

● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

• • • • •

*bws22@cornell.edu, amn32@cornell.edu*

*e-mail:* [eg84@cornell.edu](mailto:eg84@cornell.edu)



WILEY  
InterScience®  
DISCOVER SOMETHING GREAT

## 1. INTRODUCTION

Field robotics has grown substantially in the past few decades, but a divide still remains between theoretical systems realized in ideal environments and those that must contend with the real world. The DARPA Grand Challenge sought to bridge this gap with a number of demanding technical and physical tasks, where only platforms robust both in theory and construction could hope to compete. The difficulty of the Grand Challenge spurred many unique approaches to the problem, each largely shaped by its team's technical, intellectual, and financial resources.

Cornell University entered the Grand Challenge in the summer of 2004 as a new team composed largely of undergraduate students and a few faculty advisors from the Cornell Schools of Mechanical and Aerospace Engineering, Electrical Engineering, and Computer Science. The addition of Singapore Technologies Kinetics as a primary sponsor in the fall provided the necessary financial support to make Cornell's entry a serious competitor. In the beginning, Team Cornell divided its work among three main subteams: a vehicle subteam, a sensing subteam, and an artificial intelligence subteam. As integration problems emerged between the three subteams, however, a more goal-oriented approach was necessary. The final solution divided the team according to four general goals: building a mobile hardware platform, determining the platform's orientation and location, sensing its surrounding environment, and planning a path through that environment.

This paper presents Cornell University's approach to accomplish each of these four goals, expanding on the high-level overview presented in Team Cornell's DARPA Grand Challenge technical report (Team Cornell, 2005). First, Section 2 presents a block diagram of Cornell's approach that outlines the critical subsystem as well as information flow between them. The subsequent sections present each subsystem in turn. Section 3 discusses the capabilities of Cornell's hardware platform, the Spider Light Strike Vehicle, and its actuation for drive-by-wire capabilities. Section 4 discusses the approach used to fuse inertial and navigation information to estimate the Spider's position, velocity, and orientation, with considerations for improving the robustness of the system in autonomous ground vehicles. Section 5 presents a novel terrain estimation algorithm that utilizes attitude and position estimates and terrain sensors to generate a dense elevation model, along with

a representation of uncertainty, for path planning without explicit obstacle identification. Section 6 presents a method for controlling the terrain sensors to improve the utility of data entering the terrain estimator. Section 7 presents a cubic spline-based path planner that utilizes terrain estimates and physical constraints to determine the final path traversed by the vehicle. Section 8 concludes with a performance analysis of Cornell's entry in the 2005 Grand Challenge.

## 2. SYSTEM SUBDIVISION AND INFORMATION FLOW

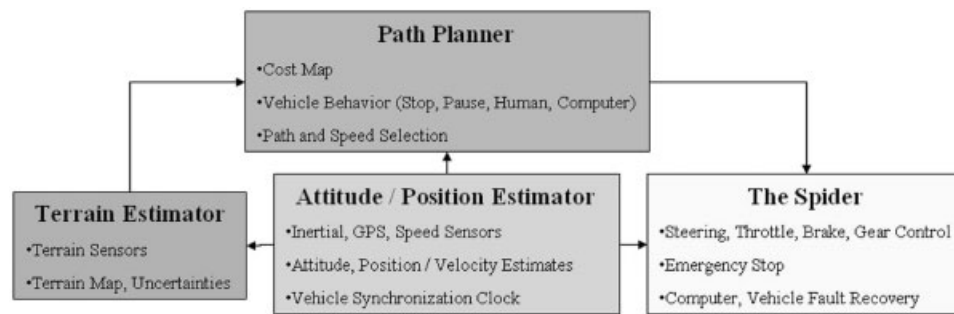
Cornell's approach to the Grand Challenge divides the problem into four separate goals: developing a robust platform, estimating the orientation and position of that platform, sensing its surroundings, and planning a path through its environment. The advantage of dividing the problem in this particular manner is that subteams can work on these goals independently using either simulated or logged data from other subteams. This structure allowed the entire team to work under a common architecture without integration bottlenecks.

