Presented at SPIE 10th Annual AeroSense Symposium. Conference 2738, Navigation and Control Technologies for Unmanned Systems. Orlando, FL. April 1996.

GPS Aided Retrotraverse For Unmanned Ground Vehicles

Karl Murphy and Steven Legowik

Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
karl.murphy@nist.gov & steven.legowik@nist.gov

ABSTRACT

A computer controlled HMMWV automatically retraces a previously recorded path using navigation data, a process we have termed retrotraverse. A Kalman filter combines the output of two navigation systems, an inertial dead reckoning system and a differential GPS, both with and without carrier phase detection. During retrotraverse, the mobility controller uses a velocity controller and pure pursuit steering. Obstacles, such as another vehicle, can be detected with a laser range imaging device.

Keywords: dead reckoning, GPS, inertial navigation, Kalman filter, LADAR, obstacle avoidance, pure pursuit, retrotraverse, speed control, UGV

1. INTRODUCTION

During 1994 and 1995 the U.S. Army brought a robotic vehicle to Ft. Hood, TX to participate in several brigade training exercises, mock battles involving hundreds of M1 tanks, Bradleys, and HMMWVs. The robot vehicle, an actuated HMMWV, was assigned to the Mustang Battalion, the1-8th Cav, and performed counter reconnaissance missions. There were two reasons to include the robot in their training exercises. First, it allowed military personnel to start thinking about unmanned ground vehicles (UGVs) and how to develop strategies involving them. Second, it gave technologists a chance to experience how military scouts perform their missions and under what constraints they work.

The robotic vehicle contained a collection of sensors and controllers. Several organizations participated in its design and integration. This included the Army Research Labs, Tooele Army Depot, the Jet Propulsion Laboratory, Northrop, Cybernet, David Sarnoff Research Laboratory, and the National Institute of Standards and Technology.

The robotic vehicle is shown in Figure 1. The vehicle performs Automatic Target Acquisition, ATA, using acoustic sensors and a turret equipped with a FLIR, a daylight camera, and laser range finder. The robot vehicle communicates with a base station vehicle using either a satellite communication system or a terrestrial data radio. The mobility controller navigates using inertial sensors, differential GPS, and a laser range imaging device (LADAR). It controls the vehicle using electric actuators on the steering, brake, throttle, transmission, transfer case, and park brake. During setup a human driver enters parameters for the ATA and mobility systems using the vehicle user interface. During operations, a remote operator at the base station, not shown, monitors and controls the ATA and mobility systems. The design and development of the vehicle is described in [1] and an earlier version in [2] and [3].

