The Overbot: An Off-Road Autonomous Ground Vehicle Testbed

Gabriel Hugh Elkaim[†], John Connors[†] and John Nagle[‡]
[†]Autonomous Systems Lab, Computer Engineering, University of California at Santa Cruz
[‡]Team Leader, Team Overbot, Redwood City CA

BIOGRAPHY

Gabriel Elkaim received his B.S. degree in Mechanical/Aerospace Engineering from Princeton University, Princeton NJ, in 1990, and both M.S. and Ph.D. degrees from Stanford University, Stanford CA, in Aeronautics and Astronautics, in 1995 and 2002 respectively. In 2003, he joined the faculty of the Computer Engineering department, in the Jack Baskin School of Engineering, at the University of California, Santa Cruz, Santa Cruz CA, as an Assistant Professor. His research interests include control systems, sensor fusion, GPS, system identification, and autonomous vehicle systems. His research focuses on intelligent autonomous vehicles, with an emphasis on robust guidance, navigation, and control strategies. Specifically, he has founded the Autonomous Systems Lab at UC Santa Cruz, and is currently developing an autonomous wing-sailed marine surface vehicle and off-road autonomous ground vehicles.

John Connors completed B.A. degrees in Computer Science and Mathematics at the University of California at Berkeley, Berkeley CA, in 2004. He worked as a PCB engineer and consultant designing audio/video switching devices before starting as a graduate student in 2005 in the Jack Baskin School of Engineering, at the University of California, Santa Cruz, Santa Cruz CA, working towards a M.S. in Computer Engineering. His research interests include electric vehicles, robotics and autonomous control. His current project centers around outfitting an autonomous offroad ground vehicle as a reconfigurable autonomous testbed for comparative analysis of various control techniques and sensor suites.

John Nagle headed Team Overbot, one of the DARPA Grand Challenge off-road robot vehicle teams, based in Redwood City, CA. He currently holds patents covering technologies for legged locomotion and dynamic simulation. He is an industry leader and is the developer of animation methods for free falling bodies and rag doll physics used by many top animation studios throughout the country.

ABSTRACT

The Overbot, originally designed to run the DARPA Grand Challenge, has been retasked as an off-road autonomous vehicle testbed. The base chassis is a Polaris Ranger 6x6 gasoline powered off-road vehicle. It has a single cylinder 30 HP engine capable of propelling the vehicle to over 40 mph. In addition to the chassis, the Overbot is equipped with a SiCK LiDAR on a custom gimbaled mount, a color firewire camera in a waterproof housing, a Crossbow AHRS, a Novatel differential GPS system and several modified 802.11.b wireless hubs. In terms of actuation, there are five Galil Motion Controllers which drive servomotors that handle shifting (H-L-N-R), braking, throttle, steering and the vision angle. Software which performs obstacle detection, path planning and low-level control is implemented in a Real Time Operating System (RTOS) on a standard Pentium class PC. The original incarnation of the Overbot was selected to run in the National Qualifying Events (NQE) of the Grand Challenge in both 2004 and 2005, but failed to proceed to actual competition. The Overbot is a very agile off-road platform, capable of passing over obstacles as high as 15 inches. Demonstration of new path following control shows crosstrack errors that are less than 1 meter and a new heading correction implementation is discussed.

BACKGROUND

DARPA GRAND CHALLENGE

The Grand Challenge (GC) is a Department of Defense (DoD) sponsored competition designed to further the research of autonomous ground vehicles to help reduce the risk to troops in the battlefield. Run by the Defense Advanced Research Projects Agency (DARPA), the results of the first GC competition in 2004 were less impressive than many had hoped. The desert course proved to be more difficult than many imagined, and the vehicles far less robust than desired. It was clear that a successful vehicle not only needed to be able to traverse difficult terrain, but needed to be keenly aware of its desert environment as well. With the experience gained from each failure, the 132 mile course laid out for the 2005 competition was successfully completed by no fewer than 5 vehicles. Currently DARPA plans to hold a new GC in 2007 in an urban setting. Vehicles will be required to obey traffic laws as well as navigate through traffic over the 60 mile course.

TEAM OVERBOT

Silicon Valley has long been a driving force in the advancement of new technologies. When DARPA released its plans for the Grand Challenge (GC), it seemed logical that Silicon Valley could answer that call. Founded by John Nagle, Team Overbot consisted of volunteers from throughout the robotics, imaging and automotive sectors. Building around a Polaris 6x6 ATV, the privately founded Team Overbot entered the National Qualifying Events for both the 2004 and 2005 GC competitions. Various difficulties

Fig. 1. Polaris Ranger 6x6

and failures kept Team Overbot from competing in the final GC events[13].