Figure 1 shows the relation between the four components of Cornell's solution. The heart of the solution is the attitude and position estimator, which provides the other three subsystems with the Spider's current position, velocity, and orientation via direct serial connection, as well as a serial timing pulse for synchronization. The terrain estimator combines that information with terrain sensors to form a map of the world, which it sends to the path planner over a dedicated gigabit Ethernet connection. The path planner combines the environment map with the vehicle's state to generate a path, which it converts to a desired steering angle and speed. It then sends the desired steering angle and speed to the Spider's microcontroller directly over a 500 kbps CAN bus. This low level microcontroller then generates the steering, brake, throttle, and transmission commands necessary to control the Spider.

## 3. HARDWARE AND ACTUATION

### 3.1. The Spider Light Strike Vehicle

Cornell's Grand Challenge entry is based on a Spider Light Strike Vehicle, manufactured by Singapore



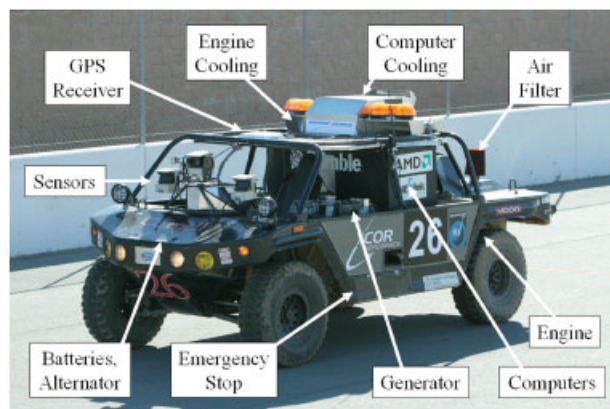
**Figure 1.** System division in Cornell's Spider. Each block uses separate computational resources and communicates to other blocks via serial, Ethernet, or controller area network (CAN) connections.

Technologies Kinetics (STK), shown in Figure 2. The Spider, nominally an off-road six-passenger military vehicle, is better suited to carry computers across harsh environments than a commercial sports utility vehicle or truck. Its fully independent suspension and 35-in. Goodyear Maximum Traction radial tires give it more than 16 in. of ground clearance, and its heavy-gauge steel tube-frame chassis, roll cage, and full underbody skid plate allow it to safely traverse desert brush and small ditches that would stop a standard commercial vehicle. The Spider's unequal-length double A-arm front suspension design is ideal for protecting its payload, as it generates negative camber gain during chassis roll to aid vehicle stability. The rear suspension is a semitrailing arm design preloaded with spring blades bolted to each trailing arm. Both front and rear suspensions are

equipped with nitrogen gas shock absorbers and two coilover spring/shock absorber pairs to further isolate the chassis.

The Spider is powered by a VM Motori 2.8 L, four cylinder common-rail diesel engine with a maximum output of 163 horsepower and 295 ft lbs of torque. The engine is coupled to an Audi three-speed semiautomatic transmission and configured for four-wheel drive, with a locked 1:1 front to rear output ratio. With a top speed of 50 miles per hour off-road and the ability to traverse grades of nearly 60%, the Spider's factory configuration is well equipped to drive the nominal Grand Challenge route and also to tolerate mistakes made by its autonomous components (Singapore Technologies Kinetics, 2006).

The Spider's electric components are run from three separate power buses: 12 and 24 V direct current (dc) sources, and a 110 V alternating current (AC) source, for design flexibility and compatibility with off-the-shelf automotive and computer components. The preinstalled 24 V source, which powers the Spider's actuators, sensors, gimbal, and emergency stop, is reinforced with two Optima deep-cycle car batteries charged by an engine-driven alternator. The 12 V source, powered from the 24 V source via dc-to-dc converter, is used to accommodate the standard automotive siren, engine control unit, and gauges, which are only available in 12 V configurations. A fuel-injected 5.5 kW Onan Commercial Mobile series generator provides 110 V ac power for the Spider's computers, which require clean, uninterrupted power even during sharp turns or when the Spider's engine is off. Five APC Smart-UPS 1000 VA units provide approximately 10 min of



**Figure 2.** Cornell's modified Spider Light Strike Vehicle.

reserve power to the ac source, allowing time for the generator to be automatically restarted if it stalls.

