I, Suresh L. Paul, hereby submit this as part of the requirements for the degree of:

Master of Science in:

Industrial Engineering

It is entitled:

Curvilinear traverse generation module for an AGV

Approved by:

Dr. Ernest L. Hall
Dr. Richard L. Shell
Dr. Ronald L. Huston
Curvilinear traverse generation module for an AGV

A thesis submitted to the Division of Graduate Studies and Advanced Research of the University of Cincinnati in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in the Department of Mechanical, Industrial and Nuclear Engineering of the College of Engineering

2003

By

Suresh L. Paul

B.E. (Mechanical Engineering)

Anna University, India, 2000

Thesis Advisor and Committee Chair: Dr. Ernest L. Hall
Abstract

The purpose of this thesis is to study the point-to-point navigation of an AGV in an unstructured environment. The problem under consideration is to compute the minimum distance of traverse of the AGV through a set of known points, where the routing through these points would determine the efficacy of the AGV to navigate the domain. The criticality of this study is to enhance the response of the AGV in exploring a specific domain. This response time is vital for minimizing power and fuel requirement of the AGV.

The response of the AGV to any change in a straight-line direction is associated with the close down – resume characteristics. A close down-resume characteristic is a kind of straight-line operation of a vehicle when it has to stop in order to change the direction of its course. Hence, the number of target points in the domain would dictate the number of times that the AGV must stop and change its course. This routing is compared with a curvilinear path, which is formulated by generating a curve through the same target points. Since the curve has less change in directional properties compared to a straight-line, the AGV can respond without a close down-resume characteristics and perform a uniform velocity traverse, thereby, reducing the time for traverse through the domain.

This study focuses on providing different methods of generating curves in a domain with a given set of target control points. The methods formulated in this study
are curve-fitting using indexing reference points, which is a geometrical construction of a curvilinear path, Voronoi curve generation method, which uses the Voronoi principle for curve generation, and a hybrid curve generation method, which is a combination of the above two methods.

The results of this study show the time required to travel in any area, given the optimum details of the terrain. There is a definite reduction of the time of travel by the AGV in comparison with the other navigation modules. It also proves to be safer to account for the maximum travel time for the AGV to explore, as its operation is restricted to fuel/power consumption. Further, the AGV can stop at any point on its travel and negotiate a reverse traverse back to the start point, if necessary.

The significance of this study is to understand and interpret the higher flexibility of the vehicle to explore the domain by sending feedback on its surroundings in a curvilinear path. The constraint for navigation is avoiding collision with any significant obstacle in its track. The AGV is pre-functioned with the obstacle avoidance operational mode, and must be able to give the exact time required for the travel to be safe as required for its functionality.
Acknowledgements

The experience with the Center of Robotics Research and being involved with a high-end technical group has certainly given me a moral boost in my career objectives. With my sincere gratitude, I would like to put forth the following acknowledgements.

My advisor, Dr. Ernest L. Hall, has been very supportive and guided me in all my research areas. His motivation has led me to perceive and propose this thesis idea. His suggestions, criticism and feedback greatly refined my work.

I would like to thank Dr. Richard L. Shell and Dr. Ronald L. Huston for agreeing to serve on the defense committee. I would also like to thank them for comments and feedbacks.

I would like to thank the Robotics Team members for being supporting and encouraging in all perspectives. I would like to thank all my friends and well-wishers who had helped me from time to time.

Finally, I would like to thank Herman and my family for everything. I owe all my success to them.
# Table of Contents

**CHAPTER 1: INTRODUCTION**

1.2 To find the minimum distance of traverse in a domain  - 14 -
1.3 To find the minimum distance between a set of three successive points  - 14 -
1.4 To compute the reference index point  - 15 -
1.5 To compute the nearest point on the circle defining the roaming distance and the reference index point  - 16 -
1.6 Analysis of Time through curve generated path through control points  - 18 -
   1.6.1 Straight line Routing  - 19 -
   1.6.2 Curve Routing  - 19 -
1.7 Interfering distance (Int)  - 21 -

**CHAPTER 2: LITERATURE REVIEW**

2.1 Sufficiency Data & Stray Data  - 23 -
2.2 Sensors  - 23 -
2.3 Visual Sensing ISCAN device  - 23 -
2.4 SICK Laser Scanner  - 24 -
2.5 Rotating Sonar  - 24 -
2.6 GPS  - 24 -

**CHAPTER 3: CURRENT NAVIGATION MODE OF UC BEARCAT ROBOT III**

3.1 Data Input & State of the Robot  - 26 -
3.2 Principle - Current Navigation Mode of UC Bearcat Robot III  - 26 -
3.3 Constraints - Current Navigation Mode of UC Bearcat Robot III  - 27 -

**CHAPTER 4: PROPOSED NAVIGATION MODE**

4.1 Distance Computations  - 28 -
4.2 GPS Navigation  - 29 -
   4.2.1 Principle of Least distance  - 29 -
4.3 Obstacle Avoidance Mode  - 30 -
   4.3.1 Principle of Voronoi for Obstacle Avoidance  - 30 -
CHAPTER 5: CASE ANALYSIS - 31 -

5.1 Case I: Navigation of an AGV through the matrix of points using the path of traverse from a start point to the end-point and independent of number of obstacles present in the domain - 31 -

5.2 Case II: Navigation of an AGV through the matrix of points using path of traverse from a start point to end point, having a fixed roaming distance “D” from the obstacles present in the domain - 34 -