HARDWARE OVERVIEW

POLARIS RANGER 6x6

The Polaris Ranger, shown in Figure 1, is a commercially available utility vehicle designed with off-road capabilities in mind. With a 3/4 ton towing capacity, the ability to drive through water up to 27 in. deep and over 6 inches of suspension travel front and rear, the Ranger is aptly suited for traversing mountainous and rocky terrain. The 30 HP engine allows a top speed of around 40 mph and offers enough torque to overcome large obstacles. The automatic variable transmission offers high, low and 6WD modes. The 4 wheel hydraulic brakes are suitably powerful to stop the 1400 lbs. vehicle[11].

SICK LIDAR

The SiCK LiDAR unit is a highly accurate laser rangefinder. The LiDAR is an active device meaning it does not require any external illumination. Using a rotating mirror, the LiDAR emits an infrared beam in a 180° viewing angle. As the beam reflects back to a sensor, the range can be calculated for each point in the beam's rotation[9].

On the Overbot, the LiDAR unit has been mounted above the vehicle on a custom gimbaled mount. This setup allows the LiDAR to project downward onto the ground, and thus provide a contour of the road ahead, as demonstrated in Figure 2. The mount itself is driven by a servo motor and can be pitched relative to the vehicle itself. In this manner, the focal point of the LiDAR can be adjusted further or closer to the vehicle to ensure adequate time to process the data as the speed of the vehicle changes.

UNIBRAIN FIRE-I 400 CAMERA

To aid in detecting obstacles and road boundaries, the Overbot incorporates a Unibrain Fire-I 400 color camera. The Fire-I is a CCD based camera capable of 659(H) x 494(V) pixels. The unit is highly sensitive and resists blooming. The FireWire interface offers up to 400 Mbps bandwidth. The camera is mounted atop the vehicle near the SiCK LiDAR unit for a similar vantage point down onto the road[10].

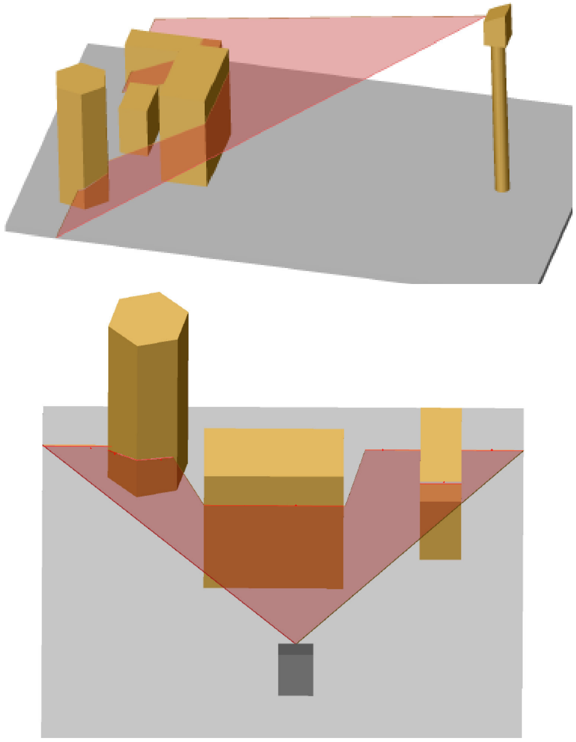

Fig. 2. SiCK LiDAR Profile of Terrain

Specification	Value
Angle Range	200 deg
Angular Bias	±0.005 deg/sec
Acceleration Range	10 g
Acceleration Bias	±12 mg
Roll, Pitch Dynamic Accuracy	±2.5 deg
Yaw Dynamic Accuracy	±4 deg

TABLE I
SPECIFICATIONS FOR THE CROSSBOW AHRS 400CB-200

CROSSBOW AHRS

The Crossbow AHRS 400CB-200 is an attitude and heading reference system, estimating roll, pitch and yaw using a 3-axis accelerometer, rate gyros, and a three axis magnetometer. By fusing these measurements the unit is able to estimate roll, pitch and yaw at high bandwidth. Together, these measurements can be used to determine the orientation and dynamics of the vehicle in motion. The AHRS 400CB-700 is accurate to ±2.5° for pitch and roll, and ±4° for yaw[7].

