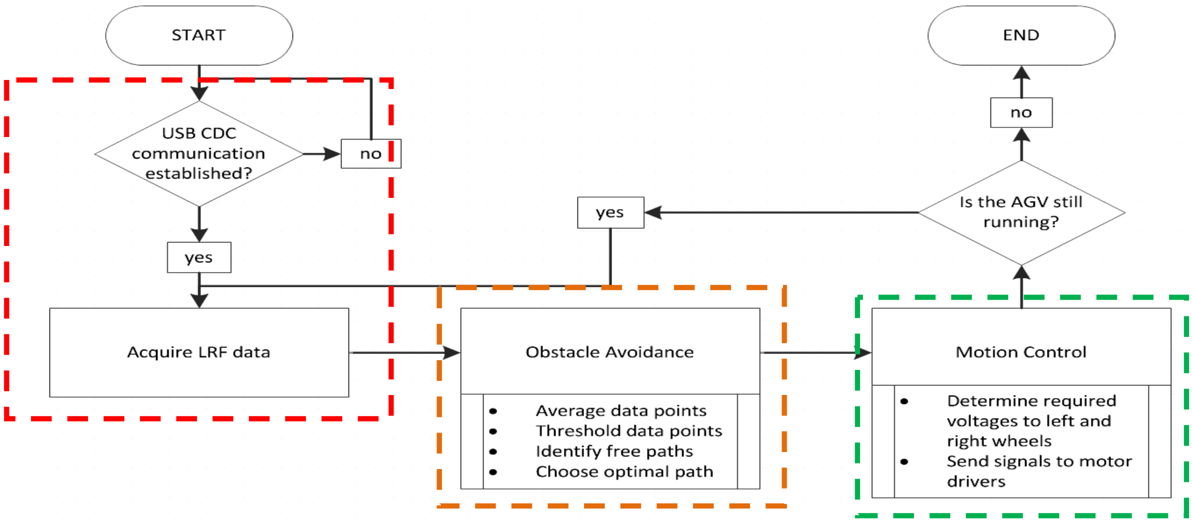

Figure 2. An overview of the software

Obstacle Avoidance Algorithm

Analysing the system as a whole, the OAA can be treated as a black box. The input data is a 2-D scan of data points produced by the LRF. Each data point contains a distance and an angle. The OAA uses this to provide an optimum path angle. This output is sent to the motion control algorithm and will attempt to steer the AGV towards this path.

Within the black box the OAA is similar to VFH [3] and VPH [4]. Polar histograms are utilised, this is advantageous for two reasons. Firstly, the 2-D scan of information produced by the LRF can be directly translated into a polar histogram of distance (mm) versus angle (degrees). Secondly, this form of polar histogram is a very useful representation of the surroundings and can be easily processed to find free paths.

The OAA will execute in order the following tasks every time a new set of data is acquired.

Data Acquisition and Noise Reduction

The LRF range data is collected in groups of thirteen points, which corresponds to 5 degrees. Each group is averaged and these values are stored in an array. This method of data acquisition minimises the amount of memory required. By averaging the data, the effects of noise and amount of data which requires processing are reduced.

Level Threshold

From the previous stage, 38 average distance values will be produced. A threshold is then introduced. If the distance of a point is above the threshold, the value is replaced by a “true”. Otherwise it is replaced by a “false”. A “true” sector represents 5 degrees of free space and vice versa for “false” values.

Available Path Identification

At this stage, the sectors that contain free space have been identified. By grouping neighbouring “true” sectors, larger areas of free space can be found. The number of “true” sectors required to be considered a free path, depends on the “Required Gap Size”.

Path Decision

Since the algorithm has been designed purely for obstacle avoidance, the objective function chooses the path of least change to maximise efficiency. For future applications, the objective function can easily be altered for goal oriented applications by instead choosing the path closest to the goal heading.