5.3 Introduction of Hybrid model of curve generation - 37 -

CHAPTER 6: ADVANTAGES AND DISADVANTAGES - 38 -

Advantages - 38 -

Disadvantages - 39 -

CHAPTER 7: ALTERATIONS AND SCOPES FOR FUTURE RESEARCH - 40 -

7.1 Three dimensional curve generation - 40 -

7.2 Regenerating curve with a new initial point - 41 -

7.3 Curvilinear routing using random routing technique - 42 -

7.4 Laser guided curvilinear traverse - 43 -

7.5 Traversing beyond a cliff: Pro-logical routing - 44 -

CHAPTER 8: ALGORITHM FOR THE FORMULATION - 46 -

8.1 Straight line routing: - 46 -

8.2 Curvilinear routing: - 47 -

8.2.1 Curve-fitting using indexing reference points - 47 -

8.2.2 Voronoi curve generation - 48 -

8.2.2 Voronoi curve generation - 49 -

8.2.3 Hybrid curve generation - 50 -

CHAPTER 9: CONCLUSION - 53 -

REFERENCES: - 56 -
Table of Figures

Chapter 1

Fig 1: Planar projection of a three-dimensional domain........................................12
Fig 2: Velocity profile for the AGV.................................................................12
Fig 3: Indexing reference points........................................................................15
Fig 4: Reference point and revised reference point..............................................16
Fig 5: Control Points.........................................................................................17
Fig 6: Velocity profile of curvilinear routing......................................................18
Table 1: Total time of traverse..........................................................................20
Fig 7: Roaming distance and Interfering distance..............................................21

Chapter 4

Fig 8: Principle of least distance.........................................................................29
Fig 9: Voronoi curve generation.........................................................................30

Chapter 5

Fig 10: Voronoi curve generation for Case I....................................................32
Fig 11: Practice Navigational Map.....................................................................34
Fig 12: Curve-fitting using Indexing reference points, Voronoi generated Curves...36
Fig 13: Hybrid Model of Curve Generation.....................................................37

Chapter 7

Fig 14: Three dimensional curve generation.....................................................40
Fig 15: Regenerating curve with a new initial point.........................................41
Fig 16: Curvilinear routing using random routing technique..............................42
Fig 17: Laser Scanner image ............................................................................43
Fig 18: Traversing beyond a cliff – Pro-logical routing....................................44
Chapter Outline

Chapter 1 introduces the existing straight-line navigation methodology and the proposed navigation module, which is explained in detail in Chapter 3 and Chapter 4. Chapter 2 lists the literature review of the hardware involved in setting up this module. The two separate cases in which the proposed navigational methodology can be applied in autonomous navigation mode are explained in Chapter 5. Chapter 6 enlists the advantages and disadvantages of this study. The main advantage of performing this study is creating a high-level intelligent navigation module for the AGV to choose, depending on the surrounding. The main disadvantage of this study is the projection of a three-dimensional terrain on a two-dimensional mapping. Chapter 7 provides the alterations required to overcome the disadvantages mentioned in Chapter 6, and scopes for future study. Chapter 8 gives the algorithms to code this formulation.
Chapter 1: Introduction

The automated guided vehicle (AGV\(^1\)) is an intelligent system capable of interfacing with the surrounding through sensors and is capable of evaluating the environment. This evaluation can be feature recognition of objects, avoiding interference and navigation. The navigation algorithm is important in several applications in several industries, where an automated guided vehicle is used for routing along a desired path dictated by constraints\(^2\). In this particular case, the study illustrates the navigation of an unmanned guided vehicle like the autonomous navigation robots. The criteria of navigation are dictated by certain constraints like route or line following, responses towards interference like obstacle detection and avoidance, speed of traverse and the time of travel. These are essential in the robot to upgrade into Intelligent Systems\(^3\) and diverse as conceptually programmed machines.

The question of perception of data in real world, the type of feedback, decision making and speed, differentiates the capabilities of machines and humans. The navigational challenge involves the robot to respond to the surroundings and provide information, which could help in the navigation of the robot through a framework of obstacles. The required data acquisitions of the robot are location, orientation and position of interfering obstacles. The data pertaining to the location of the robot is acquired by the Global Positioning System (GPS) in real-world coordinates and with those related to the obstacles can be obtained through interactive sensing devices like the Visual Scanner, Laser Scanner, Rotating Sonar and the Global Positioning System. The required data format for computing and the accuracy of the data are also crucial.
To accomplish the task of such navigation, a set of reference points is required in the domain of the traverse to define the path of traverse. These points serve as control points for the path. These control points are, however, defined with two parameters – roaming distance and interfering distance. Roaming distance (d) is defined as the optimal minimum distance from which the AGV can evaluate the control point. The Interfering distance (Int) is defined as minimum distance with which the AGV can avoid colliding with the control point.

There are several ways to formulate a navigational algorithm based on plotting the control points in a domain. Firstly, a two dimensional projection of the control points on the x-y plane is observed. We find that, there is a certain sequence of points, which determines the minimum path in the domain. Further, the AGV need not interfere with the point of interest; instead it can observe the point from a distance defined as the roaming distance. Hence, the control points are then transformed to set guide points, known as reference points in the same domain, through which the AGV can negotiate the path. After determining these reference points, a curve-fitting is done through the set of reference points, to establish a curvilinear path. The curve passes through the start and the end points, and is tangential to all the circles defining the roaming distance around each control points.

In this curvilinear traverse, the change in direction by the AGV during its path is minimal. Thus, a constant velocity at the maximum permissible speed along the curve is possible. In this case, the velocity profile of the AGV has one acceleration period, one constant velocity period and one deceleration period, irrespective of the number of control points in the domain.
1.1 Description of the Navigation Module

Fig 1: Planar projection of a three-dimensional domain
Note that the minimum traversing straight line distance is shown as a doted line in the above figure.