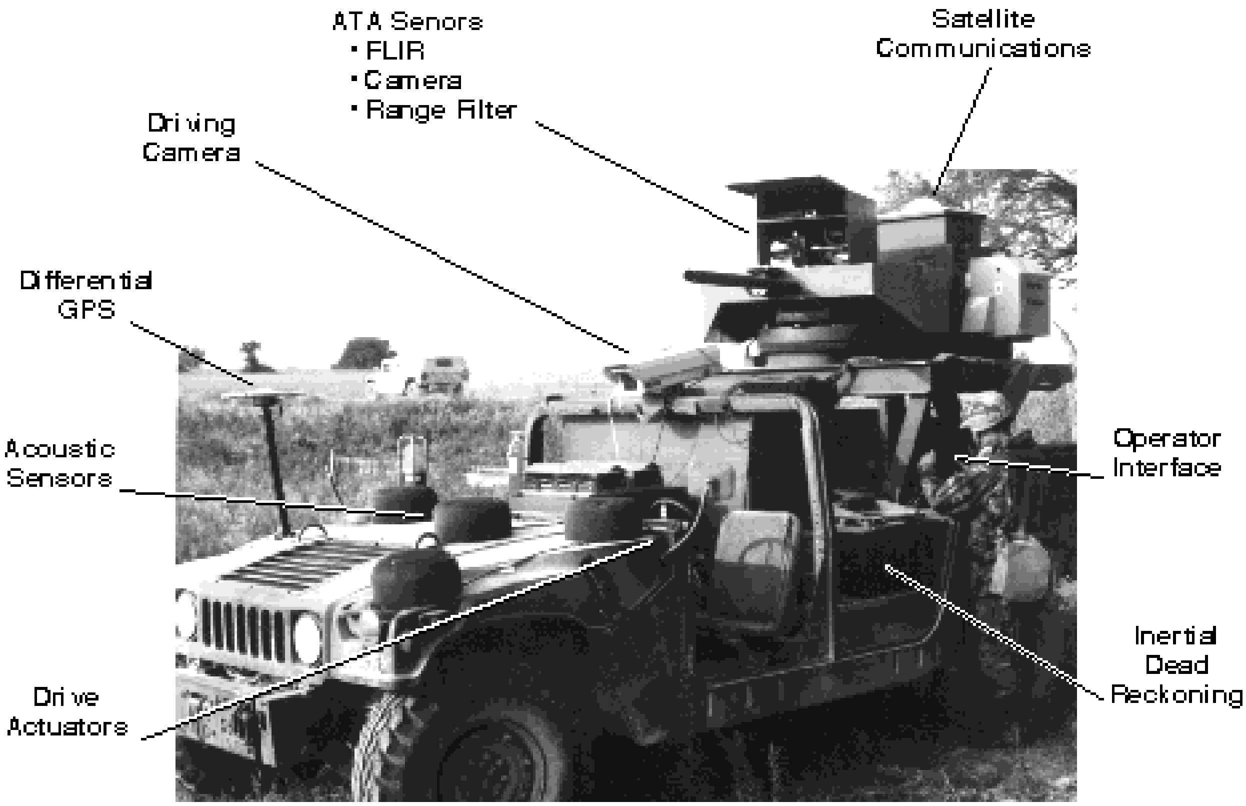

Figure 1. The robotic vehicle in operation at Ft. Hood.

During a typical defensive mission, a scout drives the vehicle to a secondary observation point. This is the location that the vehicle will autonomously drive back to later in the mission. The driver initiates path recording and then drives to the main observation point as the vehicle records the route using the navigation system. If desired, the driver indicates clear areas along the path where the vehicle can turn around. The driver then sets up the ATA system to observe specified locations called named areas of interest. The driver hands over control to the remote operator and leaves the vehicle. The ATA begins to perform its mission under the guidance of the remote operator. When the ATA detects a potential target either from an acoustic signature or from movement in an image, it alerts the remote operator. The operator can then request a zoomed image or range and grid location of the target. At some point during the mission, the remote operator can command the robot vehicle to drive back to the secondary observation point. Then the vehicle automatically retraces the recorded route, a process termed *retrotraverse*. Work is currently underway to integrate a laser range imaging device that will detect obstacles, such as other vehicles, that were not on the path during teaching.

This paper describes the mobility sensors and control. Section 2 describes the inertial and differential GPS sensors and the Kalman filter used to combine their output. Section 3 describes path recording and playback, steering and velocity control, and obstacle avoidance.

2. NAVIGATION SENSORS

The robotic vehicle uses a Kalman filter to combine navigation data from two sensors, an inertial dead reckoning system and a differential GPS sensor. The filter takes the best of both sensors, producing an output with the quick response and high update rate of the inertial dead reckoning system with the accuracy of the differential GPS.

2.1 Inertial Dead Reckoning

The U.S. Army's Modular Azimuth and Positioning System, MAPS, is an inertial navigation system containing three ring laser gyros, three accelerometers, and a rear axle odometer. MAPS can supply orientation and translation data at 5 Hz. AlliantTech developed a Navigation Interface Unit, NIU, to obtain a faster update rate. The NIU requests orientation data from the MAPS, which is available by itself at 25 Hz, and taps into the odometer to measure the distance traveled. It combines the odometer and heading data to generate vehicle position data at 25 Hz.