Motion control Algorithm

The MCA decides how the vehicle will achieve the desired path. This section of the software controls the actuators required for movement. Ideally, the MCA will achieve the desired path as quickly as possible in a smooth controlled fashion.

The MCA takes advantage of the differential drive system employed by the AGV. By altering the voltage levels to the two DC motors controlling the speed of the two rear wheels, the forwards and turning speeds can be varied. The voltage level sent to each wheel is dependent on the difference between the AGV’s current heading and desired heading as shown in figure 3.

From figure 3 it can be seen that there are two piecewise regions. This has been done to produce an intuitive result.

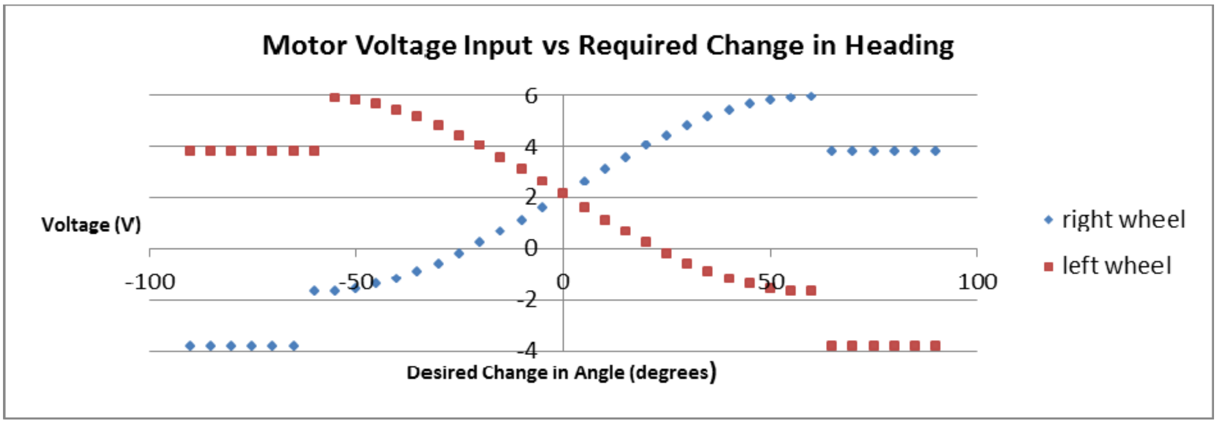

Figure 3. Motion control plot

Upper and Lower Regions

These two regions are entered when a large change in heading is desired. They are defined as the angles above a threshold value and below the negative threshold value respectively. This threshold value is defined by the user. These regions provide a constant voltage to each wheel which is defined by the user.

Central Region

This region controls the motor voltage when small to medium changes in heading are required. It is defined as the angles between the threshold and negative threshold. This sis the same threshold used to define the upper and lower limits. The voltage in this region is governed by an offset sinusoidal function of the desired change in angle. The offset is required to create a positive voltage. The user sets both the amplitude and offset values.

For example, if the AGVs desired trajectory is 25 degrees to the left, which corresponds to a +25 degree change, then the following would occur. The PWM outputs would be adjusted so that the right wheel receives 4V, and the left wheel receives 0V. Since the right wheel has been driven more than the left, the AGV will turn towards the left while still moving forwards. This method is predicted to perform well for several reasons:

- The central region contains gradual changes, which will create a smooth movement.
- The central region always has an average voltage of approximately 2V. This means the AGV should theoretically always be moving forwards for fast travel.
- The AGV will pivot to avoid dead ends or relatively close objects.

Control Interface

For convenience and generalisation, several controlling parameters have been established which can alter the characteristics of the obstacle avoidance and motion systems. By changing the parameters, these algorithms can be adapted to different situations such as varying robot size, narrow corridors, time constrained etc. The description of each parameter is as follows:

Upper Distance Threshold

The value of this parameter represents how far from the AGV, in millimetres, the obstacle must be before action is taken to avoid it. The default value is 650mm.

