# An Autonomous System Used in the DARPA Grand Challenge

Joseph C. Bebel, Nathan Howard, and Tej Patel

*Abstract* – **An autonomous system was designed and constructed to traverse the hostile desert terrain in the DARPA Grand Challenge robotics competition. Our autonomous system used GPS measurements in conjunction with a laser range finder to determine passable terrain and to negotiate around natural and man-made obstacles on the route. The primary navigation portion of this system uses a scalable, heuristic array of data, and determines the preferred heading of the vehicle, factoring desired heading from a GPS sensor and obstacle data from a laser range finder. When implemented, this system demonstrated capability to pass stationary vehicles and navigate sloping terrain while negotiating around fences and towers between sequential GPS waypoints.**

## I. Introduction

The Grand Challenge, orchestrated by DARPA, was a competition to design and build an autonomous vehicle that could drive 150 miles over rough, off-road terrain. We designed a system that avoided obstacles while proceeding to each GPS waypoint to compete in this challenge.

The basic approach taken for the autonomous vehicle was to construct a basic, simple autonomous system that could be enhanced as time permitted. A rapid prototyping approach was used, and ideas taken from extreme programming [1] were employed to insure that the code developed was robust and reliable. Although there was a time constraint regarding completion of the challenge course, the philosophy taken was to begin with a system that operated at a modest, fixed velocity, adding increased velocity in open spaces as time permitted.

The vehicle used was an Acura MDX, enhanced with the addition of Electronic Mobility Controls. These controls were originally designed to assist handicapped drivers. An analog interface was provided to the team to control steering, braking and acceleration.

Our system sensors comprised a laser range finder and a GPS unit. The laser was used to detect obstacles while the GPS unit found our position on earth. These two inputs were then put into an algorithm to determine the best heading for the car, one that went to the next GPS waypoint while avoiding impediments.

The algorithm we produced was simple and quick to integrate. The algorithm used a scalable data array, allowing for easy integration of new sensors, without having to modify large amounts of code. The algorithm was also extremely fast, the logic loop ran 4 times a second, pausing most of the interval to allow the car actuators to take effect.

Time constraints on the project forced us to implement the algorithm quickly. The algorithm had to be able to avoid desert obstacles while not deviating significantly from a set course of GPS waypoints. The algorithm also needed to be fast using minimal processing, to be suitable for embedded environments.

GPS waypoints, consisting of consecutive latitude/longitude pairs, are inputted into the system via CDROM. The system then navigates to each waypoint, until it comes within a specified radius of the waypoint. During navigation, the vehicle must stay within specified GPS boundaries as it progresses to each waypoint. Then the vehicle begins navigating to the next waypoint in the list.

For obstacle detection, the vehicle has a horizontally mounted, scanning laser rangefinder (the SICK LMS-291) mounted low on the vehicle front bumper and level so it will scan parallel to the ground. The unit scans horizontally in a plane, giving a distance measurement every 0.5 degrees in its 180 degree scan, from which we can determine the range to any impediment, and thus avoid that obstacle. This allowed us to continually refresh the inputs so no memory or tracking of objects is required; any obstacles are shown on every scan.

For processing hardware, we constructed a standard, x86-based Linux system, with an off-the shelf value of approximately 300 USD. This provided more than enough computation power to execute the laser and GPS navigation loop in a matter of milliseconds. Measurement Computing digital to analog converters were used to create the voltage levels required by the Electronic Mobility Controls.

For GPS inputs, we used a standard commercial WAAS unit (the Garmin GPS16A) capable of determining absolute position, velocity, and altitude with 3 meters of accuracy.

## II. Related Work

Much research has been performed on the topic of automated highways (see for example, [2]). Fewer publications on autonomous cross-country vehicle operation exist, but it is impossible to cite all related publications here. A few publications with similar approach will be cited.

In his MS thesis [3], Hellström describes obstacle avoidance for autonomous forest vehicles using Vector

Manuscript received May 1, 2004.Authors are listed in alphabetical order. Joseph Bebel is currently a Sophomore at Palos Verdes High School, Palos Verdes, Ca.

Nathan Howard is currently a Junior at Palos Verdes High School, Palos Verdes, Ca.

Tej Patel is currently a Sophomore at Palos Verdes High School, Palos Verdes, Ca.

Field Histograms [4], a technique based on the Virtual Force Field method [5]. This method is similar to the method we use, although no sensor fusion is described, and the algorithms were tested only in a MATLAB simulator. A comprehensive system for autonomous vehicle navigation was described in a Carnegie Mellon Univ. technical report [6]. Obstacle detection in a flat plane, segmented into intervals, was similar to our approach, but did not provide for direct fusion with GPS and was superceded in that work with 3-D detection. Another publication [7] also describes 2-D laser range finding without sensor fusion.

The University of Cincinnati's BEARCAT III autonomous vehicle served as a test bed for three obstacle avoidance systems, two sonar systems and one laser rangefinder system [8]. The approach using the laser rangefinder was similar to our approach and was similar to the system we implemented. Wall and colleagues [9] described a low-cost autonomous vehicle platform that could host the algorithms described in this paper.