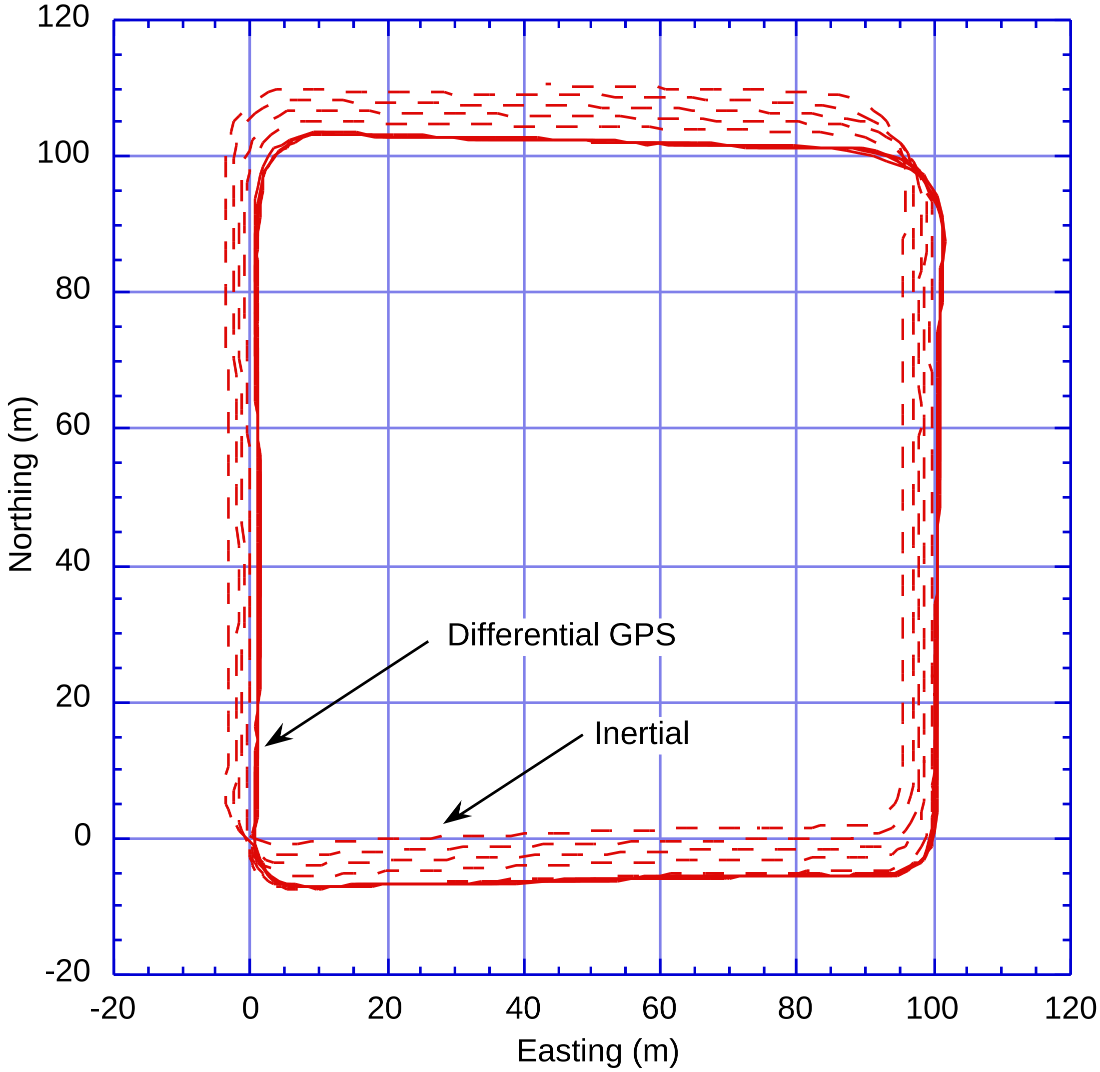

Figure 2. The inertial dead reckoning system drifts as the HMMWV was driven repeatedly around a roughly square course in a counterclockwise direction. The inertial system drifts to the north and to the west by a similar amount each pass. The differential GPS trace provides a truer representation of the path.

The MAPS/NIU position measurement drifts as the vehicle moves, due mainly to wheel slippage. On pavement and in dry grass the system drifts at a rate slightly more than 0.1% of the distance traveled. On slippery surfaces and steep slopes, the system will drift more. Figure 2 shows the inertial dead reckoning system drifting as the vehicle is driven around a square track. In this case the vehicle was driven through patches of snow causing the inertial dead reckoning system to drift at 0.4% of distance traveled.