Lower Distance Threshold
The value of this parameter represents how close an object must be, in millimetres, for the AGV to automatically pivot. The default value is 150mm.

Required Gap Size
This parameter represents how wide a gap should be to be considered as a free path. Every increment represents a 5 degree sector. The default value is 9, which represents an angle of 45 degrees.

Max Variable Speed
This parameter changes the range of available speeds. The “Max Variable Speed” and “Central Speed Offset” can have a total value between 0 and 255, which corresponds a possible voltage range of between 0V and 10.8V. The default value is 110.

Central Speed Offset
This parameter creates a speed offset over the central region as in figure 3. This is required since the sinusoidal function has a mean of zero. The “Max Variable Speed” and “Central Speed Offset” can have a total value between 0 and 255 which corresponds a possible voltage range of between 0V and 10.8V. The default value is 70.

Wheel Speed Threshold Angle
This parameter alters the turning capability of the AGV. It can have a value between 0 and 90 which corresponds to the range of angles where the AGV will attempt to travel smoothly. For example, with a “Wheel Speed Threshold Angle” of 35 the AGV will attempt to travel smoothly for changes in heading between -35 degrees and 35 degrees. The default value is 60 degrees.

Maximum Change in Heading
This parameter limits the maximum desired change in heading to prevent oscillations. The default value is 75 degrees.

Maximum Change in Speed
This parameter limits the maximum change in motor speed. This prevents large reverse currents in the motors but limits the acceleration and deceleration. This value has a range between 0 and 255. The default value is 70.

3. Results

The completed AGV system was tested under several scenarios as listed below, which test the robustness of the obstacle avoidance algorithm.

S-Curve path
The path shown in figure 4 tests the smoothness of trajectory and bi-directional obstacle avoidance abilities of the system. The system was able to cope with this path successfully, traversing without collision in all trial runs.

Dead End scenario

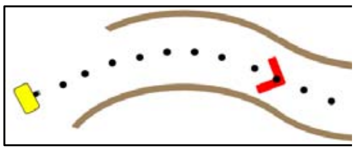

Figure 4 S-Curvec

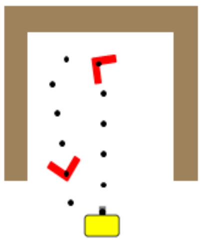

Figure 5 Dead End scenario

The setup shown in figure 5 tests the system’s ability to cope with large turns and backtrack. The system was able to cope with this scenario exiting the dead end without collision 90% of the time.

Cluttered Environment
The setup shown in figure 6 tests the system within a generic environment containing several obstacles within close proximity. The system had some trouble with this environment and only successfully navigated the environment 60% of the time.

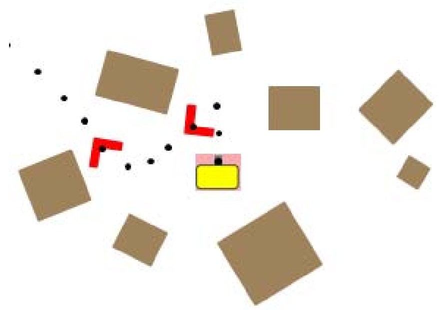

Figure 6 Cluttered Environment

4. Discussion

From the results gathered it can be concluded that the system can successfully avoid obstacles in a smooth and controlled manner in most situations. However, when dealing with multiple obstacles within close proximity the system does have a chance of colliding. This is likely due to the low refresh rate and limited scan angle of the LRF. Both of these characteristics introduce the possibility of the system been within collision range of an undetected obstacle.
However, this is a limitation introduced by the limited computing power and resources involved with low-cost, embedded systems.
Other limitations are introduced when utilising a LRF as the only main range sensor. This involves a reduced reliability in detecting obstacles which are black, transparent, or reflective.