NOVATEL DIFFERENTIAL GPS

For a vehicle to autonomously maneuver itself to a new location, it must know its current location along the way. The Overbot uses a commercial Global Positioning System (GPS) device from Novatel for this purpose. The Novatel ProPak-LB features a 24 channel ”all in view” GPS receiver. When coupled with the OmniSTAR L-Band service, differential corrections can applied to reduce typical errors to less than 0.10 meters[8].

GALIL MOTION CONTROLLERS

The DMC-1400 series Motion Controllers from Galil Motion Control inc. are designed to offer convenient control of brushed, brushless and servo motors. With digital and analog inputs, and digital outputs, the Galil devices are some of the most versatile controllers available. The seamless integration of motors, encoders, limit switches and user defined error events makes them easily integrated into any project[5]. On the Overbot chassis, the Galil controllers are used to control the steering, throttle, brakes and shifting.

The Polaris Ranger has hydraulic disc brakes on four of its six wheels. These are driven by a pedal and a dual master cylinder, so that there are two brake hydraulic systems, front and back. The actuator is a screw drive device driven by a brush-type servomotor. The sensors include two limit switches on the actuator, and a pressure switch and pressure sensor on the front hydraulic system. All of these devices are connected to a Galil DMC-1416 mounted under the hood.

The stock Ranger has manual steering through a steering box on the front axle with an upward-pointing lower steering shaft mounted to a universal joint at each end. The steering wheel is attached to the upper end of this shaft. The lower steering shaft has been replaced with a shorter shaft, which connects to a large Faulhaber servomotor through a shock coupling. Below the shock coupling is a BEI rotation sensor. The actuator for the steering system is a planetary gear head driven by a brush-type servomotor and encoder. The BEI rotation sensor is a multiturn potentiometer/encoder combination. These devices are connected to a Galil DMC-1416 and Galil DMC-1460 interface box with optoisolation, mounted under the hood. The Galil DMC-1460 is also used to operate solid-state relays for the ignition, starter, and 6WD unit.

The actual throttle actuator is a modified cruise control unit. It has control over the engine, including engine start and run, and has many sensors attached. The cruise control motor has been replaced by a small servomotor. There is a reverse limit switch at throttle idle, and a forward limit switch at full throttle. There's also a safety device, an electromagnet, which must be energized for the throttle to move from idle.

INTERCONNECT

The Overbot consists of a series of subsystems, such as a GPS unit, motion controllers, vision devices and computers. Each system is connected together through an Ethernet network and a series of routers. Communication is handled by transmitting UDP packets from device to device. Each system is given a unique, globally known IP address (Table II). Some devices, like the Galil controllers, are already Ethernet enabled. Others, such as the LiDAR and GPS unit, have USB, Firewire or serial ports, and are adapted to communicate on the network.

IP Address	Device ID	Function
10.100.100.132	gcrear0	QNX Computer
10.100.100.134	gcrear2	Vision Computer
Galil controllers		
10.100.100.137	gcbrake	Brake Control
10.100.100.138	gcsteer	Steering Control
10.100.100.139	gcthrottle	Throttle control
10.100.100.140	gctransmission	Transmission Control
10.100.100.141	gctilt	LiDAR Gimbal

TABLE II
SUBSYSTEM IP ADDRESS ASSIGNMENT FOR OVERBOT VEHICLE

SOFTWARE DESIGN

GPSINS SERVER

The highest level of the software hierarchy is the GPSINS Server. This software communicates with the Novatel GPS receiver to determine the vehicles coordinates. With the addition of the Omnistar High Performance subscription, the position readings are accurate to decimeter levels. These measurements are available at a rate of 20 Hz.

The GPSINS Server also interfaces to the Crossbow AHRS 400CB-200 roll, pitch, yaw and angular rate estimator. The Crossbow unit internally calculates Kalman filtered angles at 60hz. The tilt measurements are passed to the Move Server to estimate vehicle dynamics.

When the Novatel GPS receiver has enough satellites in view and is receiving the GPS and C-band differential signals, the vehicle has decimeter position accuracy. When signal is lost, the Overbot uses a dead reckoning filter based on AHRS heading and both odometry and doppler radar based velocity measurements to track its position[2][3]. While a traditional INS approach would be to integrate the acceleration readings from the Crossbow AHRS device to yield velocity changes and then integrating a second time to determine position changes, the included sensors are far too noisy to produce a useful position estimate. Instead, the GPSINS Server uses a dead reckoning filter to produce position estimates at 60 Hz. Due to drift in heading and numerical errors, these measurements can only be trusted for about 10 minutes[12].