### 3.2. Design for Drive-By-Wire Operation

The Spider's steering, throttle, brake, and transmission are all actuated for computer control and drive-by-wire operation. Of these four actuated components, throttle control is the most straightforward. The Spider is wired at the factory for electronic throttle control through the engine control unit, so no additional work is necessary. For steering, where accurate tracking, no slip, and human drivability are all important design requirements. The Spider's steering column is directly actuated with a 1.1 hp MOOG 6000 series ac brushless motor and a pair of 1:1 spur gears that can be disengaged for human operation. A Celesco CLP-200 linear potentiometer is also installed directly on the tie rod for absolute steering angle measurements independent of the MOOG's encoders.

Like the steering system, the Spider's hydraulic disc brakes are actuated at the brake pedal to preserve human operability. The brake pedal is pulled toward the Spider's floor by a Maxon RE 40 brushed dc motor with a 43:1 GP 42C planetary gear head and sheet metal spool. This gear and spool system is capable of delivering 72 lbs force to the brake pedal, but more importantly, it can only pull the brake pedal downward. It therefore does not resist human operation, and it cannot be driven in reverse. The system is also equipped with a HEDS 5540 500 count-per-turn digital optical encoder on the planetary gear head for feedback.

The final actuated component enables gear selection on the Spider's transmission. Like the other three actuation schemes, the primary consideration in design of the transmission actuator is to preserve human operability. In the Spider, the transmission shifter has been modified to a single-axis shift pattern so it can be directly driven by a Maxon EPOS 70-4 rotary motor. This Maxon is equipped with the same model optical encoder as the brake actuator for position feedback.

Commands are issued to each of these actuators through a Motorola HCS12 16-bit microcontroller over the CAN bus. The HCS12 acts as the interface between the Spider's hardware and its computer payload, running 150 Hz proportional, integral, derivative loops to convert computer-supplied steering angle and speed commands into brake, throttle, and

steering wheel commands (Franklin, Powell & Emami-Naeini, 2002). This configuration allows the hardware interface to be completely transparent to the computers, so path planning and hardware development can take place simultaneously.

The HCS12 also monitors vehicle health to keep the Spider running in several contingency situations. It has a direct interface to the Spider's engine control unit, so it can detect engine stalls and restart the Spider if necessary. It also monitors the generator's output to restart it if it stalls. Finally, the HCS12 timestamps the commands received from the path planning computer. If these commands are ever interrupted, the HCS12 centers the Spider's wheel and brings it to a stop while the path planner restarts.

## 4. ATTITUDE AND POSITION ESTIMATION

Because the Grand Challenge route is specified as a set of absolute latitudes and longitudes, it is necessary for each Grand Challenge vehicle to be able to localize and orient itself with respect to the Earth. At first glance, both these tasks can be accomplished using one or more off-the-shelf Global Positioning System (GPS) receiver to measure vehicle position, velocity, and heading. Slow update rates, errors in the navigation solution, and signal interruptions, however, make it difficult for even the best differential GPS receivers to consistently localize a vehicle to sub-meter accuracy required to avoid obstacles. Early obstacle detection is also difficult without accurate position and orientation, as an obstacle detected at 20 m with  $\pm 5^\circ$  heading uncertainty has at least  $\pm 2$  m uncertainty in location just due to heading uncertainty alone.

The above considerations make a more accurate system necessary for long-horizon planning and robust path tracking. The system must be more than accurate, however; it must also be smooth and consistent. A proven solution to such requirements is the combination of GPS with a "strapdown" inertial navigation system (INS), where absolute GPS measurements are fused with vehicle accelerations and rotation rates measured at high frequency by an inertial measurement unit (IMU) (Ohlmeyer et al., 1997). The strapdown INS is self-contained and platform-independent, so it may be "strapped down" at any location on any rigid body to estimate its position and orientation. This flexibility is an important benefit in the Spider's design, where space is limited.



It also facilitates easy transfer between multiple test beds without the need for vehicle-specific dynamics models.