5. Conclusions

The obstacle avoidance and motion control algorithms which have been developed provide the required functionality to a sufficient standard in the majority of applications. However, due to the embedded low-cost nature of the project, there is a limitation when multiple objects are encountered within close proximity of each other. These problems can be overcome with a more powerful processor and multiple range sensors.
The obstacle avoidance and motion control algorithms are ready to be incorporated with goal seeking and localisation methods to provide a complete AGV system.

Acknowledgements

We would like to thank the lab technician L.J. Stuart for his technical expertise and assistance.

References

1. Hokuyo Automatic Company, Ltd. - Laser distance sensor for Robots. Retrieved April 20, 2011 from
2. http://www.hokuyo-aut.jp/02sensor/07scanner/download/data/URG_SCIP20.pdf
3. Atmel Corporation - Atmel AVR 8- and 32-bit. Retrieved June 13, 2011 from http://www.atmel.com/dyn/products/product_docs.asp?category_id=163&family_id=607&subfamily_id=760&part_id=3875
4. Borenstein, J. and Koren, Y. The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots. IEEE Journal of Robotics and Automation, 7 (1991), 278-288.
5. An, D. and Wang, H. VPH: A New Laser Radar Based Obstacle Avoidance Method for Intelligent Mobile Robots. 5th World Congress on Intelligent Control and Automation, June 2004, 15-19.

Requirements for Stable Path Tracking in a GPS Guided AGV

M.L. Cheuk, L.H.T. Nguyen
The University of Auckland,
Department of Mechanical Engineering,
Mechatronics Engineering,
New Zealand

Abstract
One of the challenges in autonomous ground vehicles (AGV) is to autonomously follow a path either defined manually or generated by a path planner. Previous work at the University of Auckland has shown this to be especially difficult for an AGV guided by global positioning system (GPS) due to large amounts errors in the position reading. From the hardware and software changes made to the vehicle this year and the past few years, we have determined that having good localization is vital in stable path tracking. This is achieved by choosing the correct sensors and correctly tuning Kalman filter. In doing this, the results obtained this year show an average of 75% increase in path tracking accuracy.

Keywords: path tracking, AGV, follow the carrot, GPS

1. Introduction

Path tracking in the context of autonomous ground vehicles (AGV) is the ability for the vehicle to drive itself to follow a given path. With AGVs used commonly in industry, this is quite straightforward as the vehicle just needs to follow tape laid out on the ground [1]. However with a GPS guided AGV, this is much more difficult. First of all, the environment is not modified and controlled to direct the movement of the AGV like the former. Secondly, the positioning of the vehicle using GPS is far from accurate. This complicates the AGV's task of deciding on the future steering commands to keep itself on the path. For many years, research in path tracking has been carried out at the University of Auckland for a GPS guided AGV. Up till this point, the path tracking performance has not been satisfactory mainly due to oscillation occurring when the vehicle is attempting to track the path. Building upon past and current research into path tracking algorithms for GPS guided AGVs at the University of Auckland, the basic requirements for good path tracking performance were determined this year.

2. Platform Overview

The UoA AGV is a steered vehicle driven by motors mounted at the rear and coupled with a differential gear box. The motor torque can be varied by varying the PWM signal which it is driven by. It contains an Altera DE2 development board configured with a NIOS II soft processor which handles all the logic and can be programmed in embedded C. Lastly, on board are numerous sensors which will be discussed in the next section. An illustration of the AGV is shown in Figure 1.

Several changes have been made this year to the software such as implementation of wireless sensor monitoring and hardware such as an addition of an ultrasonic sensor. However for the purpose of this article, the focus will be on the changes which directly impacted the path tracking performance of the vehicle.