Fig 2: Velocity profile for the AGV
Consider the given situation of a traverse\(^5\) from the start point, P\(_1\), till the end point, P\(_2\), and three control points, I\(_1\), I\(_2\) & I\(_3\). The roaming distance, d, around each control point is put down to investigate the formulation of the reference points. The dotted line shows the minimum distance straight-line routing possible for this set of points. However, the AGV will have an initial acceleration period at the start point, P\(_1\), traverse at constant velocity to I\(_1\) and a deceleration period at I\(_1\), before it proceeds to I\(_2\). Similarly, there is an acceleration period, constant velocity period and deceleration period at each of the control points I\(_2\) and I\(_3\), till it reaches the end point, P\(_2\). From the above velocity profile, we find that there are four acceleration periods\(^6\), four constant velocity periods\(^7\) and four deceleration periods\(^8\).

This level of navigation is the most preliminary way of navigating, which is, traveling in straight-line routing\(^9\). Though, straight-line routing is the easiest formulation of a navigational module, it has several limitations in generating routes in unstructured environments. For most cases, there may not be a straight line between two control points. Further, straight-line navigation is time-consuming and poses constraints on the power capacity of the AGV.

If a curved path represents the points in the domain from the start point to the end-point, having only one acceleration period, one constant velocity period and one deceleration period, then, it will overcome the above-mentioned constraints of the straight-line navigation module. This option would take less time to negotiate the path.
1.2  **To find the minimum distance of traverse in a domain:**

To find the minimum distance along the set of control points, the distance between any three successive points in the domain is evaluated to facilitate the construction of the curve geometrically. The same is repeated for all combinations of successive three points, and the least sum of the distance between points is taken. The path which provides this sum is the path of least distance. For example, the minimum distance of traverse from start point to the end point is through I₁, I₂ and I₃, in the same order. This is obtained by optimizing the distances between P₁, I₁, I₂, I₃ and P₂ with each other.

1.3  **To find the minimum distance between a set of three successive points:**

For every set of three points, each point has a preceding point and a successor point. The intermediate point can be used for the generating a point, known as *relative indexing point*, which is at the least distance from the preceding point with respect to the successor point. The relative index points are outside the circle defining the roaming distance. Hence, these points need to be projected to the roaming distance circle, forming the reference points for curve generation.

From each preceding point and successor point, tangents are drawn to the circle defining the roaming distance of the intermediate point. The tangents which intersect along orientation of the curve are considered, and the points of intersection of these
tangent lines are called the relative indexing point, $i_1$, $i_2$, $i_3$ and $i_4$. Similarly, the next sets of three points are taken, and the relative indexing points are found.

![Indexing reference points](image)

**Fig 3: Indexing reference points**

where, Point 1 – Successor Point  Point 2 – Preceding Point  $i_i$ – Indexing points

$d$ – Roaming distance

The point of intersection of these two lines is the reference point for the change in direction of the AGV. Note that the point of intersection lies outside the roaming distance field of the intermediate point.

1.4 **To compute the reference index point**

The intersection of tangents to the circle defined by the roaming distance yields 4 indexing points – $i_1$, $i_2$, $i_3$ and $i_4$. For the given set of preceding and successor points,
there will be one and only one indexing point which will have the nearest possible
distance from both the preceding and successor points. Evaluating the sum of the distance
of each indexing points with respect to the preceding and successor point, one indexing
point with the least values of distance is found. (Say $i_1$)

1.5 To compute the nearest point on the circle defining the roaming distance and
the reference index point

After the reference point has been found for a set of three points, this point is
projected on the circle defining the roaming distance. The reference point is then indexed
on the circle defining the roaming distance, by the intersecting point of the line joining
the reference point and the center of the circle. The dotted line in the diagram, drawn
from the center of the roaming distance circle to the reference point, intersects the
circumference of the circle at a point. This point of intersection becomes the revised
reference point. In a similar way, the reference points are projected onto their respective
roaming distance circles to form a set of revised reference points. These revised reference
points are only used for curve generation.

*Fig 4: Reference point and revised reference point*
Thus, the indexing point on the circle (revised reference point) gives the shortest sum of distances from the Point 1 and Point 2. The path generation is formulated by sequencing this set of three points (Point 1 – Intermediate – Point 2) from the start point to the end point. The resultant path obtained by generating revised reference points for each intermediate point, will be an optimized straight-line path for the traverse of the AGV. The start point, revised intermediates and the end point form a set of control points required for curve generation.

**Fig 5: Control Points**

Control Points = [Start Point, Ri, End Point]
1.6  Analysis of Time through curve generated path through control points

After the domain reference points are constructed, the curve is generated through the given set of control points in order to obtain the least possible change in direction. There are several algorithms developed for curve generation, and these curves can be simulated in Matlab. Now, there is only one velocity profile for through the entire navigation of the domain. The velocity of the AGV observes a constant velocity of steer through the curve, thereby avoiding local acceleration and deceleration at control points.

Fig 6: Velocity profile of curvilinear routing
1.6.1 Straight line Routing:

For a set of ‘n’ control points (start & end inclusive), we have ‘n-1’ equal acceleration and deceleration periods, and ‘n-1’ variable constant velocity periods.