2.2 GPS

The satellite based Global Positioning System, GPS, has an accuracy of 100 m in the stand alone mode for civilian receivers and around 10 m to 30 m for stand alone military receivers. Higher accuracy data is required for retrotraverse. To improve accuracy, differential GPS is used. A stationary receiver at a known location transmits perceived position errors to the moving GPS receiver. The mobile receiver uses this data to improve its position estimate. Using this technique, accuracies from several meters down to sub-meter can be obtained. Even higher accuracies can be obtained by including carrier phase measurements in the transmissions. Using this carrier phase differential GPS, accuracies of a centimeter or better can be achieved.

The receivers report position data at a rate 1 Hz. Unfortunately, due to computational and transmission delays, the position reported by the receivers can be delayed up to two seconds. Thus, the receivers do not report where the vehicle is, but rather where it was two seconds ago. A position solution can be obtained with less delay, around a quarter of a second, but at a reduced accuracy of around 10 cm.

An important feature of carrier phase differential GPS, available within the last two years on commercial systems, is the ability to resolve the cycle integer ambiguities of the carrier phase while the remote receiver is moving. This allows the vehicle to lose satellite lock (caused by driving under a bridge, near buildings, through foliage, etc.) and then reacquire phase lock on the fly. When there are not enough satellites visible for GPS to calculate an accurate position, the robot relies solely on the inertial dead reckoning system.

2.3 Kalman Filter

Both navigation systems have advantages and disadvantages. The inertial dead reckoning system has a higher data rate of 25 Hz, but it drifts. Whereas the GPS system does not drift over time, but it has a noise component, a slower data rate of 1 Hz, and is not always available. To get the best of both systems, their outputs are combined using a complementary filter, shown in Figure 4.

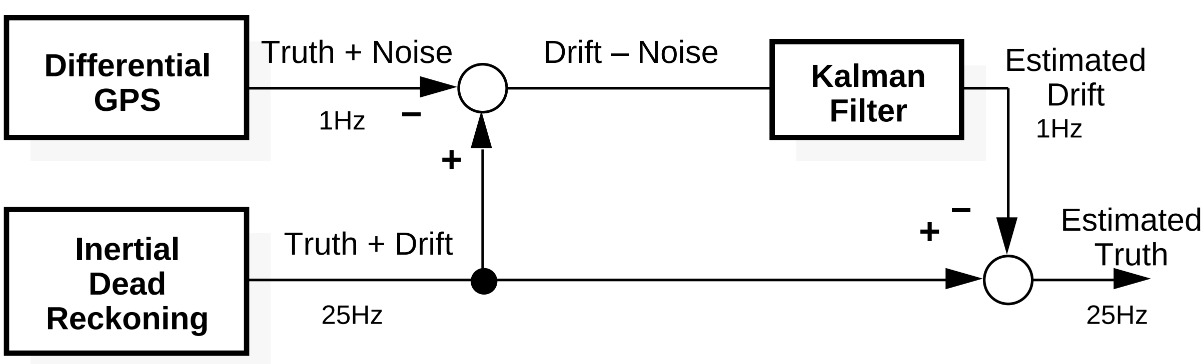

Figure 4. Acomplementary filter is used to estimate the true position of the vehicle.

The vehicle is at some position we will call *truth*. Both navigation sensors try to measure this but have some inherent error. The inertial dead reckoning system reports truth with an error we will call *drift*, and the GPS also reports truth but with an error we will call *GPS noise*. Taking the difference between the output of the two sensors results in the difference of the drift less the GPS noise. The Kalman filter tries to remove the noise and estimate the drift. Subtracting this drift estimate from the inertial dead reckoning system output yields an estimate of the true vehicle position. The Kalman filter runs once for each GPS update, producing drift estimates at 1 Hz. However, truth is estimated at 25 Hz using the current inertial dead reckoning data and latest drift estimate. In general, drift changes slowly allowing the calculation of position between GPS updates and during GPS dropouts.