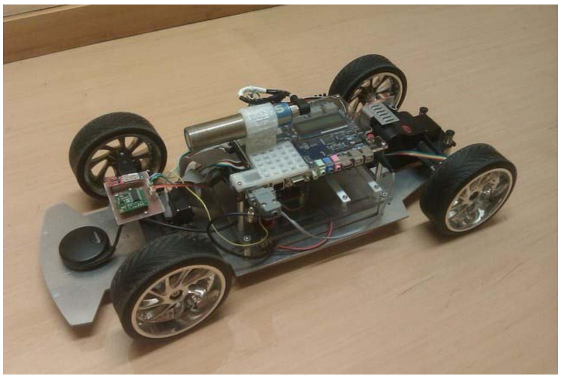

Figure 1: The final AGV platform in 2011

3. Sensor Requirements

Before any path tracking algorithm can be implemented, the vehicle must have sufficient information from the sensors to accurately determine its position.

Several changes have been made to the sensors over the past 5 years, all of which were done to improve path tracking. Before 2010, the sensors in the AGV consisted of a GPS, analogue compass and an inertial measurement unit (IMU). The GPS is the main positioning sensor because it is the only source of the absolute coordinates for the vehicle. However it has an error of approximately of up to 15m which is not suitable for total reliance especially since the AGV is only 0.5m in length [2]. Therefore the other sensors are used to refine the positioning. The process will be discussed later.

In 2010, a wheel encoder was added in lieu of the suggestion in the previous year [2, 3]. The wheel encoders give information about the velocity of the wheels and therefore the position can be estimated from there. This is far more useful than estimating the position using acceleration data from the IMU because errors can accumulate much more easily when double integrating acceleration to give position. Accelerometers are also prone to noise due to vibrations and can suffer from drift if the vehicle is on an inclined surface (since there will be a vector in the plane perpendicular to the direction of gravity) [4]. This proved to be one of the most important changes that contributed to stable path tracking in 2011 as the results will show.

This year in 2011, the compass module which gives the absolute orientation of the vehicle was also improved to be less prone to noise. Analogue compasses can be affected by mechanical vibrations as it uses a free moving part internally. This was solved by using a digital compass which will not be as affected by vibrations as it only measures the magnetic field strength. The results in Figure 2 show much improved noise rejection in the digital compass compared to the analogue.

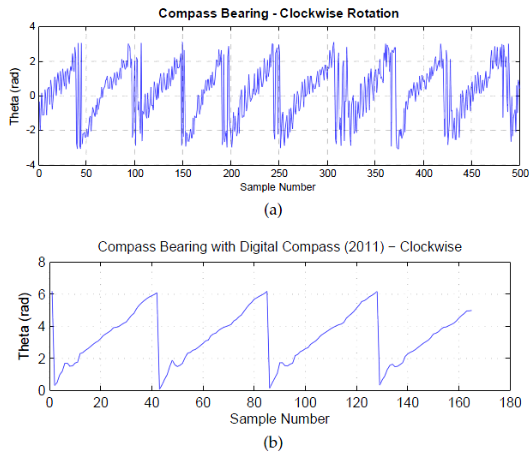

Figure 2: Bearing data obtained from a circular path using (a) Analogue Compass and (b) Digital Compass

The IMU was unused completely this year to simplify the AGV model. Although the more sensors the better, they contributed far less in the estimation of the position so can be disregarded for now and re-implemented later.

A summary of the sensors on the vehicle that were used this year in path tracking are:

- Garmin GPS 18LVC
- Avago AEDB-9140 Series
- Devantech CMPS03

4. Localisation

Localisation is perhaps one of the most important factors for accurate and stable path tracking. In order to determine what the next set of actions should be, the vehicle must know where it currently is. Large fluctuations in position estimates can cause erroneous actions.

As mentioned previously, information from sensors other than the GPS can improve the position estimate. Although inside this particular GPS, noise has been already reduced by time averaging, it can be further improved. The fusion of this information is done using Kalman Filtering. In state space, the minimum of 3 states is required to describe the AGV's position at any time. These are the north and east coordinates and the bearing θ. These are shown in Figure 3 along with other parameters such as steering angle φ, turning radius ρ and velocity v.