For, ‘n-1’ acceleration periods; \( \text{Time}_{\text{acceleration}} = (n-1) \times \text{Time for acceleration} \)

‘n-1’ deceleration periods; \( \text{Time}_{\text{deceleration}} = (n-1) \times \text{Time for deceleration} \)

‘n-1’ constant velocity periods; \( \text{Time}_{\text{constant velocity}} = \sum \text{Time for each constant velocity period} \)

Total Time for traverse = \( T_{\text{traverse}} = (n-1) \{ \text{Time for acceleration} + \text{Time for deceleration} \} + \sum \text{Time for each constant velocity period} \)

1.6.2 Curve Routing:

For the same set of ‘n’ control points, there is only one period of acceleration, constant velocity and deceleration periods.

Acceleration period = \( \text{Time}_{\text{acceleration}} = 1 \times \text{Time for acceleration} \)

Deceleration period = \( \text{Time}_{\text{deceleration}} = 1 \times \text{Time for deceleration} \)

Constant velocity period = \( \text{Time}_{\text{constant velocity}} = \text{Time for constant velocity period} \)

Total Time for traverse = \( T_{\text{traverse}} = \{ \text{Time for acceleration} + \text{Time for deceleration} \} + \text{Time for each constant velocity period} \)

Consider, \( \text{Time for acceleration} = A \)

\( \text{Time for deceleration} = D \)

\( \text{Time for constant velocity} = CV \)
**Total Time for traverse** \( T_{\text{traverse}} \)

<table>
<thead>
<tr>
<th>Straight line Routing</th>
<th>Curve Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>((n-1) {A + D} + \sum CV_i)</td>
<td>({A +D} + CV)</td>
</tr>
</tbody>
</table>

**Difference between Time** Straight line Routing and Time Curve Routing:

\[
K = \{(n-1) \{A + D\} + \sum CV_i\} - \{A +D\} + CV
\]

\[
= \{(n-2) \{A + D\}\} + \sum CV_i - CV = K_1 + K_2
\]

where

\[
K_1 = \{(n-2) \{A + D\}\}
\]

\[
K_2 = \sum CV_i - CV
\]

And, \(K_1 \geq 0\); since the number of control points ‘\(n\)’ \(\geq 2\) (start and end points are always specified).

---

**Table 1: Total time of traverse**

The two constants, \(K_1\) and \(K_2\), are found, which are the difference of acceleration periods and deceleration periods and the difference in constant velocity periods. ‘\(K_1\)’ will always be non-negative, as the factor “n-2” is always non-negative, where ‘n’ is the number of control points. This is because, the start and the end point is always specified. The term, ‘\(K_2\)’, maybe positive or negative based on the difference in the constant velocity periods between straight-line routing and curvilinear routing.
1.7 Interfering distance (Int):

Interfering distance (Int) is defined as minimum distance with which the AGV can avoid interfering\textsuperscript{11} with the control point. Instances in which the straight line between the indexing points and the preceding or the successor point, interferes with the circle defined by the roaming distance, there is a maximum distance that the AGV can penetrate the circle without interfering with the intermediate point. The interfering distance is governed by the response of the AGV to stop at the roaming distance and switch on to the subsequent path to the successor point. The distance from which the roaming distance is calculated is from the center of gravity of the AGV to the intermediate point. The collision of the AGV may occur when the center of gravity lies within the roaming distance circle.

\textit{Fig 7: Roaming distance and Interfering distance}
Chapter 2: Literature Review

There are several navigation methods and systems for mobile robots. The main idea in developing these methods of navigation is to create ways for the mobile robots, AGV, to interact with different environments and perform different tasks. One of the major tasks for an AGV is to navigate through a set of points in the least possible time, i.e. in an efficient way. In underlining the evolution of the navigation systems, which formed the basis for any navigation model, there are several systems such as navigation using Odometry and other dead-reckoning methods, Inertial Navigation, Active Beacon Navigation, Land-mark navigation, Map-based positioning and the Global Positioning Systems. The evolution of these navigation methods makes the robot-environment interaction as flexible as possible.

The sensor systems that the AGV uses to navigate in an unstructured environment are Digital CCD camera, I-scan Image processing unit, Laser Scanner and Differential Global Positioning System.
2.1 **Sufficiency Data & Stray Data:**

The data required for processing and computing are defined as Sufficiency Data, and all those data that are out of the scope of the intent are considered as Stray Data. The importance of recognizing the sufficiency data and eliminating the stray data decides the complexity of the application and prompt response of the robot to conditions defining the surroundings.

2.2 **Sensors:**

The sensors used in the UC Bearcat Robot III are Visual Sensing ISCAN device, SICK Laser Scanner, Rotational Sonar and the Motorola & Garmin GPS Units. The sensors are used to provide data in accordance with the certain conditions. The parameters considered for the limitations of the sensors are basically to avoid excessive data which delays the response time of the robot.

2.3 **Visual Sensing ISCAN device:**

Visual Sensing ISCAN device is an optical sensing device, which includes two light cameras on each side of the robot, to observe the line of traverse of the robot. The ISCAN views the image as a two-dimensional image, within a window of sight specified by the program to avoid stray data. The robot is programmed to shift its control between the cameras, under conditions of a void signal returned by either one of the cameras, thereby ensuring that the robot never goes out of track or astray in the specified domain. The visual sensing requires calibration of the cameras to focus on the desired reference
before each traverse, to validate the optimum operation of the ISCAN towards the desired result.

2.4 **SICK Laser Scanner:**

SICK Laser Scanner is a sensor, which operates on sending and receiving laser signal, and thereby identifying the data from the surrounding. It emits a single level laser beam on a 180 degree arc in the front of the robot, and frequency resolution of the beam can be tuned as per required (usually 1 degree). The beam can operate in a range of 80 meters and 30 meters in outdoor and indoor environments respectively and sends back data pertaining to the distance of the object and the width and orientation of the object.

