Way-point autonomous navigation in unstructured environments

J. Shen, Z. Gong, J. I. Guzmán¹, A. Tay², and M. Adams³

Abstract – A key component for autonomous navigation is path planning. It determines the manner in which the vehicle will move from its start position to the desired target, based on information supplied by onboard sensors. The results of a path-planning algorithm that allows the safe guidance of a skid-steer vehicle evolving in a jungle like terrain are presented. The algorithm uses the modified distance transform plus a vector field histogram for path planning. It takes into account the dynamics of the vehicle and uncertainties in the environment models available from the perception mechanisms. The resulting system has been implemented on a modified drive-by-wire vehicle and deployed in the jungle at speeds of up to 4 m/s and distances beyond 2 km.

Keywords: Modified distance transform, Way-point, Local path planning, Navigation, Autonomous unmanned ground vehicle

1 BACKGROUND

There are considerable interests on the use of Autonomous Unmanned Ground Vehicles (AUGVs) for industrial, commercial and defence related applications. To guide a vehicle autonomously, we need to know where it can move, where it is and where it needs to move. An AUGV is equipped with a perception system to determine areas where the vehicle can traverse, a localisation system to know its position and a path planner to guide the vehicle to the target position whilst avoiding obstacles. That is, perception, decision-making and actuation are required to function in a continuous manner in real-time.

Earlier work employed ultrasonic range sensors and machine vision for vehicle guidance [2-4]. Localisation was achieved using dead-reckoning [5] or beacons [6] and landmarks [7]. Currently other perception systems are deployed notably time-of-flight lasers, stereovision, radar and infrared cameras [8], [9,10]. For localisation purposes, there is extensive use of GPS, inertial sensors or both [8,11].

A core component for vehicle navigation is the path planning and obstacle avoidance algorithm; it determines the vehicle motion. Pioneering work by Khatib presented the concept of Artificial Potential Fields for real-time obstacle avoidance [12]. Moravec and Elfes introduced the concept of Certainty Grids, a map representation well suited for sensor data accumulation and sensor fusion [3]. By integrating two concepts, Boren-stein and Koren developed the Virtual Force Field method (VFF) [13]. An enhancement to this is the Vector Field Histogram (VFH); it searches gaps in the locally constructed polar histogram [14]. Our experiments using VFH have shown that there are still some disadvantages; it is difficult to select suitable threshold values when faced by T-shaped junctions. The Modified Distance Transform (MDT) [15] is a simple, fast path planning algorithm, which can be introduced to help solving the T-shaped junction problem with the VFH.

2 OBJECTIVE

The purpose of this research was to develop a navigation system capable of guiding autonomously a 12-ton vehicle within a natural from a starting position to given global way-points in day and night conditions.

3 METHODOLOGY

The major challenge for a path planner is that it can only be tested when all the system components are in operation. The planner needs environment information, positioning data, for the vehicle to be computer controllable, etc. Only when these components are available, the algorithms will be truly tested, as the vehicle will move under its control.

Initially, a path planner simulator was built to demonstrate principles and test our approach; it used simulated sensor data and later data streams from our perception systems. Next, tests were performed on a pickup where sensors were mounted with a driver guiding the vehicle using data generated from the navigation unit. This enables the testing of algorithms in quasi real-conditions. Finally, once the vehicle was fully converted onto a drive-by-wire unit and the perception systems tested at the target terrain, it was possible to close the autonomous navigation loop. It is only then when it was possible to determine whether or not the vehicle could be guided autonomously.

3.1 Experimental Vehicle

The experimental vehicle (Fig. 1) was installed with SICK laser scanners and DIGICLOPS stereo

¹ Research Division Electronics Systems Department, Renault, France
² School of Computer Engineering, Nanyang Technological University, Singapore
³ School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
vision cameras for local obstacle and slope detection, GPS and IMU for localisation, IR camera for obstacle detection at night and microwave radar for collision avoidance. The set up in the pickup was used first to acquire data that gives streams of scene information for vehicle guidance coupled to localisation data in order to use it for off-line for algorithm development. The second was to make dry runs in which a driver guides the vehicle using data generated by the planner in the form of heading and speed. The rationale is to reduce the risk of closing the loop when commissioning the algorithms onto the large vehicle.

The architecture of the navigation system is shown in Fig 3. It consists of two major components: the Planner; and the Heading & Speed unit. The planner generates the vehicle next desired position that guides it towards the global way-point, whilst the Heading & Speed Unit determines the vehicle speed and heading according to its dynamic constraints and current pose. The planner uses a global and local map to generate the trajectory; it then affects a traversability test of the terrain via an elevation analysis using the estimated slope information from the guidance sensors. The vehicle pose provides feed-back information as well as vehicle dynamics data. Information on road availability (centroids), occupancy maps from the combined perception systems are used as inputs. The diagram also shows the vehicle following block, in operation when in the vehicle following mode it provides global way-points used as target locations for the vehicle to follow, further details can be found in [18].