The Kalman filter uses a simple model for drift that assumes the drift in Northing and Easting are independent of each other. This results in two separate scalar filters. Recall that the filter is estimating drift and not position. The state and output model is

$$\begin{aligned} d_{n+1} &= d_n + w_n \\ z_n &= d_n + v_n \end{aligned} \qquad (1)$$

where

- d_n is the drift at time n,
- w_n is the system noise with covariance q_n,
- z_n is the measured drift (z_n = Inertial - GPS), and
- v_n is the measurement noise with covariance r_n.

The system noise w_n models wheel slippage and other factors that cause the drift to change. As drift tends to increase with the distance traveled, the covariance of the system noise, q_n, is modeled as being proportional to the distance traveled since the last update. The measurement noise v_n is the GPS noise. Its covariance, r_n, is modeled as a constant.

The Kalman filter reduces to following scalar equations (one set for Northing and an identical set for Easting)

$$\begin{aligned} K_n &= p_n / (p_n + r_n) \\ \hat{d}_{n+1} &= \hat{d}_n + K_n (z_n - \hat{d}_n) \\ p_{n+1} &= p_n (1 - K_n) + q_n \end{aligned} \qquad (2)$$

where

- K_n is the Kalman Filter gain, and
- $\hat{d}_n$ is the estimated drift with an error covariance p_n.

The recursive filter is started with an initial guess for $\hat{d}_o$ and p_o.

The position of the vehicle is then estimated as

$$\text{\~{}Truth} = \text{Inertial} - \hat{d}_n \qquad (3)$$

Figure 5 show the results of the Kalman filter for the first three laps of the square course shown in Figure 2. The GPS receiver is operating in the standard differential mode which provides position data accurate to 1 meter. Both the measured drift, z, and the estimated drift, $\hat{d}$, increase in magnitude with time. During each lap the vehicle drifts in a similar direction at the same location producing a repetitive pattern about every 100 seconds. The sensor noise, i.e. the GPS noise, cannot be determined exactly, but it can be estimated as $(z - \hat{d})$. Note the peaks at 50 s, 150 s, and 250 s. These were caused when the vehicle slipped on a snow patch causing a large change in drift which the filter erroneously attributed to GPS noise. This mistake is corrected in a few seconds as further GPS updates corroborate the large change in drift. Figure 6 shows the overall result of the complimentary filter, that is, the estimated position of the vehicle. The estimated position is the difference of the inertial dead reckoning less the drift estimated by the Kalman filter.

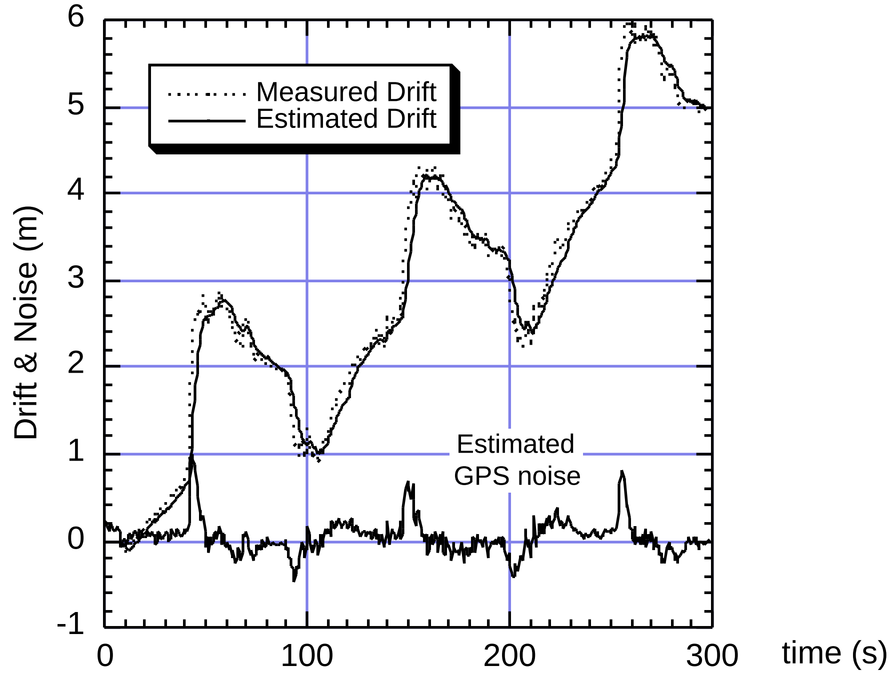

Figure 5. The Kalman Filter computes estimated drift and noise in the Northing direction as the vehicle was driven around the square course shown in Figure 2.