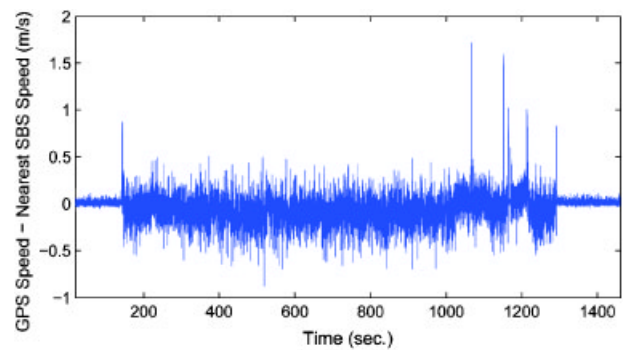
Although several off-the-shelf positioning systems are available, team Cornell opted to design its own for several reasons. First, off-the-shelf models are black boxes, so system faults arising from these systems are difficult to pinpoint. Second, high accuracy off-the-shelf models are expensive, whereas the team could obtain each individual sensor from sponsors cheaply. Finally, building the positioning system in house allowed it to be specialized for autonomous ground vehicle operation, thereby improving fault detection and recovery.

#### 4.1. Sensors for Inertial Navigation

The Spider's fused position and orientation estimates are provided by three sensors: a Litton LN-200 IMU, a Trimble Ag252 GPS receiver, and a vehicle speed sensor. Each of the three sensors affects the capabilities of the positioning system differently, and fusing them together brings the best qualities of each sensor to bear on the problem. The sensors are discussed below in turn.

The first, the LN-200 IMU, is a set of three-axis rate sensors and three-axis accelerometers. It provides measurements of vehicle accelerations and rates of rotation relative to an inertial frame. These measurements permit dead reckoning: numerical integration of accelerations and rates to obtain changes in position, velocity, and orientation. The IMUs 400 Hz-configured output makes it more than twice as fast as any other sensor on the Spider, and its tactical grade fiber optic rate sensors and silicon accelerometers have biases stable to within  $10^\circ/\text{h}$  and 3 mg, respectively (Northrop Grumman Navigation Systems Division, 2000). These capabilities help the positioning system generate smooth outputs at high rates.

The second sensor, a Trimble AgGPS 252 GPS receiver, provides absolute position and velocity measurements so the vehicle may be localized on the surface of the Earth. The Trimble boasts a number of important features, including dual frequency carrier phase filtering and ionospheric corrections, OmniSTAR high precision (HP) differential corrections, and multipath mitigation. These combine for position accuracy within 10 cm and velocity accuracy within 5 cm/s when tracking the OmniSTAR signal, and submeter accuracy without it (Trimble, 2004).



**Figure 3.** Errors between the SBS and the magnitude of GPS velocity in ideal GPS signal conditions.

Although the Trimble only provides measurements at 10 Hz, its absolute measurements keep numerical integration errors bounded.

The final sensor, a speed sensor (SBS), is mounted directly to the Spider's transmission. The SBS primarily serves to augment the inertial sensors in the absence of GPS signals, where it slows the growth rate of numerical integration errors. This sensor was pre-installed on the Spider, so its characteristics were determined by experimentation. Figure 3 compares the most current SBS speed to the magnitude of GPS velocity during a field test in the presence of the OmniSTAR HP signal. Fortunately, the errors between the two signals are statistically unbiased, with an experimental mean of  $-0.0465$  m/s in the data set of Figure 3. The standard deviation in the error signal,  $0.1531$  m/s, also gives an indication of the quality of the SBS measurements for tuning the positioning system.

#### 4.2. Positioning System Equations of Motion

##### 4.2.1. Attitude Dynamics

The attitude of a positioning system can be represented by one of several different sets of variables: Euler angles, quaternions, or a direction cosine matrix (DCM). The Cornell implementation uses a triad of ZY'X'' Euler angles: yaw  $\psi$ , pitch  $\phi$ , and roll  $\theta$ , to represent the orientation of the Spider with respect to an East North Up (ENU) reference frame at the Spider's current latitude and longitude. The Euler angles are chosen over quaternions or a DCM for two reasons: they have an intuitive physical inter-

pretation for quick debugging, and they are a minimal set of parameters. The latter point is important for Cornell's estimators, which cannot provide state estimates if the covariance matrix  $P$  is never invertible. The dynamic relations for the Euler angles defined for this problem are given below (Savage, 1998; Triantafyllou & Hover, 2004):