STEER SERVER

The Steer Server acts as the path planning control for the vehicle. Using the current location of the vehicle the Steer Server plots a path to the desired destination. Obstacle information is included and trajectories are planned to avoid these obstacles as well as the imposed corridor constraints. Using this path, a turn radius and speed are calculated and sent to the Move Server as commands.

Details of the underlying algorithms are discussed later in this paper.

MOVE SERVER

The majority of the vehicle dynamics and error checking occur inside the Move Server. Desired *distance*, *max speed* and *curvature* are passed in from the Steer Server. In general, the Move Server attempts to keep the vehicle moving as fast as possible while obeying the *max speed* limitation.

Within this limit, other dynamics may effect the target speed. Limits such as pitch and acceleration rates are enforced through simple vehicle dynamics. These measurements come from the accelerometer included in the Crossbow AHRS unit. The speed is also limited by the needed turning radius to ensure vehicle safety. Calculations of actuator performance, based on previous behavior, are used to make further adjustments. The Move Server decides on ideal speed, gearing and curvature before issuing commands to the Speed and Direction Servers.

SPEED SERVER

The main purpose of the Speed Server is to interface with the hardware motion controllers. The Speed Server has control over the throttle, brakes and gear selection. The state of the vehicle is maintained and commands are accepted from the Move Server to change the speed, acceleration, gear and state. The server replies with the actual values for each parameter.

The server maintains state in order to restart the vehicle in the event that it should fail. Because the Ranger transmission needs to be shifted when stopped, the Speed Server manages the stopping of the vehicle before changing gears. The decision to change gears is made in the Move Server, although the Speed Server will offer suggestions when the engine RPM and vehicle speed indicate the need. By comparing odometer readings to data from a Dickey-John Radar Velocity Sensor, the Speed Server is also able to detect when the vehicle wheels begin to slip.

DIRECTION SERVER

The steering counterpart to the Speed Server is the Direction Server. The software controls the lower level hardware controllers attached to the Ranger's steering rack. The vehicle's steering box is actuated by a servo motor and the resulting motion is captured through the use of an optical encoder. The Direction Server acts as a translation between the the Move Server commands and the instructions sent to the hardware controller. It also has the ability to track drift or saturation of the steering actuators.

Upon vehicle startup, the Direction Server initializes the steering system and homes the vehicle's wheels to a straight line orientation. During autonomous operation, the software commands various wheel positions as well as angular rates. The Direction Server is also available for other processes to provide accurate feedback of steering angle and behavior.

PATH PLANNING

PLANNING ALGORITHM

The path planning for the Overbot uses a simple goal oriented algorithm. The Move Server evaluates a simple map of the terrain. Corridor constraints, obstacles and unknown cells are marked in the map. Unmapped cells are distinguished from obstacles as these cells can be approached and mapped. The map also contains the location of the vehicle and a goal point.

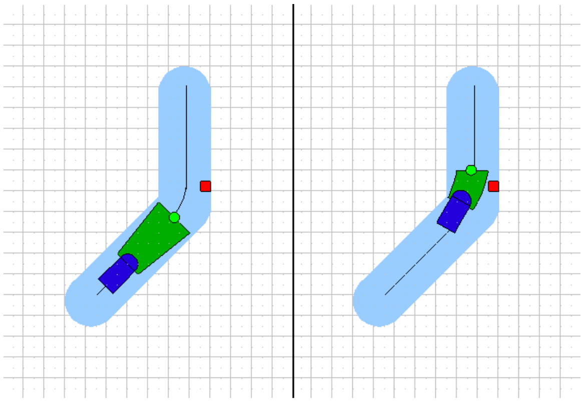

Fig. 3. Curved wedge indicates a possible path.

A wedge is defined starting from the current location having a width the same as the vehicle. The wedge extends in a tangent arc to the goal point, getting wider as it approaches the goal point. The increase in width is designed to compensate for the uncertainty in the future location of the vehicle (Figure 3). The wedge also includes a constraint requiring the vehicle to be able to stop before it reaches the goal point. Therefore, the closer the goal point is to the current location, the slower the vehicle can travel.

In order to be able to traverse a tangent arc, the vehicle must determine that the entire wedge is passable and clear of obstacles. If a suitable wedge can not be found, the goal point is moved closer to the vehicle to limit the final width of the wedge. The goal point can be moved side to side away from the straight line path until a suitable wedge is found.