2.5 **Rotating Sonar:**

Rotating Sonar functions on the sonar principle of transmitting and receiving sound waves. The principle of identifying the orientation using sonar is based on splitting the region into desired zones and then identifying the zone from which the data is obtained. The reflectivity of the sound waves causing the weakening of the signal is a vital constraint in using sonar for detection purposes.

2.6 **GPS:**

Motorola and Garmin GPS units are used in unison in the robot to generate differential GPS principle and obtaining the global coordinates which are accurate than using either one of them. The expected accuracy of the GPS systems is within one foot.
The sensors receive data from its source at a rate less than a second. Observing this amount of data, which involves further computation, requires a superior interfacing and processing speeds of the processor. Avoiding the interference between the signals received at one instant and the next, should be appropriately distinguished to avoid error during computation. Thus, we require refreshing\textsuperscript{18} of the buffer in the interface to drain processed data and to separate them from fresh data obtained through the sensors. The tuning of the sensors and the draining of the buffer interface ensures the elimination of stray data in the system.
Chapter 3: Current Navigation Mode of UC Bearcat Robot III

3.1 Data Input & State of the Robot:

- Autonomous Mode
- Health Monitoring\textsuperscript{19} of Batteries
- Global Coordinates
- Visual Sensor Tuning and Calibration
- Sensors Status ON
- Set of points defining GPS testing

3.2 Principle - Current Navigation Mode of UC Bearcat Robot III:

The robot follows the line track guided by the visual sensors along the traverse distance, until it encounters an interfering signal from either of the sensors in the case of an obstacle. The line-tracking mode is then shifted to the obstacle avoidance mode guiding the robot to make a turn in order to avoid the obstacle. The turn is carried out until the obstacle is considered a stray data that no longer lies in the path of the robot. The robot then traverses until the line is captured by either one of the camera and follows that line. This operation is cyclic until the entire distance is covered. In case, the line disappears on one side of the robot, the control would shift to the other side camera to check the availability of the line on the other side, and propagate until the line on the other side is captured.
The robot can traverse through a domain specified by a set of points in global coordinates obtained from the GPS, in which case, the robot can be made to traverse from point to point from the start till the end. The speed of traverse is limited to a specific value, to render the testing and experimenting of the robot under safe limits.

### 3.3 Constraints - Current Navigation Mode of UC Bearcat Robot III

- Since, we cannot establish the route chosen from the start point along the required path, it is uncertain to determine the time of traverse of the robot.
- Point-to-Point navigation is the most basic level of programming an AGV to navigate.
Chapter 4: Proposed Navigation Mode

It is feasible to develop a traverse path for the robot, before the robot actually starts. By defining the path as decisive, we can compute the distance of travel along the path. If the domain is reasonably flat, then the computed distance of travel is approximately equal to the actual distance to be traversed by the robot. Since the speed of the robot is definite, the time for the traverse can be optimally found. The criticality of the time of traverse lies in analyzing whether the robot can perform the navigation through the matrix of points before the power diminishes below the threshold value, or to abort the operation without an attempt.

4.1 Distance Computations

The distance computations for obstacle detection and avoidance, and point-to-point traverse using the GPS involve two separate logical approaches. In an obstacle detection and avoidance mode, the robot need not consider all the obstacles to confront and avoid, instead choose a path that has the minimum number of obstacles. On the other hand, the GPS navigation involves the robot to navigate through a close proximity of each mentioned point, and then formulating the shortest path of traverse. We witness the same situation in the GPS waypoint navigation contest in the International Ground Vehicles Competition (IGVC) organized by the Association of Unmanned Vehicle Systems international (AUVSI).
4.2 GPS Navigation

4.2.1 Principle of Least Distance\textsuperscript{20}:

The principle involves the computation of the various possible distances from one point to another, and then finding the various possible routes from the start point throughout the grid of points that has the least distance. Distance in a plane between two points \((x_1,y_1)\) and \((x_2,y_2)\) in the plane is 
\[ \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \]
and the total distance is
\[ \sum \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \]
and can be determined before the actual operation of the robot. Thus, time for traverse can be determined using the distance covered by the robot.

The following schematic sketch explains the Principle of Least Distance [18] adapted to find out the minimum distance of travel by the robot. Note that these points are primarily located by the GPS and a substantial error in the GPS coordinates value may result in an entire new path set for traverse of the robot.

![Fig 8: Principle of least distance](image)

- Start and End Points
- Intermediate Points
- Distance between any 2 points
4.3 Obstacle Avoidance Mode

4.3.1 Principle of Voronoi for Obstacle Avoidance

The principle of Voronoi\textsuperscript{21} involves the generation of the maximum deviant path from each obstacle in the direction of traverse. The optimum path of traverse is computed in stages of interpolation for this deviant path. The principle used for path generation may require the navigation robot to operate with two constraints. Firstly, the path through the terrain from a start point to an end, independent of number of obstacles. Secondly, the path through the terrain from a start point to an end following each obstacle and returning data pertaining to each obstacle interpreted through the sensors.

\textbf{Fig 9: Voronoi curve generation}
Chapter 5: Case Analysis

5.1 Case I: Navigation of an AGV through the matrix of points using the path of traverse from a start point to the end-point and independent of number of obstacles present in the domain.

This is a special case of a navigational problem where the AGV is required to traverse from the start point to the end point, without colliding with any obstacles present in the domain. All the control points present in the domain need not be used for computing the path of travel by the AGV. Only those obstacles in the way of the AGV should be considered. This kind of a navigational problem is widely prevalent in several AGV applications.

There maybe two situations of the same navigational problem depending on the type of environment. There is a different kind of a formulation, if the environment is known and the obstacles are definite and static, or when the environment is unknown.