### 3.2 Navigation System

The navigation system closes the external loop of the control system guiding autonomously the vehicle, as shown in Fig. 2. The desired position is given by the mission commands, the feedback and guidance is provided by localisation data and obstacle map, it then generates a correction signal in the form of heading and speed that are fed into the plant (the vehicle, which in itself has a control system), to guide the vehicle towards the target position.

The architecture of the navigation system is controlled by the Master Controller which determines the working modes between “cross country” and “vehicle following” according to the functionality desired by the mission operator.

### 3.3 Motion Control

The large weight (12-tons) of the vehicle and relative high speed (4 m/s) imply that its dynamics and kinematics must be included to plan points physically feasible.

#### 3.3.1 Simple Kinematics Model of Wheeled Vehicle

Consider the simple model of a wheeled vehicle as given in Fig. 4. Let the steering angle \( \alpha \) be the turn angle of the front wheel (Fig. 4). Assuming no slip-page, the turn curvature \( \rho \) (the inverse of the turn radius \( R \)) of the centre of the front wheel is given by:

\[
\rho \equiv \frac{1}{R} = \frac{1}{L} \sin(\alpha)
\]  

where \( L \) is the distance between the centres of the front and rear wheels, or roughly the vehicle length. Then vehicle turn speed \( \omega \) is given by:

\[
\omega = \frac{v}{L} \sin(\alpha)
\]  

where \( v \) is the vehicle forward speed.

Notice, that in this model, \( L \) is the minimum turn radius when the steering angle \( \alpha \) is 90°. With a fixed \( L \),
the turn radius is solely determined by the steering angle $\alpha$. The vehicle forward speed $v$ has no influence on the turn path; with the vehicle turn speed $\omega$ proportional to the vehicle forward speed $v$.

\[ L v \left( 1 - i_R \right) \omega_R r \] (3)
\[ V_L = \left( 1 - i_L \right) \omega_L r \] (4)
\[ V_x = \frac{V_R + V_L}{2} \] (5)
\[ V_y = i_y V_x \] (6)
\[ \psi = \frac{V_R - V_L}{b} \] (7)

where $r$ is the track rolling radius, $b$ is the tread, $\omega_R$ and $\omega_L$ are the angular velocities of the right and left track sprockets, $i_R$, $i_L$, and $i_y$ are the slip coefficients, $\psi$ is the vehicle turning speed in the global reference $\{X, Y, t\}$, i.e. $\psi = d\psi/dt$.

In the global reference $\{X, Y, Z, t\}$, using standard coordinate transformations, we have the following equations:

\[ V_x = V_y \cos \psi - V_y \sin \psi \] (8)
\[ V_y = V_y \sin \psi + V_y \cos \psi \] (9)
\[ \psi = \frac{V_R - V_L}{b} \] (10)

where $V_x$ and $V_y$ are respectively the vehicle speed components in the X and Y directions.

Fig. 4. A simple wheeled vehicle steering model.

3.3.2 Kinematics Model of AUGV

In the instantaneous vehicle-fixed reference $\{x, y, t\}$ of a tracked vehicle, let $V_R$ and $V_L$ be the forward speeds of the right and left tracks, respectively. The notations of the outside and inside track speeds are avoided, as it could be troublesome for system implementation. The vehicle may turn right or left during motion (we cannot keep on switching the vehicle models all the time.) Let $V_x$ and $V_y$ be the longitudinal and lateral speed components of the vehicle centre, respectively. Let $\psi$ be the orientation of the vehicle in the global reference $\{X, Y, t\}$ (see Fig. 5). We then have the following basic equations on tracked vehicle kinematics:

\[ V_R = \left( 1 - i_R \right) \omega_R r \] (3)
\[ V_L = \left( 1 - i_L \right) \omega_L r \] (4)
\[ V_x = \frac{V_R + V_L}{2} \] (5)
\[ V_y = i_y V_x \] (6)
\[ \psi = \frac{V_R - V_L}{b} \] (7)

where $r$ is the track rolling radius, $b$ is the tread, $i_R$, $i_L$, and $i_y$ are the slip coefficients, $\psi$ is the vehicle turning speed in the global reference $\{X, Y, t\}$, i.e. $\psi = d\psi/dt$.

In the global reference $\{X, Y, Z, t\}$, using standard coordinate transformations, we have the following equations:

\[ V_x = V_y \cos \psi - V_y \sin \psi \] (8)
\[ V_y = V_y \sin \psi + V_y \cos \psi \] (9)
\[ \psi = \frac{V_R - V_L}{b} \] (10)

where $V_x$ and $V_y$ are respectively the vehicle speed components in the X and Y directions.