The majority of the computations are performed in determining whether a particular wedge is clear of obstacles. The original Overbot path planning algorithm begins at the base of the wedge, on the outside curve and works along the curved edge one cell at time. For each cell, the algorithm scans across the cells, perpendicular to the centerline until the inside curve is reached. Starting from the outside curve ensures that each cell is examined at least once.

Moving along a curved arc can be done efficiently with the arc variant of Bresenham's line drawing algorithm and, of course, moving along a straight line requires only the classic version of the algorithm[4]. Bresenham developed this algorithm to reproduce lines in a pixelized environment and remains a popular tool in computer graphics. In this way, the Move Server can compute which map cells will be passed through by the Overbot. By moving from the outside curve inward, all cells in the wedge will be cover by Bresenham's algorithm, and the entire wedge can be evaluated for obstacles.

CLOSING THE LOOP

In its original form, the Overbot did not include any motion feedback. The lower level controls tracked their respective motions but the vehicle motion was inherently open-loop. We have modified the vehicle to perform feed forward path planning with feedback corrections (Figure 4).

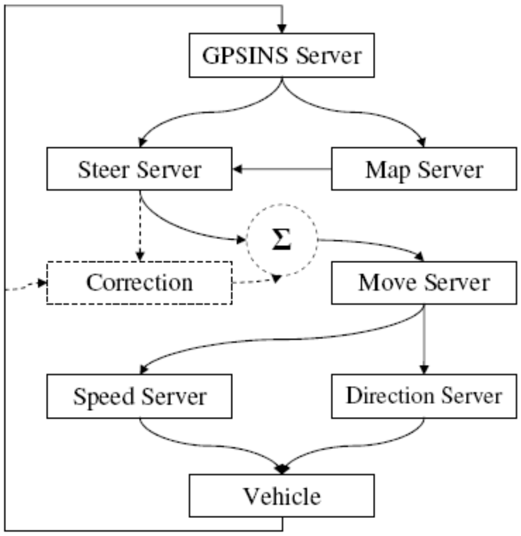

Fig. 4. The Overbot now performs feed forward path planning with feedback corrections.

This setup brings together the accuracy of feedback control without hindering the ability to modify the route around sudden obstacles. We use the same path planner as before, updating the route once per second.

For each 50 ms GPS update, we now compute a path error. A point c is extended in line with the vehicle varying in distance based on the speed of the vehicle. c is projected onto the curve at point p such that c lies on a line normal to the curve at point p. The distance between points c and p is the cross track error y_{err} and Ψ_{err} is the angular difference between vehicle heading and the curve tangent at point p (figure 5). The steering control, therefore, is based on the feed forward term, δ_{ff} and defined as

$$\delta = \delta_{ff} + k_{\Psi} * \Psi_{err} + k_y * y_{err} + k_i * \int y_{err} ds.$$

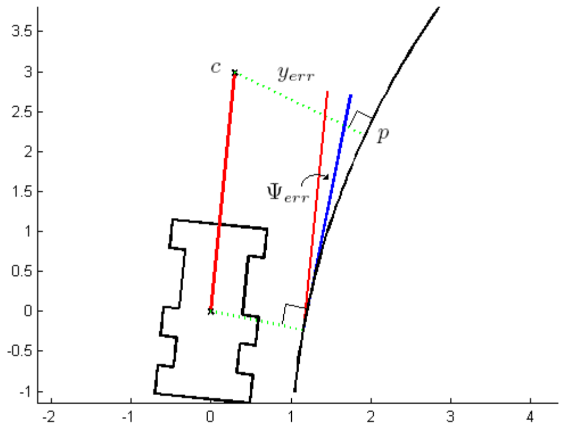

Fig. 5. Error in vehicle position is based on a point c, projected in front of the vehicle, and unto the ideal cure(black) to point p.

PERFORMANCE

TESTING METHODS

The Overbot vehicle is designed to follow a series of way points. Way points are geographic points denoted by their GPS coordinates. A route file is created with a series of these way points and a defined corridor that the vehicle is free to maneuver within. All way points are traversed in fixed order. We define the ideal path to be a straight line route from point to point. The error then is the crosstrack, or length of the line normal to the straight line path and passing through the vehicle position.

Our experiment setup consisted of straight line path across level, flat ground. The way point file consisted of a straight line path, approximately 80 m in length. For each controller, the Overbot was placed at the initial way point with a heading equal to the heading of the path. The Overbot completed the path ten times for each controller tested.