## III. Description Of The Autonomous Software Operation

A field of **trajectories** is created as an array of all possible directions the vehicle could proceed. **Viabilities** (ratings based on desirability of each path according to a sensor) are given to each trajectory in values from 0 to 1, where 0 is the worst and 1 is the best. The distance between trajectories is the smallest unit that the sensor with the least resolution can measure. In our case, the GPS viabilities have almost infinite resolution through interpolation. Therefore, the resolution of the laser determined the spacing of the trajectories; the laser measured every 0.5 degrees, so one trajectory was placed every 0.5 degrees.

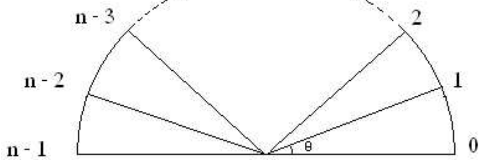

**Figure 1:** A sample array of *n* trajectories with a spacing of θ between each trajectory

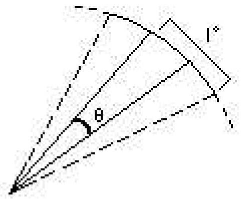

**Figure 2:** An example set of trajectories with spacing θ = 0.5°

Each trajectory is given a set of viabilities, one for each sensor based on how appropriate that trajectory is, according to the parameters of that sensor. If the laser detects an obstacle 2 meters away spanning a range of trajectories, then that particular range of trajectories will be given very low laser viabilities. However, detection of an obstacle 70 meters away would receive higher laser viabilities.

At the beginning of each processing loop, we determine the heading from the vehicle's current position to the target GPS waypoint.

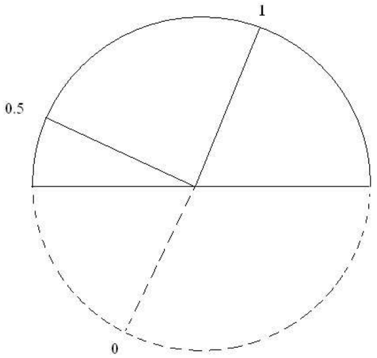

**Figure 3:** GPS viabilities vary from 1, toward the target, to 0.5 at a right angle from the target, and to 0 at 180 degrees away from the target

The GPS viabilities (a numerical rating) of a trajectory begin at 1, the most desirable trajectory, then decrease proportionally, to 0 at 180 degrees from the best trajectory.

In (1), $\gamma$ is the viability, $i$ is the trajectory number, $\theta$ is the spacing in degrees between trajectories, and $\beta$ is the desired trajectory according to GPS.

$$\gamma_{gps} = 1 - \frac{\theta\,|i-\beta|}{180} \quad (1)$$

For simplicity, we want each laser measurement to be rounded down to the greatest multiple of 5m below the measurement. This is done by dividing by the distance of the step (in our case by 5 m) and rounding the resulting number down to the greatest integer, then dividing by the number of steps to normalize the viabilities between 0 and 1, as shown in Equation 2. This also helps avoid problems in situations where features on an obstacle confuse the sensors.

For each laser measurement, we calculate the viability $\gamma$ using Equation (2) by rounding down to the greatest integer the product of the ratio of the measurement $\rho_\alpha$ to the maximum measurement $\rho_{max}$ and $\xi$ (the step we want).

$$\gamma_{laser} = \frac{1}{\xi}\left[\rho_\alpha \frac{\xi}{\rho_{max}}\right] \quad (2)$$

However, each single trajectory is often not sufficiently wide to accommodate the span of the vehicle. Therefore, the software examines nearby trajectories to determine the final laser viability of a trajectory. This can be accomplished by setting each final laser viability to the lowest viability of several trajectories to the left and right.

$$\gamma_\alpha = \frac{1}{2}\left(\kappa_{gps}\gamma_{gps} + \kappa_{laser}\gamma_{laser}\right) \quad (3)$$

Each trajectory alpha can be assigned a 'total' viability compiled from the two sensor viabilities multiplied by respective weights $\kappa_{gps}$ and $\kappa_{laser}$. These weights allow the algorithm to be adjusted to give more influence to particular sensors. For example, if the vehicle tends to wander away from the given path, more weight on the GPS viability may be necessary. If the vehicle tends to hit obstacles, more laser weight may be needed.

If more than two sensors are providing inputs to the autonomous system, the general formula is:

$$\gamma_\alpha = \frac{1}{N}\sum_{n=1}^{N} \kappa_n \gamma_n \quad (4)$$

The weights are not required to be constant. Based on factors such as terrain or vehicle velocity, the weights may be changed. For example, if there are no obstacles directly ahead of the vehicle closer than 60 m. and the vehicle is already heading toward the GPS waypoint, the GPS weight is increased in order to prevent the vehicle from turning away from the waypoint because of high laser viabilities to the sides of the waypoint. Once the total viability of all trajectories has been computed, the trajectory with the highest total viability is the desired trajectory.