3. RETROTRAVERSE

During retrotraverse the mobility controller drives the vehicle, retracing a previously recorded path. Vehicle position and velocity data provided by the navigation system are used to compute steering, brake, and throttle commands.

The taught path is represented as a series of line segments defined by a set of x-y positions, or knot points, that denote the start and end of each path segment. As the scout drives the vehicle during teaching, the navigation data is monitored and a new knot point is recorded every meter. The vehicle's speed is recorded at each knot point and is used to control the speed during playback. The driver indicates a turn point to either the right or left of the path when the vehicle is at an area that is clear of obstacles. When the final position is reached the driver passes control of the vehicle to the remote operator and leaves.

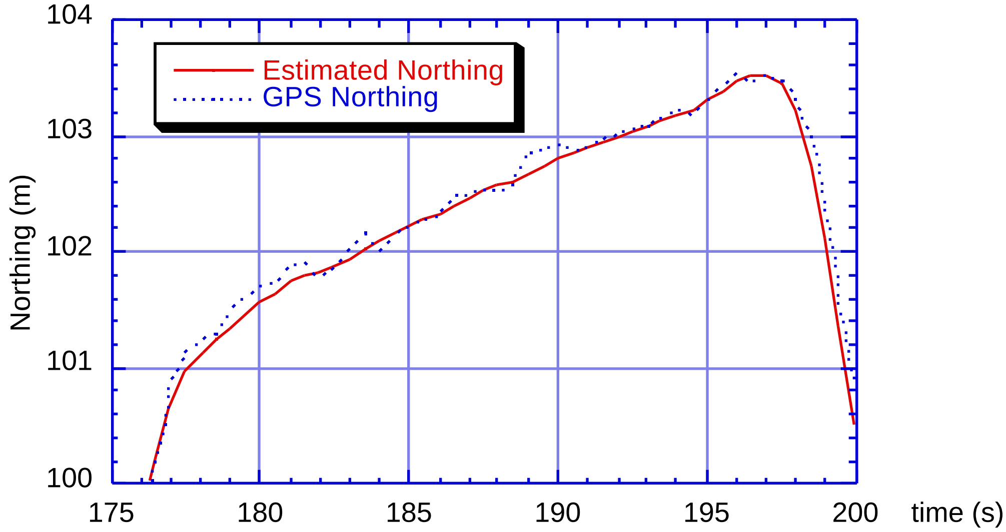

Figure 6. The Complimentary Filter uses the drift estimate from the Kalman filter to compute the estimated position in the Northing direction as the vehicle is driven once around the top of the square course shown in Figure 2. GPS measurements are shown for comparison.

After the vehicle has completed its ATA mission at the current observation point the remote operator commands a retrotraverse. The mobility controller then starts the engine, shifts to reverse, and begins backing up along the path using pure pursuit steering and velocity control, both described below. As the first turn point is approached the steering wheel is turned, either right or left, steering the vehicle into the clear area. After the vehicle turns 120°. it stops and shifts into forward. It then steers back onto the path and continues following the path using pure pursuit steering. A similar procedure is followed at the final turn point, allowing the vehicle to back into the final position. Obstacles such as another vehicle can be detected with a laser range imaging device, described in Section 3.3.

3.1 Pure Pursuit

Pure pursuit steering, as proposed by Amidi [4] and [5], is a very simple and stable steering algorithm that works well with piecewise linear paths. This algorithm steers towards a goal point on the path a set distance ahead of the vehicle. The mobility controller selects a steering radius that will cause the vehicle to intersect the path at this point. See Figure 7. The goal point, turn radius, and steering angle are repeatedly updated as the vehicle moves.