During each test run, GPS coordinates were continuously logged at a rate of 10 Hz. The initial and final segments of each run were truncated to include only the middle 50 m section of the run. This was done in order to exclude the effects of the conversion time of the internal GPS algorithms and any lead-in or transient time associated with the various controllers.

The crosstrack error was calculated as the length of the line normal to the path and passing through the vehicle position.

$$error_t = sin(\Psi_{Path} - \Psi_{WP(t)}) * dist(WP, Pos_t),$$

where Ψ_{Path} is the heading along the path, $\Psi_{WP(t)}$ is the angle between the initial point of the path and the position of the vehicle at time t, WP is the initial point of the path, and Pos_t is the position of the vehicle at time t. These results are analyzed below and presented in Table III.

OPEN LOOP CONTROL

The Overbot, as original designed and built, does not include any high-level feedback in it's path planning algorithms. It implements the planning described in Section (Section). These paths are generated every 100ms based on the current vehicle location, as reported through GPS, and the way point file. As a result, open loop errors and steering biases force the vehicle off track. These issues are indirectly dealt with by the algorithm's priority for paths ending along the ideal path. As the crosstrack error increases, the path curvature will increase and compensate for steering biases and results in a parallel path as demonstrated in Figure 6.

The effects of these errors can be seen in our results, presented in Figure 7. A misalignment in steering causes the vehicle to be offset from its nominal path in one direction. Once significantly off course, the adjustment of the path planning results in a fairly parallel path, as can be seen in the graph. The results indicate an average crosstrack error of 114.5cm and a standard deviation of 25.4cm.

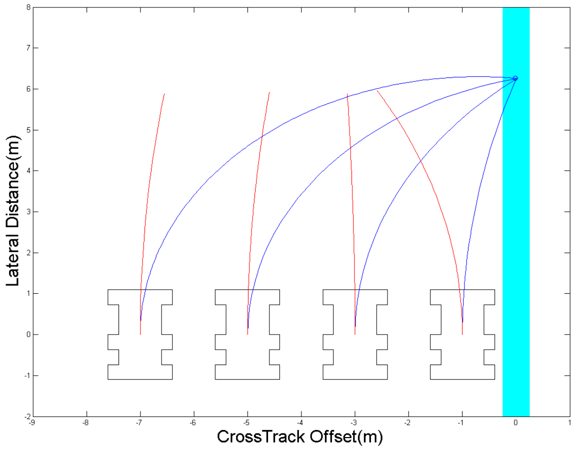

Fig. 6. Because the path planner gives priority to paths ending on the ideal path, larger crosstrack errors can overcome steering biases but result in an offset. The blue lines show the intended paths and the red indicates actual paths given a constant bias.

PROPORTIONAL AND DERIVATIVE CONTROL

The first step we took in improving the performance of the Overbot was to add feedback control in the path planning loop. We slowed the computation of new paths from every 100 ms to every 500 ms. In addition, we computed error measurements every 100 ms. We compute the crosstrack error as the length of the line normal to the curve and passing through the vehicle position, and the heading error as the difference between the vehicle heading and the heading of the curve at the point of intersection with the normal line used in the cross track error. These values are then included proportionally into the steering control along with the original feed forward term to create a control law

$$\delta_{Control} = \delta_{ff} + k_{\Psi} * \Psi_{err} + k_y * y_{err}.$$

The results presented in Figure 7 demonstrate this controllers ability correct for some of the errors introduced by the vehicle dynamics. The control techniques actively track the ideal path and demonstrates a smaller average crosstrack error of 62.4 cm. The error that still exists as a constant offset in the crosstrack is due to a steering bias in the vehicle. This problem can be further diminished by the introduction of integral control. In addition to the reduced offset, the use of the PD controller also reduces the standard deviation to 7.4 cm, approximately two and half times better than the straight open loop control.

PROPORTIONAL, INTEGRAL AND DERIVATIVE CONTROL

After adding proportional and derivative control to our path planning, we found a great increase in accuracy but still had issues with physical steering biases resulting in an offset error. To solve this problem, we included an additional error term in our calculations. An integral error was computed as the area between the actual path and the ideal curve. This term was then included in the control in the same way as the proportional and derivative terms where the control was now

$$\delta = \delta_{ff} + k_{\Psi} * \Psi_{err} + k_y * y_{err} + k_i * \int y_{err} ds.$$