3.3.3 Velocity Control

The principle of the vehicle total energy is applied [16] to derive a control law for the AUGV. This is expressed in Eq. (11):

\[ \dot{x} = K_1/K_2 \left( x_d - \left( 1 + s^2 \frac{M}{K_1} \right) x \right) \] (11)

where $s$ stands for $\delta/\delta t$, $M$ is the mass of AUGV, $K_1/K_2$ is the gain constant and $\chi_d$ is the target point.

From Eq. (11), the velocity signal can assume any values, dictated by $\chi_d$, $X$. In reality of course, an AUGV cannot travel at any speed and will be limited to $\pm U$ m/s. We can take this into account, by replacing the linear amplifier with gain $K_1/K_2$ (Eq. (11)), with a non-linear ideal saturation with the same gain but saturation levels of $\pm U$. The same is true for torque controlled motoring, in that the non-linearity would produce a saturated torque signal. With this inclusion, we can consider the response of the motor dynamics to unbounded velocity (or torque) signals, since the speed controller now restricts the motor input. Hence, a realistic control loop which models the derived velocity control law is shown in Fig. 6.

From the data flow of navigation systems view, the path planner (details discussed in the following section) will keep sending the new desired target point to the velocity controller based on the desired path trajectory.

Fig. 5. A schematic diagram of kinematic model of AUGV.

Fig. 6. Non-linear control system representing the derived velocity control law. The linear region within the saturation element has a gradient $K_1/K_2$.

From the data flow of navigation systems view, the path planner (details discussed in the following section) will keep sending the new desired target point to the velocity controller based on the desired path trajectory.
3.4 Hybrid Local Path Planning Algorithm

We analysed the VFH algorithm and MDT algorithm. Experiments showed that the VFH faced the threshold tuning difficulties when at T-shaped junctions while the MDT made the robot move too close to the obstacles. Our approach is to hybridise the VFH with MDT. Before presenting how these can be combined, we introduce the two algorithms first.

3.4.1 Vector Field Histogram Algorithm [14]

The Vector Field Histogram is generated by the following Eqs. (12)-(15):

\[
\beta_{i,j} = \tan^{-1}\left(\frac{y_j - y_0}{x_j - x_0}\right) \tag{12}
\]

\[
m_{i,j} = (c^*_{i,j})^2(a - bd_{i,j}) \tag{13}
\]

where:

- \(a, b\) – Positive constants
- \(c^*_{i,j}\) – Certainty value of active cell \((i,j)\)
- \(d_{i,j}\) – Distance between active cell \((i,j)\) and the VCP
- \(m_{i,j}\) – Magnitude of the obstacle vector at cell \((i,j)\)
- \(x_0, y_0\) – Present coordinates of the VCP
- \(x_i, y_j\) – Coordinates of active cell \((i,j)\)
- \(\beta_{i,j}\) – Direction from active cell \((i,j)\) to the VCP

For each sector \(k\) (Fig. 7), the polar obstacle density \(h_k\), is calculated by:

\[
h_k = \sum_{i,j} m_{i,j} \tag{14}
\]

where \(k = \text{INT}\left(\beta_{i,j} / \alpha\right)\), \(\alpha\) is sector \(k\)’s angle.

Figure 8 shows an example of a polar histogram. From it, we can decide the traversable valleys according to the selected threshold. Combining with the target position information, we finally can decide the valley the robot should move in.

Figure 9 illustrates how to decide the steering reference angle \(\theta\) under four different situations. It tells that the value of \(\theta\) searching sector \(S_{\text{max}}\) affects the distance \(d_s\) between the robot and the obstacle. The bigger the value of \(S_{\text{max}}\), the further away the obstacle is from the robot.

3.4.2 Modified Distance Transform (MDT) Algorithm

The algorithm begins with an initialisation of a 2-dimensional (2D) \(m \times n\) grid map into cells \(CL_{ij}\) [where \((i,j)\) is found on the coordinate \((m,n)\)] with a cell traversal cost of \(C_{ij}\) and a cell target value \(Tv_{ij}\) initialised to a very large number \(k\). \(C_{ij}\) represents the minimum total traversal cost from the cell \(CL_{ij}\) to the end point cell. C-space is the area expanded from obstacles with half width of the AUGV.
In Fig. 10, we assume cell (0) as end point and cell (112) as start point. The following procedures find the shortest path from start point (112) to end point (0):