When the environment is known and the obstacles are definite and static, the control points influencing the path of the AGV are those which are closer to the line connecting the start and the end points. Those control points away from the line connecting the start and the end points, by an extent more than the width of the AGV can be ignored. Having these set of control points, a curve must be generated from the start to the end point, in a manner that the control points deviate the curve away from it. Hence, the principle of voronoi curves can be used. A voronoi curve using a set of control points
deflects the curve away from those control points. The deflection is brought-out by the relative weight-age each control point has in affecting the curve. In some cases, some control points can affect the curve stronger than the other control points. In which case, the deflection near the stronger control point will be larger than the other control points.

The logic used to give weights to these control points are the distance of the control point from the line connecting the start and the end points. If the distance from that line to the control point is less, then the weight of that control point will be more, and vice-versa.

**Fig 10: Voronoi curve generation for Case I**

- Intermediate Points
- Intermediate Points influencing the curve
- Start & End Points
The applications for voronoi for navigational algorithm purposes have not been tested. However, the concept that the voronoi curves are deflected by control points, and the flexibility of using the obstacles in the domain as control points for curve generation, voronoi generated curves can be used for curvilinear path generation.

**Case Example: The Navigational Challenge: Navigation Contest at International Ground Vehicles Competition**

The challenge in this event is for a vehicle to autonomously travel from a starting point to a number of target destinations (waypoints or landmarks) and return to home base, given only a map showing the coordinates of those targets. Coordinates of the targets will be given in latitude and longitude as well as in meters on an x-y grid. Construction barrels, trees, and light poles will be located on the course in such positions that they must be circumvented to reach the waypoints. The map of the domain shows a typical course for the navigation challenge for the contest. Coordinates on this map are in meters measured from the origin.

**2003 Navigation Map: GPS Waypoint locations**

<table>
<thead>
<tr>
<th>Waypoint</th>
<th>Latitude (North)</th>
<th>Longitude (West)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oakland Univ. – Start / Finish</td>
<td>42.67845</td>
<td>83.19521</td>
</tr>
<tr>
<td>SAE</td>
<td>42.67904</td>
<td>83.19553</td>
</tr>
<tr>
<td>AUVSI</td>
<td>42.67924</td>
<td>83.19523</td>
</tr>
<tr>
<td>TACOM</td>
<td>42.67892</td>
<td>83.19534</td>
</tr>
<tr>
<td>GM</td>
<td>42.67901</td>
<td>83.19517</td>
</tr>
<tr>
<td>CSI – Wireless</td>
<td>42.67907</td>
<td>83.19480</td>
</tr>
<tr>
<td>DARPA</td>
<td>42.67883</td>
<td>83.19487</td>
</tr>
<tr>
<td>NDIA</td>
<td>42.67866</td>
<td>83.19482</td>
</tr>
<tr>
<td>Theta Tau</td>
<td>42.67859</td>
<td>83.19550</td>
</tr>
</tbody>
</table>
**Fig 11: Practice Navigational Map**

<table>
<thead>
<tr>
<th>Latitude North</th>
<th>Longitude West</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.678</td>
<td>83.1946</td>
</tr>
<tr>
<td>42.678</td>
<td>83.1948</td>
</tr>
<tr>
<td>42.678</td>
<td>83.1950</td>
</tr>
<tr>
<td>42.679</td>
<td>83.1952</td>
</tr>
<tr>
<td>42.679</td>
<td>83.1954</td>
</tr>
</tbody>
</table>

**Suggested Path**

![Map of Suggested Path]
5.2 **Case II: Navigation of an AGV through the matrix of points using path of traverse from a start point to end point, having a fixed roaming distance “D” from the obstacles present in the domain.**

Considering the domain of obstacles defined as matrix points, where the AGV is required to perform its navigation based on a fixed roaming distance from each obstacle present in the domain. The roaming distance is constant and should be sufficient for the sensors to take reference of the obstacles in course of its path. Unlike the previous case, the AGV should traverse the domain taking in consideration of all the control points in the domain. Therefore, the path formulation is more complicated. The curve generation in this case can follow either the curve-fitting using indexing the reference points as discussed earlier or using the principle of voronoi.

If curve-fitting with indexing reference points is used, then the previous illustration with three intermediate points can be extended to any number of intermediate points. In this case, the path is pre-determined upon formulation and the time of travel and the distance traveled is known prior to the start of the AGV. However, this method has its own limitation. One of the limitations is when two control points are close to each other, in such a way that their circles defining the roaming distance overlap with each other. When this happens, the optimum route is not the route which comes through indexing the reference points. Instead, the optimum path is traveling tangential to both the control points roaming distance circles.
When the principle of voronoi is used to generate the curve, there is no specific rule in assigning weights to the control points, which are very close to each other. When this happens, two points deviates the curve equally towards each other, causing the curve not to move in either way. So the resulting curve through voronoi generation will not be the optimum path. It may even cause the AGV to interfere beyond the roaming distance, and may cause the AGV to collide with the obstacle.

*Fig 12: Curve-fitting using Indexing reference points and Voronoi generated Curves*
5.3 Introduction of Hybrid model of curve generation:

This specific case with a crowded matrix of control points was taken to demonstrate the limitations of both the curve-fitting using indexing reference points and the voronoi curve generation. However, a hybrid model for curve generation using Straight-line routing, Curve-fitting using indexing reference points and voronoi generation can be used. When this hybrid model is considered, the matrix is divided into sub-segments which can be handled individually by any of the three navigation procedures.