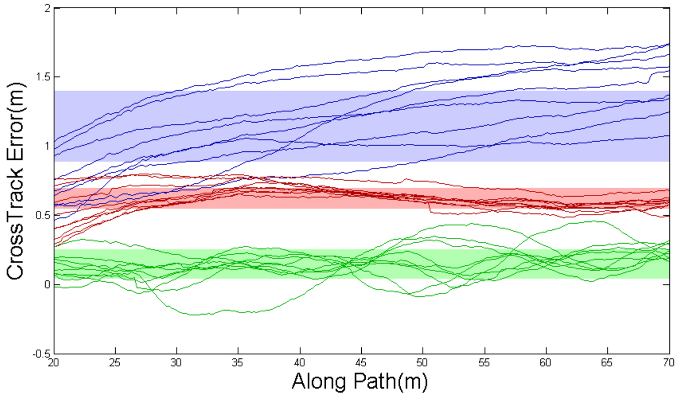

Fig. 7. Results of test runs performed in the Overbot. Blue lines show the original open loop approach, red lines show the PD control and green lines represent the PID control. The shaded areas depict the average ± the standard deviation.

Controller	Crosstrack Error(cm)	Standard Deviation(cm)
Open Loop	114.5	25.4
PD Control	62.3	7.3
PID Control	14.8	10.3

TABLE III
SUBSYSTEM IP ADDRESS ASSIGNMENT FOR OVERBOT VEHICLE

We expected to see the offset issue resolved as the integral term grew at a cost of an increase in the integral gain. While our results of an average 14.8 cm and standard deviation of 10.3 cm do correlated with our expectations, we did not see an elimination of the crosstrack error. In analyzing our data, we found that other errors were reducing performance in addition to the steering bias. Inaccuracies in the heading sensors, when magnified by the derivative gain, k_{Ψ}, resulted in similar bias issues.

HEADING CORRECTIONS

Because much of the Crossbow AHRS's heading estimates come from the use of a magnetometer, it is highly sensitive to extraneous magnetic fields. Changes in engine speed, the passing of another vehicle or building, or any reconfiguration of the vehicle can drastically effect the accuracy of the readings. Because much of the interference varies over time, the error bias is not consistent. We therefore have developed a dynamic approach to correct for the discrepancies.

When the vehicle is in motion, heading can be fairly accurately computed from the velocities reported through the Omnistar GPS service. If working in North East Down(NED) coordinates, the heading becomes a simple arc-tangent function. The raw results of this calculation, Ψ_{NED}, can have their own shortcomings given the accuracy of the GPS readings and the way measurements can change during vehicle turning. We therefore wish to smooth the calculations

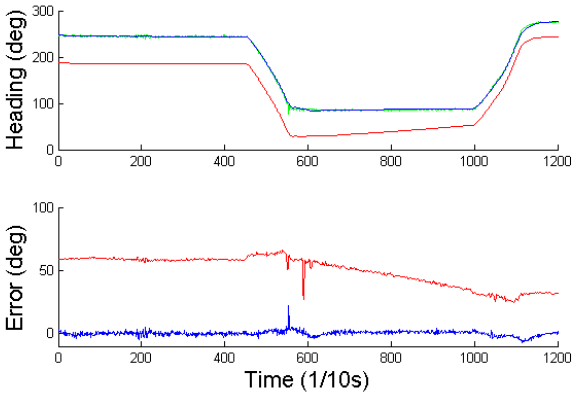

Fig. 8. Demonstration of heading corrections performed on vehicle data. Red shows sensor readings, green shows the heading computed from GPS and blue plots the corrected heading estimate.

by filtering over a sample history. This method would of course introduce a time delay into the system. Hence, we fit a least squares solution to the recent history of the arc-tangent calculations to estimate the true heading. From this estimate, we subtract the heading as reported by the AHRS unit, Ψ_{AHRS} to arrive at the heading error, $\Psi_{err} = \Psi_{NED} - \Psi_{AHRS}$. This result is further smoothed through a low pass (2nd order Butterworth) filter to produce a dynamic estimate for the AHRS error. By applying this correction to the AHRS measurements, we obtain a more reliable estimate of the true vehicle heading, while maintaining the responsiveness and sensitivity of the Crossbow AHRS[15].

We have developed this method and tested it on actual data sets collected from the vehicle. The graphs in Figure 8 demonstrate the improved accuracy that can be gained from such methods. The raw measurements consistently show errors of $30^\circ - 50^\circ$. After applying our correction techniques, the error drops to approximately $\pm 2^\circ$. It is clear to see why the graph of the heading as derived from the GPS velocities is not an ideal solution and motivates the use of the AHRS unit for the baseline measurement. Even though the bias estimation is generated from a time history, the correction is applied to real time measurements and no time lag is incurred. When the vehicle is not in motion it can be assumed that the heading is not changing. During very slight movements, where numerical error may become an issue in the above algorithm, the previous bias estimate can be applied until the vehicle speed sufficiently increases.