Since our vehicle did not move at more than 18.5 km/h (10 knots) autonomously at any time, a closed-loop steering controller was not necessary. Because the vehicle does not cover much ground before the system restarts the control loop, effects similar to a closed-loop are obtained as the software reevaluates the desired pathway and adjusts steering as required every 0.25 sec. However, in a higher speed vehicle, the autonomous system would finish the loop after much more ground had been covered, so a much finer-grained steering control would be required.

## IV. Results

In repeated tests in a parking lot empty of other vehicles, the vehicle successfully avoided 4' high obstacles (garbage cans) while navigating between GPS waypoints. With careful determination of the proper viability weights and calibration, success rates increased. At the official qualification rounds for the challenge we encountered an issue with steering adjustments on grass and dirt. However, after proper adjustments, the vehicle proceeded past eight of the fourteen obstacles on the course, successfully navigating over a hill, past a tower and around a vehicle situated immediately over the GPS waypoint. The car was seeded 10th of fifteen vehicles for the challenge itself. The day before the challenge, our car was successfully able to proceed through the first 25 waypoints of the race. Unfortunately, on the day of the challenge our vehicle experienced some electrical problems while leaving the starting gate, and collided with a concrete barrier, which officially removed us from the challenge. Ultimately, the car performed as well as or better than a number of the fifteen vehicles. It is difficult it compare our results with published literature, as we attempted to solve the specific problem posed by the DARPA Grand Challenge, while others have attempted to solve different problems. In comparison with other vehicles in the challenge, our vehicle completed over 50% of the qualifying course, ranking above 5 other vehicles with more complex and expensive systems.

## V. Conclusions

We demonstrated the feasibility of rapidly prototyping and implementing a complete functional autonomous system. The system was capable of navigating through basic desert terrain. The viability array handled distinct obstacles and rapid decision making between conflicting sensor inputs with ease.

Future research on this system should be performed to investigate techniques in adding a quasi-closed loop system for more precise steering at higher speeds. Furthermore, new research should concentrate on integrating multiple sensors (Lasers and vision system) to get a more comprehensive view of what is in front of the vehicle. An Inertial Measurement Unit would provide additional localization information. A second GPS unit could provide heading information without requiring the vehicle to be in motion. Moreover, refinements such as placing a corridor algorithm in the system to increase the GPS weight as you approach a corridor boundary may be effective. Some limitations of the approach should be addressed in future versions. Any obstacles above or below the scan line cannot be detected .Additionally, passable slopes appear as obstacles.

In addition, the use of a floating point processor is not essential. With slight modifications, a floating point processor is not required.

## VI. Acknowledgements

The authors would like to acknowledge the sponsors that made the autonomous system described here possible, including American Honda, Electronic Mobility Controls, Univ. of Southern California, Measurement Computing, SICK North America, the Palos Verdes High School Booster Club, and many parent and teacher contributors. Daniel Hirsch provided critically needed systems programming. USC Professors Alice Parker, Petros Ioannou, Andrew Howard, Maja Mataric and Gaurav Sukhatme, several USC graduate students, and Dr. Donald Bebel provided engineering advice. Many Palos Verdes High School students provided valuable advice and assistance in preparing and testing the vehicle.

## VII. References

[1] Kent Beck, *Extreme Programming Explained: Embrace Change,* Addison-Wesley Pub Co; 1st edition, 1999.

[5] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation, *Proceedings of the IEEE Int. Conf. On Robotics and Automation*,”1991.

[4]J. Borenstein and Y. Koren, “The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots,” *IEEE Transactions on Robotics and Automation*, 7(3): 278-288, 1991.

[3] T. Hellström, “Path Tracking and Obstacle Avoidance Algorithms for Autonomous Forest Machines,” Masters Thesis, Umeå University, Sweden, April, 2003.

[6] S. Singh, et al., “A System for Fast Navigation of Autonomous Vehicles,” Carnegie Mellon University Robotics Institute Technical Report CMU-RI-TR-91-20, Sept. 1991.

[2] P. Ioannou, editor, *Automated Highway Systems*, New York City: Plenum Press, 1997.

[7] J. Gonzalez, A, Ollero and A. Reina, “Map building for a mobile robot equipped with a 2D laser rangefinder,” *Proceedings, 1994 IEEE International Conference on Robotics and Automation,* vol.3, May 1994, San Diego, CA, pp. 1904 – 1909.

[8] S. Modi, “Comparison of three obstacle avoidance methods for an autonomous guided vehicle,” Master's thesis, Department of Mechanical, Industrial, and Nuclear Engineering, Univ. of Cincinnati, 2002.

[9] R.W. Wall, J. Bennett, G. Eis, K. Lichy, E. Owings, “Creating a low-cost autonomous vehicle,” IECON02, Nov. 2002. pp 3112-3116.