Calculating the turn radius is straight forward. Compute the position of the goal point in the vehicle frame, (x_g, y_g). Then, noting that the center of the turn circle is (0, r), one obtains

$$(x_g - 0)^2 + (y_g - r)^2 = r^2 \qquad (4)$$

Solving for r and taking the reciprocal yields the steering curvature for pure pursuit, γ_{pp}

$$\frac{1}{r} = \gamma_{pp} = \frac{2y_g}{x_g^2 + y_g^2} \qquad (5)$$

The look ahead distance is selected according to vehicle speed. If the look ahead distance is too large the vehicle will cut corners on tight turns. If the distance is too small the controller will be unstable at higher speeds. The robotic vehicle uses 6 meters for speeds up to 30 km/h and 12 m for speeds up to 80 km/h. Typically this results in lateral errors of less than 0.1 m during path following.

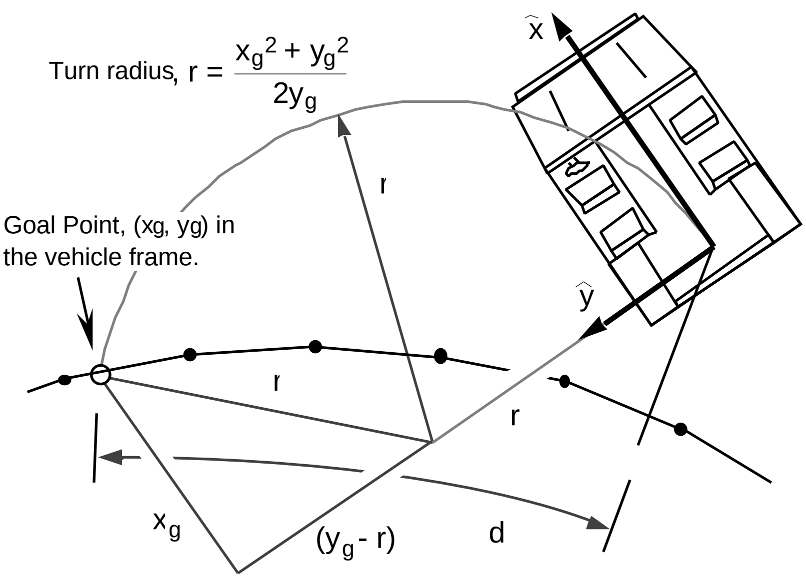

Figure 7. Pure Pursuit Steering. The goal point is a point on the path a specified distance, d, ahead of the vehicle. The turn radius, and hence the steering angle, is selected to drive the vehicle over the goal point. The goal point, turn radius, and steering angle are repeatedly updated as the vehicle moves.

3.2 Velocity Control

Brake and throttle positions are controlled to maintain the recorded velocity using a gain scheduling algorithm. Throttle position, T, is set to

$$T = T_o + K_v (v - v_{cmd}) + K_p \text{ Pitch} \qquad (6)$$

where

T_o, K_v, and K_p are the control gains (which are a function of commanded velocity),
v is the vehicle velocity,
v_{cmd} is the commanded velocity, and
Pitch is the vehicle pitch, positive being up hill.

The control gains are set depending on commanded speed. A table of values T_o, K_v, and K_p for speeds of 5, 10, 20, 30, 40, 60, and 100 km/h is empirically determined. The values are interpolated for intermediate speeds. T_o should provide just the right amount of throttle to produce the commanded velocity on level ground. K_p should be set to increase and decrease throttle to maintain speed as the vehicle drives up and down hills. K_v is selected to provide sufficient feedback to handle errors in the other control variables and variations in terrain roughness. This algorithm provides robust speed control allowing the same table of control gains to be used on four actuated vehicles with varying payloads.

When the computed throttle position given in (6) becomes negative, the throttle is set to zero and the brakes are applied in an amount proportional to (6). A small dead band at the zero crossing is used to prevent the throttle and brake from oscillating back and forth.

A stall condition can occur at slow speeds in rough terrain. For example, when the rear wheels are in contact with a large rock an increase in throttle must be applied. Unfortunately, as the rear wheels begin to climb over the obstacle the rear of the vehicle lifts up and the downward pitch is interpreted by (6) to be a down sloping hill and the throttle is reduced, preventing the vehicle from climbing over the obstacle. To counter this, when the speed drops to zero, the controller slowly increases the throttle until the vehicle moves forward, at which time the throttle reverts back to (6).