CONCLUSIONS

We have taken the Overbot Grand Challenge entry vehicle and are in the process of transforming this vehicle into a research platform suitable for autonomous vehicle experimentation. The vehicle itself remains quite capable, and can easily traverse very challenging terrain, and most of the sensors and actuators at this point remain the originals. We have begun to experiment with the path planning and control of the vehicle, in order to better utilize the existing sensors to avoid obstacles while maneuvering at speed.

Current deficiencies are apparent in the heading estimate of the Crossbow AHRS, which shows a large error and slow drift, and is very sensitive to external magnetic fields.

We have a rudimentary heading bias estimator based on the GPS heading while under motion, which seems to improve the estimate a great deal. Likewise, we have implemented a closed loop path controller, which demonstrates a much tighter control while tracking straight line segments over the original controller (7.3 cm vs. 25.4 cm crosstrack error standard deviation). In order to address the steering angle bias, integral control was added to the system. While this did improve the mean (14.8 cm vs. 62.3 and 114.5 for the PD and Original controllers respectively), it did so at the cost of much looser control. Future work will focus on tightening up the control with many more bias estimation states, improving the tight line following control, and implementing a spline based path planner to maintain speed while avoiding obstacles.

ACKNOWLEDGEMENTS

We would like to thank Team Overbot for their years of work designing, building and testing the Overbot. We could not have finished this work without the generosity of John Nagle, both in contributing the vehicle to UC Santa Cruz, but also to making himself available for help and support. We would also like to thank Chris Woodruff and Eli Kwitman from Harvey Mudd College for their hard work and long hours in the lab and the field. The contributions of everyone in the Autonomous Systems Lab at UC Santa Cruz helped ensure the success of this project.

REFERENCES

[1] E. Abbot, T. Morris, L. Alexander, and M. Donath. Land-vehicle navigation using gps. *Proceedings of the IEEE*, 87(1):145, 1999.
[2] Z. Berman and J.D. Powell. The role of dead reckoning and inertial sensors in future general aviation navigation. *Proceedings of the 1998 IEEE Position Location and Navigation Synposium*, page 510, 1998.
[3] D.M. Bevly, A. Rekow, and B. Parkinson. Evaluation of a blended dead-reckoning carrier phase differential gps system for control of an off-road vehicle. *Proceedings of the ION GPS Meeting, Nashville, TN*, page 2061, 1999.
[4] Jack E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25, 1965.
[5] Galil Motion Control. Dmc-14x5/6 user manual. 2005. `www.galilmc.com/support/manuals/man14x5_16.pdf`.
[6] G. Elkaim, M. O'Connor, T. Bell, and B. Parkinson. System identification of a farm vehicle using carrier-phase differential gps. *Proceeding of the Institute of Navigation, ION-GPS*, page 485, 1996.
[7] Crossbow Technology Incorporated. Ahrs400 series users manual. 2005. `www.xbow.com/Support/Support_pdf_files/AHRS400_Series_Users_Manual.pdf`.
[8] NovAtel Incorporated. Novatel propak-lb plus user manual. 2005. `www.novatel.com/Documents/Manuals/om-20000046.pdf`,.
[9] SICK Incorporated. Laser measurement systems technical description. 2003. `www.mysick.com/saqqara/im0012759.pdf`.
[10] Unibrain Incorporated. Fire-i 400 industrial camera technical specifications. `www.unibrain.com/Products/VisionImg/tSpec_Fire_i_Industrial.htm`.
[11] Polaris Industries. `www.polarisindustries.com`.
[12] Khian Hao Lim. Position estimation for team overbot. *Stanford University*, 2003.
[13] John Nagle, Achut Reddy, and Khian Hao Lim. Team overbot. `www.overot.com`.

[14] M. O'Connor, T. Bell, G. Elkaim, and B. Parkinson. Automatic steering of farm vehicles using gps. *3rd International Conference on Precision Agriculture, Minneapolis, Minnesota*, 1996.

[15] T. Schonberg, M. Ojala, J. suomela, A. Torpo, and A. Halme. Position an autonomous off-road vehicle by using fused dgps and inertial navigation. *Int J Syst Sci*, 27(8):745, 1996.