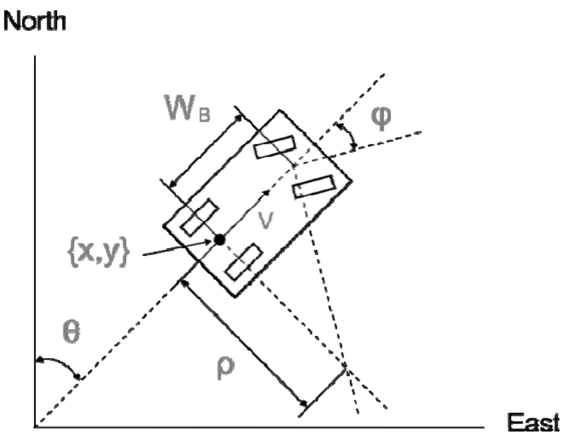

Figure 3: Model of AGV

The translation of the AGV can be described by the following set of discrete time equations.

$$f(x_{k-1}, u_k) = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} T_s v_k \sin\theta + x_{k-1} \\ T_s v_k \cos\theta + y_{k-1} \\ \dfrac{T_s v_k \sin\varphi}{W_B} + \theta_{k-1} \end{bmatrix}$$

And the inputs to the system are given as:

$$u_k = \begin{bmatrix} v \\ \varphi \end{bmatrix}$$

Where T_s is the sampling time.

This kinematic model can then be applied in a Kalman filter. Because there are sine and cosine terms involved, the extended Kalman filter (EKF) for non-linear plants must be used. The EKF predicts a position at time k+1 based on information at time k and the kinematic model. It then corrects this prediction using the position and orientation measurements from the GPS and digital compass respectively.

The advantage of using only three states becomes apparent when tuning the EKF. The matrices which can be varied are Q and R, which are the process noise and measurement noise covariance matrices respectively. Previously, six and eleven state EKFs were used so at least twelve parameters were required to be tuned. However now that there are three, only six parameters need to be tuned. Of course performance can be improved later on when more sensors are added.

It is important that the EKF is tuned from real data collected because tuning the EKF on simulated noise can give misleading results, which was done in some years [ref]. The vehicle was driven in a square path following the markings on a basketball court and the sensor data were recorded. A MATLAB model simulating the EKF was then run on this data and the parameters were varied until the results were satisfying. This is valid because the EKF does not affect the actions of the AGV since it is manually controlled so the simulation will match the actual result very closely if it were run on the vehicle.

5. Path Tracking

The three most common path tracking algorithms are follow the carrot, pure pursuit and vector pursuit [5]. Follow the carrot is the simplest but does have disadvantages such a tendency to cut corners [6]. However currently we are only trying to achieve stable path tracking on simple paths which do not require strict cornering ability hence Follow the Carrot will suffice. [This is supported by evidence that follow the carrot performs just as well in simulation as vector pursuit on simple linear paths]. Furthermore, if path tracking can be performed well using this simple algorithm, then it can be further improved with more complex algorithms such as pure pursuit or vector pursuit.

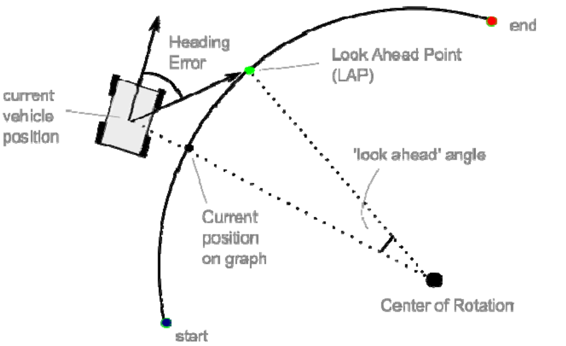

Figure 4: A vehicle tracking a circular path with all the parameters labelled

At each iteration, the algorithm proceeds as follows:

1. Find the look-ahead point
2. Calculate the difference in angle between the vehicle heading and the look-ahead point (heading error)
3. Multiply this difference in angle by a gain K to find the desired steering angle

The look-ahead point is found by determining the closest point the vehicle is at to the path and adding a fixed "look ahead distance". For a straight line, this is just the perpendicular projection of the vehicles position on the line. For a circle, this is the intersection of a line between the centre of rotation and vehicle position with the circumference. The key points calculated in the algorithm are depicted in Figure 4.