1) Find the children of end point, put them into the FIFO stack
2) Pop the cell in FIFO stack, calculate the target value of the cell\[ T_{ij} = C_{ij} + \min(\text{neighbour’s } T_v) \] and find the cell’s children, then put them into FIFO stack
3) Repeat step 2 until FIFO stack is empty or have reached the start point cell
4) From the start point, search the neighbour with the minimum \( T_v \), if there is more than one candidate, choose the one with the shortest distance to the end point
5) From the chosen cell, search the neighbour with the minimum \( T_v \), if there is more than one candidate, choose the one with the shortest distance to the end point
6) Repeat step 5 until the end point is reached
7) If the end point can not be reached, then no solution

Thus, we find the shortest path (black line) from the start point cell (112) to the end point cell (0). In order to find the curvature of the path, we adopt the Bezier spline to smooth it. This is expressed by:

\[ P(t) = \sum_{i=0}^{n} B_i^n(t)P_i \quad (0 < t < 1) \]  

where \( B_i^n(t) = C_i^n t^i (1-t)^{n-i} \), \( C_i^n = n! / i!(n-i)! \).

Now, we can obtain segments of the path curve within the allowed error \( \epsilon \). By putting the endpoint of the first segment as \( \chi_d \) in Eq. (11), we can constrain the AUGV’s velocity.

### 3.4.3 Hybrid Algorithm

As referred, it is difficult to determine the threshold at T-shaped junctions. If the threshold is set too small (due to dynamic environments, it is hard to tune and pre-set a suitable threshold), there will be no traversable valleys in front of the AUGV. The MDT is used to guide the AUGV (see Fig. 12), with the VFH used to move the AUGV away from the obstacles by selecting a suitable \( S_{\text{max}} \), as shown in Fig. 13.

Thus, we find the shortest path (black line) from the start point cell (112) to the end point cell (0). In order to find the curvature of the path, we adopt the Bezier spline to smooth it. This is expressed by:

\[ P(t) = \sum_{i=0}^{n} B_i^n(t)P_i \quad (0 < t < 1) \]  

where \( B_i^n(t) = C_i^n t^i (1-t)^{n-i} \), \( C_i^n = n! / i!(n-i)! \).

Now, we can obtain segments of the path curve within the allowed error \( \epsilon \). By putting the endpoint of the first segment as \( \chi_d \) in Eq. (11), we can constrain the AUGV’s velocity.

### 3.4.4 Global Versus Local

The target way-point in the world frame is achieved by successful attempts of reaching a local target point in the body frame. The local target point is selected among the preset candidates (red dots in Fig. 14) with the shortest distance to the global way-point. The coordinate transform between body frame \( \{x, y, z\} \) and world frame \( \{N, E, D\} \) is based on Eq. (17):
where $c \theta = \cos \theta$ etc. $\theta$, $\Phi$, and $\psi$ represents the pitch, roll, and yaw of the vehicle. In Eq. (17), $P(N_v, E_v, D_v)$ is the coordinates of body frame’s origin in the world frame.

\[
\begin{bmatrix}
N^v \\
E^v \\
D^v
\end{bmatrix} = 
\begin{bmatrix}
\cos \psi + \sin \Phi \sin \psi & -\cos \Phi & \sin \Phi \\
-\sin \psi + \sin \Phi \cos \psi & \cos \Phi & \sin \Phi \\
\sin \Phi \sin \psi & -\sin \Phi \cos \psi & \cos \Phi
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} + 
\begin{bmatrix}
N_v \\
E_v \\
D_v
\end{bmatrix}
\]

(17)

\[
\begin{align*}
\text{Trajectory} (17)
\end{align*}
\]

Fig. 14. The relationship between world and body coordinate frames.

4 RESULTS & DISCUSSION

Test results have showed that the navigation system is able to guide the AUGV through sparse obstacles fields and narrow roads without oscillation and to reach several global way-points (refer to Figs. 13 and 15). The max driving speed has attained is 4 m/s. In addition, the navigation processing speed is around 5 Hz on an 850 MHz PC. It ensures that the AUGV has sufficient time to react with the environment.

Fig. 15. Playback of obstacle map recorded on-site.

5 CONCLUSIONS

We have presented a unique navigation system based on early evaluation of terrain traversability, and demonstrated this system in a 2.2 km traverse of unmapped cross-country terrain. The overall formulation using the hybrid path planner together with the dynamic constraints is a unique solution that has enabled us to control a large vehicle in safety. The algorithm was verified by actual experimentation. The results show that the algorithm is efficient and applicable.

6 INDUSTRIAL SIGNIFICANCE

Although the technology is developed for outdoor vehicle applications, it can also be used for navigation of AGV or free ranging robot in indoor environments. There are other applications on intelligent vehicles, advanced transportation systems and search & rescue tasks, which can benefit from this technology. In factory compounds, transports of heavy goods alone using forklift trucks or intelligent AGVs can still use this developed technology.

REFERENCES