$$\begin{pmatrix} \dot{\psi} \\ \dot{\phi} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & s\theta/c\phi & c\theta/c\phi \\ 0 & c\theta & -s\theta \\ 1 & s\theta t\phi & c\theta t\phi \end{pmatrix} \left[ \underline{\omega}_{\text{IMU}}^B - \underline{b}_{\text{rg}}^B - \underline{\eta}_{\text{rg}}^B - \underline{\omega}_E^B - \frac{\underline{e}_R^B \times \underline{v}^B}{R} \right] \quad (1)$$

using the abbreviations  $s \cdot = \sin(\cdot)$ ,  $c \cdot = \cos(\cdot)$ , and  $t \cdot = \tan(\cdot)$ , and letting superscripts denote the coordinate frame in which a particular vector is expressed. In Eq. (1),  $\underline{\omega}_{\text{IMU}}^B$  is the vector of IMU rates,  $\underline{b}_{\text{rg}}^B$  is the vector of rate sensor biases,  $\underline{\eta}_{\text{rg}}^B$  is a zero mean, white noise process,  $\underline{\omega}_E^B$  is the Earth's angular velocity,  $\underline{e}_R^B$  is a unit vector from the center of the Earth toward the vehicle expressed in body coordinates,  $\underline{v}^B$  is the vehicle's velocity relative to the spinning Earth expressed in body coordinates, and  $R$  is the vehicle's distance from the center of the Earth. Intuitively, the bracketed vector term is the angular velocity of the vehicle relative to the ENU reference frame and expressed in the vehicle body ( $B$ ) frame. This quantity is obtained from the IMU rate sensors by correcting for the rotation of the Earth and ENU frame, as well as bias and sensor noise. Note Eq. (1) defines rate gyro errors in terms of additive biases  $\underline{b}_{\text{rg}}$  and white noise  $\underline{\eta}_{\text{rg}}$ . The additive biases are modeled as random walk processes

$$\dot{\underline{b}}_{\text{rg}} = \underline{n}_{\text{rg}}, \quad (2)$$

where  $\underline{n}_{\text{rg}}$  is a zero mean, white noise process. Other gyro errors, such as  $g$  sensitivity, scale factor, and misalignments, are not included in this model to improve computational speed. Despite these exclusions, it is shown below that the filter is still statistically significant, due in part to the quality of the fiber optic gyros.

#### 4.2.2. Position and Velocity Dynamics

Like attitude, the position and velocity dynamics of a positioning system have several different representations, including latitude, longitude and altitude, Earth-Centered Inertial coordinates, Earth-Centered Earth-Fixed (ECF) coordinates, and ENU coordinates. This implementation tracks vehicle position and velocity in ECF coordinates relative to the rotating Earth, primarily due to the simplicity of the dynamics equations. The position and velocity dynamics for this coordinate frame can be derived from the vector equations of velocity and acceleration relative to a moving coordinate frame (Moon, 1998; Savage, 1998):

$$\dot{\underline{p}}^{\text{ECF}} = \underline{v}^{\text{ECF}}, \quad (3)$$

$$\begin{aligned} \dot{\underline{v}}^{\text{ECF}} = & \underline{a}_{\text{IMU}}^{\text{ECF}} - \underline{b}_a^{\text{ECF}} - \underline{\eta}_a^{\text{ECF}} + \underline{g}^{\text{ECF}} - (\underline{\omega}_E^{\text{ECF}} \times \underline{\omega}_E^{\text{ECF}} \\ & \times R \underline{e}_R^{\text{ECF}}) - 2(\underline{\omega}_E^{\text{ECF}} \times \underline{v}^{\text{ECF}}), \end{aligned} \quad (4)$$

where  $\underline{b}_a^{\text{ECF}}$  is a vector of accelerometer biases,  $\underline{\eta}_a^{\text{ECF}}$  is a vector of zero mean white noise,  $\underline{g}^{\text{ECF}}$  is the gravity vector, and the final two terms correct for centripetal and coriolis accelerations, respectively. Note Eq. (4) specifies an accelerometer error model consisting only of biases  $\underline{b}_a^{\text{ECF}}$  and white noise  $\underline{\eta}_a^{\text{ECF}}$ . Like the rate gyro biases, the accelerometer biases are modeled as random walk processes