6. Results

The path tracking ability using the above configuration was tested by defining paths via a start and end point. For arcs, an additional parameter defining the radius was used. The tests were carried out on flat open area. In all tests, the look ahead distance LAD = 1m and gain K=2.

Rectangle

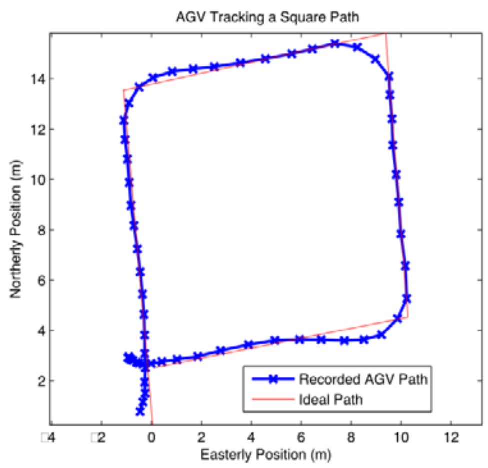

Figure 5: The recorded positions of the AGV while tracking a rectangular path

A rectangular path is defined by 4 linear paths joined together. Figure 5 shows the EKF estimated positions which were recorded during the trial, along with the ideal path for comparison. As shown, the shape of the path is followed very closely, with the exception of some cutting of corners and overshoot of about 0.5m.

Triangle

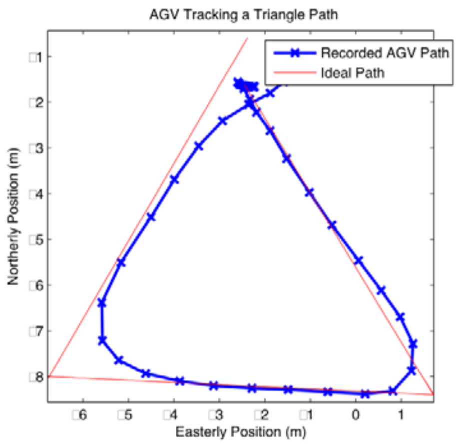

Figure 6: The recorded positions of the AGV while tracking a triangular path

Similar to the rectangular case, a triangle can be defined by 3 straight lines. Again the results shown in Figure 6 are satisfactory. The only issue is the more apparent corner cutting because of the smaller angle between the lines. One thing to note is that although the vehicle did not start exactly at the define start point, the vehicle is able to quickly steer itself back to the defined path.

Arc

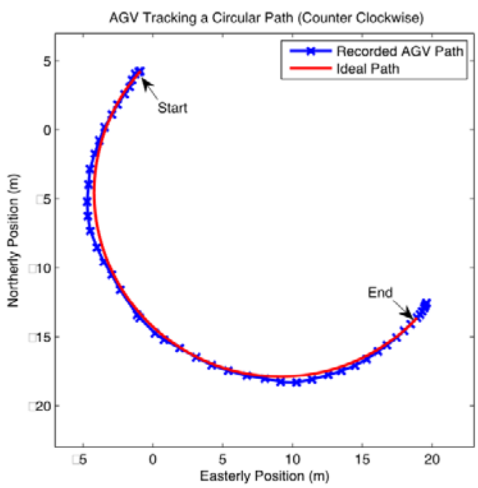

Figure 7: The recorded positions of the AGV while tracking an arc

Circular paths were tested by defining two points, radius and a direction. Figure 7 shows the result for a semi-circle in the CCW direction. This is done very well. The only problem is that it stops slightly after the desired end point. This could be attributed to the vehicle moving due to inertia as there is no braking action when path tracking has complete. The maximum of deviation observed is about 0.5m.