Fig 13: Hybrid Model of Curve Generation
Chapter 6: Advantages and Disadvantages

Advantages:

- Curvilinear routing is a more advanced formulation of a navigation algorithm
- Curvilinear routing has only one acceleration period and one deceleration period. Therefore, the AGV does not stop or slow down in between its course of traverse.
- Curvilinear routing has only one constant velocity period. Hence, the time of traverse at maximum velocity (Constant velocity) is higher. Therefore the time taken for traverse is less compared to straight-line routing.
- Curvilinear routing also provides an alternative for the straight-line routing, for the mobile robot to decide the navigation mode.
- Curvilinear routing formulates the curved path even before the mobile robot begins to move at the initial point. Hence, it has more control of time of travel, distance of travel and power consumption.
- Curvilinear routing helps the mobile robot to become more flexible and adaptive. The AGV may choose any of the modules – Curve-fitting using indexing reference points, Voronoi curve generation and Hybrid curve generation, based on the type of environment and the orientation of the control points in the domain.
**Disadvantages:**

- Curve generation is based on two dimensional projection of a three dimensional world plane. Hence, the formulation will not be accurate in areas where the height gradient is high.
- Complicated formulation and coding
- High hardware, system and processor requirements
- Highly adaptive sensors and encoders required
- Fails when the control points are too crowded
Chapter 7: Alterations and Scopes for future research

7.1 Three dimensional curve generation

Fig 14: Three dimensional curve generation

The three-dimensional curve generation is similar to the two-dimensional curve generation. The points of interests are placed in a three-dimensional coordinate system and the shortest straight-line route is determined. In the above illustration, the shortest straight-line routing is assumed to be from Point 1 to Point 2, and then to Point 3. With this routing scheme, curvilinear path from Point 1 to Point 3 is traced through Point 2 using any of the previous explained curve generation techniques. In real world, this three-dimensional curve routing is very appropriate to take care of the increase or decrease in the gradient.
7.2 *Regenerating curve with a new initial point*

In certain cases, the curvilinear path from the start point to the end point has a steeper turn in its course. Steep turn of the AGV on its curvilinear course may lead to instability of the AGV and, on extreme cases, may even cause the AGV to topple down. So, the objective of regenerating the curve along the curvilinear traverse of the AGV is to flatten the course as the AGV progresses within the control points of the domain. By re-iterating the initial point (start point), the AGV generates a completely new curved track which is comparatively flatter than the original curved path.

*Fig 15: Regenerating curve with a new initial point*
7.3 Curvilinear routing using random routing technique

Fig 16: Curvilinear routing using random routing technique

For the mobile robot to transform into an intelligent and an adaptive system, it needs to exhibit options of multiple curved paths. These multiple curved path options are critical if the control points are placed in a manner that may produce different curved path with same traversing time and distance. The above illustration shows three options of curvilinear paths in a domain of six control points. If the AGV can lay a set of ‘n’ options of exploring a domain, we could be certain that the domain has been completely inspected in different angles and perspectives. Further, this behavior of the AGV to draw multiple paths takes away its monotonous behavior with respect to one particular domain.
7.4 Laser guided curvilinear traverse

Laser scanner is an important tool for obstacle detection and avoidance. In an objective which needs the AGV to explore the domain and also avoid other obstacles, we need a proper synchronization between the curvilinear navigational module and the obstacle avoidance system. This synchronization may be a challenge in most unstructured environments where the details of the points of interests with respect to real-world conditions are very limited.

The following is the profile of domain from laser scanner image as seen by the AGV.

Fig 17: Laser Scanner image [courtesy: http://www.laseroptronix.com/ladar/morb.html]

Thus the laser scanner can provide an active feedback loop for the navigational algorithm during the traverse.
7.5 *Traversing beyond a cliff: Pro-logical routing*

**Illustration 1:**

When the AGV is negotiating a path on a flat plain, even objects which are suspended directly above the AGV can become alien, if the range of the AGV is not sufficient to cover the object. The illustration shows a red cross suspended on a ceiling and the AGV (black dot) moving on Path 1. The range (white circle) does not reach the suspended object. If this object is a control point or a point where an observation is to be made, the AGV must recognize that the object is beyond reach instead of missing the object completely. Further, the AGV must request an inclination in height sufficient enough to observe the suspended object, as depicted by Path 2.

**Fig 18: Traversing beyond a cliff – Pro-logical routing**
Illustration 2:

When an AGV approaches a cliff, it must be able to recognize that there exists no path beyond the cliff. Further the AGV must be able to decide the possible route to the next point of observation avoiding the cliff.

All of these scopes for the AGV provide an active involvement of real-time real-world experience and challenges.
Chapter 8: Algorithm for the formulation

8.1 Straight line routing:

1. Input Start and End points
2. Input Static control points
3. Compute the planar projected control points
4. Define roaming distance and interfering distance
5. Compute the path of minimum traverse
6. Simulate velocity profile
   - Find out traverse time and distance
7. Launch the AGV
8.2 Curvilinear routing:

8.2.1 Curve-fitting using indexing reference points

Input Start and End points

Input Static control points

Compute the planar projected control points

Define roaming distance and interfering distance

Compute the path of minimum traverse

Draw tangential intersects at intermediate control points – “Reference Points”

Project reference points on the circle defining the roaming distance – “Revised Reference Points”

Fix start point, new control points (revised reference points) and end points

Fit a curve in the control points in the domain. Compute time of curvilinear traverse
1. Compare straight-line traverse time and curvilinear traverse time.