3.3 Obstacle Avoidance

Since the battle field is a dynamic environment, the vehicle cannot trust that the path recorded during the teaching will remain free of obstacles. Troop and equipment movements may result in a previously clear path being blocked when retrotraverse is performed. Thus, the vehicle must be able to detect obstacles in its path to drive autonomously.

NIST is currently integrating a Ladar, a laser range imaging device, into the mobility system in order to detect and respond to obstacles that were not on the path during teaching. A Ladar scans the terrain in front of the vehicle producing a grid of range measurements. Using this range image the mobility controller can detect, and determine the position of, obstacles in the path of the vehicle.

The vehicle uses a rugged commercial Ladar that produces a 128 by 64 pixel range image once per second. The sensor has a maximum range of 50 m and a measurement resolution of 6 cm. The data produced by the Ladar compares favorably with data produced by stereographic methods, both in noise immunity and object definition. The sensor is unaffected by ambient lighting conditions and works equally well in day or night operations.

One obstacle detection algorithm being tested, but not yet integrated with retrotraverse, is SMARTY, a program developed at CMU [6]. It generates a local map of obstacles from the Ladar range image and calculates clear paths in front of the vehicle. Using this information the mobility controller can attempt to plan a path around the obstacle. If it fails, it will alert the operator who will teleoperate around the obstacle back to the retrotraverse path.

4. SUMMARY

A number of Laboratories have contributed to the development of a robotic vehicle to try out various capabilities in a realistic battlefield environment. This vehicle has participated in a number of military training exercises so far, allowing valuable interaction between system developers and the end users of the technology.

One of the capabilities implemented on the vehicle was retrotraverse, which enables the vehicle to autonomously retrace a previously traveled path. Retrotraverse is implemented in two phases: the teach phase when a path is recorded, and a playback phase when the recorded path is retraced. During the playback phase the vehicle's steering and velocity are controlled by the mobility system. Steering control is performed using the pure pursuit algorithm which generates a steering command based on the current position of the vehicle and its relationship to the recorded path.

The velocity is controlled using a gain scheduling algorithm.

The vehicle position data used by retrotraverse is generated by combining the outputs of two different navigation systems: the MAPS/NIU inertial dead reckoning system, and a differential GPS. The position data provided by these systems have different update rates, positional accuracies and error characteristics. By combining information from both systems using Kalman filtering techniques it was possible to produce a navigation system that combines the best characteristics of both. This system produces position updates with the high update rate and low noise of the inertial dead reckoning system, with the low drift characteristics of differential GPS.

To cope with obstacles in the path of the vehicle an effort is underway to integrate a laser range imaging sensor into the vehicle's mobility system. By processing the range image produced by this sensor it is possible to locate obstacles in the path of the vehicle. When an obstacle is detected the mobility system can plan a path around the obstacle or notify the remote operator.

5. REFERENCES

[1] F. Fisher, S. Balakirsky, M. Coleman, P. David, P. Gaylord, G. Haas, K. Murphy, S. Legowik, and L. Sadler., "Project Mustang, a UGV for RSTA applications", to be published in Battlefield Systems International 96, Chertsey, Surrey, UK, June 1996.

[2] S. Balakirsky, P. David, P. Emmerman, F. Fisher, and P. Gaylord. "Semi-Autonomous Mobile Target Engagement System", Proceedings: 20th Association for Unmanned Vehicle Systems. pg. 927-46. June 1993.

[3] S. Szabo, H. Scott, K. Murphy, S. Legowik, R. Bostelman, "High-Level Mobility Controller for a Remotely Operated Unmanned Land Vehicle", Journal of Intelligent and Robotic Systems, Vol 5: 63-77,1992.

[4] O. Amidi, Integrated Mobile Robot Control, M.S Thesis, Carnegie Mellon University, May, 1990.

[5] K. Murphy, "Analysis of Robotic Vehicle Steering and Controller Delay." ISRAM '94, M. Jamshidi, C. Nguyen, R. Lumia, J. Yuh, Volume 5, Robotics and Manufacturing, pg 631, Wailea, Aug, 1994.

[6] M. Hebert, "Pixel-Based Range Image Processing", Proc. IEEE International Conference on Robotics and Automation, San Diego, 1994.