7. Discussion

The simplest path tracking algorithm, follow the carrot, was used in all of the above tests. The results show that it performed better than many of the more advanced and theoretically better algorithms such as pure pursuit and vector pursuit implemented previously such as the result in Figure 8. The deviations in tracking linear paths reduced from about 2.5m to about 0.5m hence showing a 75 per cent improvement in error reduction (without taking into account corner cases). Most importantly, the oscillations experienced before have been eliminated. Although circular tracking was not carried out in the past so a comparison cannot be made, the amount of deviation is similar to that of linear cases. This means that most of the poor performance experienced in the past years was not due to the choice of path tracking algorithm, but the localisation and sensor choice. Therefore if a more advanced path tracking algorithm was implemented later, then the path tracking performance for complex paths such as circular arcs can be expected to improve.

Wheel encoders coupled with a compass are able to estimate the position of the vehicle rather accurately for short distances without any source of the absolute position (such as a GPS). Again, prior to 2010, there was no wheel encoder. In addition, the accelerometer was very noisy that it has been unused for many years and so less information is there to fuse [ref some years].

In path tracking, it is also important to have a correctly modelled and tuned EKF. A lot of effort was placed in ensuring that the EKF produced an acceptable position estimate. By ensuring the EKF produced a satisfactory result, the true result will be closer to simulation (which assumes perfect localisation). With the combination of the choice of sensors and a well implemented EKF, stable path tracking can be achieved as demonstrated by the results.

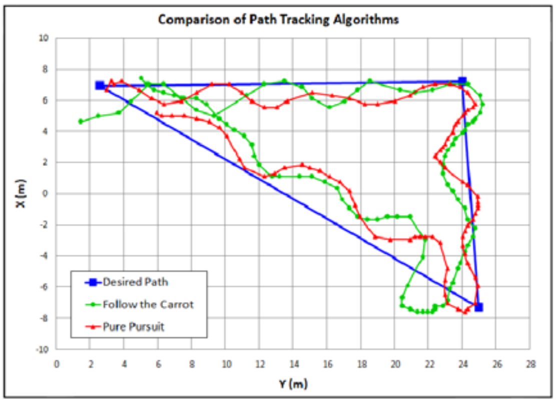

Figure 8: Results from Follow the Carrot and Pure Pursuit in 2008

8. Conclusion

The most important aspect of obtaining good path tracking is good localisation. Good localisation can come from the choice of sensors, and the algorithm which fuses the information from these sensors to give a single position estimate. In this case, the two most important sensors required to complement the GPS guided AGV for the purpose of path tracking is a digital compass and wheel encoder.

Once good localisation is achieved, the choice of the path tracking algorithm will only depend on the geometry of the paths that will be tracked. The results have shown that for simple linear and circular paths without strict corner following requirements, follow the carrot will suffice.

The final implementation based on these concepts has shown an average of 75% reduction in error from the path which is a significant improvement over past results at the University of Auckland.

Acknowledgements

We would like to thank Mr Logan Stuart for his help in obtaining hardware components. We also would like to extend our thanks to Dr Karl Stol for his help in understanding the modelling and tuning process for the EKF.

References

1. JBTCorporation, "Pallet Handling Automatic Guided Vehicles (AGVs)," ed, 2009.
2. J. P. Jackson, "Localisation and Path Tracking in a Steered-Wheel Autonomous Ground Vehicle," University of Auckland2010.
3. C. C. Wei, "Improvement to Sensor Accuracy and Path Tracking of Autonomous Ground Vehicle," University of Auckland2009.
4. J. Z. S. Wong, "GPS Guided Autonomous Ground Vehicle with Improved Localisation and Path-Tracking," University of Auckland2007.
5. T. Hellstrom, "Autonomous Navigation for Forest Machines," Umea University, 2002.
6. J. S. Wit, "Vector Pursuit Path Tracking for Autonomous Ground Vehicles," University of Florida2000.