2. If straight-line traverse time greater than curvilinear traverse time:
   - Define mode of traverse as curvilinear traverse.
   - Compute the path of minimum traverse.
   - Draw tangential intersects at intermediate control points – “Reference Points”.
   - Project reference points on the circle defining the roaming distance – “Revised Reference Points”.
   - Fix start point, new control points (revised reference points) and end points.
   - Fit a curve in the control points in the domain. Compute time of curvilinear traverse.
   - Simulate velocity profile under curvilinear travel.
   - Find out traverse time and distance.

3. If straight-line traverse time less than or equal to curvilinear traverse time:
   - Define mode of traverse as straight-line traverse.
   - Compute the planar projected control points.
   - Define roaming distance and interfering distance.
   - Compute the path of minimum traverse.
   - Simulate velocity profile under straight-line travel.
   - Find out traverse time and distance.

4. Launch the AGV.
8.2.2 Voronoi curve generation

1. **Input Start and End points**

2. **Input Static control points**

3. **Compute the planar projected control points**

4. **Define roaming distance and interfering distance**

5. **Compute the equation of the line connecting start point & end point - “Line of Travel” and \( i=1 \)**

6. **Draw the perpendicular distances from all control point to the “Line of Travel”**

7. **Is the perpendicular distance from all control point greater than “0.5 * Robot Width + \( \varepsilon \)”**

   - **NO**
     - Find the number of control points interfering within “0.5 * Robot Width + \( \varepsilon \)” - “Revised control points”
   - **YES**
     - Line of travel is along the connecting line from the start and end points.

8. **Compute voronoi curve for the revised control points**

- **C1**
- **C2**
Load Straight-line navigational algorithm

Compute time of travel and distance

Simulate the Velocity profile

Launch the AGV

Load Curvilinear navigational algorithm

Compute time of travel and distance

Simulate the Velocity profile

Launch the AGV
8.2.3 Hybrid curve generation

- Input Start and End points
- Input Static control points
- Compute the planar projected control points
- Define roaming distance and interfering distance
- Compute path of minimum traverse, \( i = 1 \)
- Compute distance between \( i^{th} \) and \( i+1^{th} \) control points = \( D \)
  - If \( D \leq 2 \times \text{Roaming distance} \) NO
    - Draw a curvilinear path through the \( i+1^{th} \) point
    - Append Track
  - YES
    - Draw a straight line from the \( i^{th} \) till \( i+1^{th} \) point
    - Append Track
- Is \( i+1^{th} \) point = end point NO
  - YES, \( i = i+1 \)
- C
Load Track

Compute time of travel and distance

Simulate the velocity profile

Launch the AGV


Chapter 9: Conclusion

This study finds its suitability to the Bearcat Robot III based on its mechanical design. Bearcat Robot III has two independent motors for the front wheels and can operate at different motor speeds. The basis for the suggested curvilinear path is obtained by the differences in speeds in wheel spins. The tracking of the curve can be facilitated by the encoders and sensors, by which the AGV can upgrade on its path. Thus, the study portrays that an efficient navigational algorithm can be developed using the approximation of the straight-line route to generate a curvilinear path of traverse.

The time of travel in straight line routing and in curvilinear routing is also compared. The two constants, $K_1$ and $K_2$, are found, which are the difference of acceleration periods and deceleration periods and the difference in constant velocity periods. ‘$K_1$’ will always be non-negative, as the factor "n-2" is always non-negative, where ‘n’ is the number of control points. This is because, the start and the end point is always specified. The term, ‘$K_2$’, maybe positive or negative based on the difference in the constant velocity periods between straight-line routing and curvilinear routing. In general, if there is more number of control points then time for curvilinear routing is definitely less than the time for straight-line routing.

Further, this approach gives the AGV, for a set of points, the option of selecting between the straight line routing and curvilinear routing. Within curvilinear routing, the option of curve-fitting using indexing reference points, voronoi curve generation and
hybrid curve generation techniques have been discussed. Each of these procedures has its own uniqueness. Straight-line routing is the easiest and is the best method is there are only two points in the domain. A curve-fitting using indexing reference point is an adaptive procedure and can be extended to any number of control points. Voronoi Curve generation is a useful procedure for control point avoidance without the need to observe any control point. Hybrid curve generation overcomes the limitation of crowded matrix of control points in a domain, and generates a curve in segments. The selection of any one of these procedures is, however, based on certain justifications made by the AGV in terms of efficient navigation with respect to the given set of constraints. So, an amount of flexibility and the ability of the AGV to decide its choice are provided by this additional navigational algorithm.
Footnotes:

1. Automated Guided Vehicle
2. Check, limiting factors, obstruction
3. Responsive and Adaptive machines
4. Area of interest
5. Negotiate, navigate
6. Period in which the AGV accelerates from rest to the maximum speed
7. Period in which the AGV travels at constant maximum speed
8. Period in which the AGV decelerates from maximum speed to a stop position
9. AGV, traveling in a straight line from one point to another
10. AGV, traveling along a curved path with a small change in directional properties
11. Colliding
12. Navigational method, using encoders to measure wheel rotation and/or steering orientation
13. Navigational Method using gyroscopes and accelerometers to measure rate of navigation
14. Navigation method which computes the absolute position of the AGV by measuring the direction of incidence of three or more actively transmitted beacons
15. Navigation method using distinctive landmarks at known locations
16. Navigation method using a map or a world model of the environment
17. Navigation method using worldwide radio navigation system formed from a 24 satellite constellation network
18. Flushing
19. Level of charge in batteries
20. the straight line or shortest distance is the natural motion of a body, Newton's first law of motion
21. Generating a voronoi graph to help the robot traverse in an array of points.
References:


17. http://math.uc.edu/~kingjt/Matlabcourse/Matlab_course/interp_poly/interp_poly.html