$$\dot{\underline{b}}_a = \underline{n}_a. \quad (5)$$

Other error terms, such as scale factors and misalignments, are not modeled. As with the rate gyros, it is shown below that statistical significance is still maintained within the filter. Gravity is also defined using a simple ellipsoidal model

$$\underline{g}^{\text{ECF}} = -9.8 \frac{\left( p_x p_y \frac{a_e^2}{b_e^2} p_z \right)^T}{\sqrt{p_x^2 + p_y^2 + (a_e^4/b_e^4) p_z^2}}, \quad (6)$$

where  $a_e$  and  $b_e$  are the lengths of the semimajor and semiminor axes of the WGS-84 ellipsoid model of the Earth, and  $p_x$ ,  $p_y$ , and  $p_z$  are the three components of the ECF position of the vehicle. It is shown below that

the filter can be made statistically significant despite the gravity model. A more complicated spherical harmonic expansion was therefore not tried.

#### 4.3. Fusing the Inertial Navigation Sensors

The three sensors discussed in Section 4.1 are fused into an estimate of the Spider's position, velocity, and attitude using an extended Kalman Filter (EKF). The filter operates in two steps: a prediction and an update (Shalom, Rong Li & Kirubarajan, 2001). In the prediction step, the estimates of attitude, position, velocity, and bias are integrated across one IMU sample interval ( $dt=0.0025$  s) according to Eqs. (1)–(5) to make an *a priori* prediction of the estimates at the next time step. Because Eqs. (1) and (4) have no general closed-form solution, a fourth order Runge-Kutta numerical integration is performed in place of an exact integration (Battin, 1999). During each numerical integration, the IMU accelerations and rates of rotation are assumed constant.

The filter also maintains an estimate of its estimation error covariance,  $P(k)$  at each time index  $k$ . During the prediction step, the estimated error covariance is advanced in time to form an *a priori* estimate  $\bar{P}(k+1)$  of the state error covariance. Because the attitude and velocity dynamics are nonlinear, a linearization and discretization of the continuous dynamics is made at each time step (Bar-Shalom et al., 2001):

$$\begin{aligned}\bar{P}(k+1) &\approx F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k), \\ F(k) &= I + dtJ_x(k) + \frac{1}{2}dt^2J_x^2(k) + \frac{1}{6}dt^3J_x^3(k), \\ \Gamma(k) &= [dtI + \frac{1}{2}dt^2J_x(k) + \frac{1}{6}dt^3J_x^2(k) + \frac{1}{24}dt^4J_x^3(k)]J_v(k),\end{aligned}\quad (7)$$

where  $J_x(k)$  and  $J_v(k)$  are the Jacobians of Eqs. (1)–(5) with respect to the state and noise inputs, respectively, evaluated at their current estimates, and  $Q(k)$  is a matrix of white noise intensities assigned to the noise inputs during filter tuning.

In the update step, the predicted state  $\bar{x}(k+1)$  and its covariance  $\bar{P}(k+1)$  are combined with sensor measurements  $\underline{z}(k+1)$  of certain elements of the state in a linear minimum mean square estimator to form

*a posteriori* state estimates  $\hat{x}(k+1)$  and their updated error covariance  $P(k+1)$  (Bar-Shalom et al., 2001):

$$\underline{z}(k+1) = H(k+1)\underline{x}(k+1) + \underline{w}(k+1),$$

$$S(k+1) = H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1),$$

$$W(k+1) = \bar{P}(k+1)H^T(k+1)S^{-1}(k+1),$$

$$\underline{\hat{x}}(k+1) = \underline{\bar{x}}(k+1) + W(k+1)\underline{v}(k+1),$$

$$P(k+1) = \bar{P}(k+1) - W(k+1)S(k+1)W^T(k+1), \quad (8)$$

where  $\underline{w}(k+1)$  is a vector of zero mean white measurement noise with covariance intensity  $R(k)$ , set during filter tuning, and  $\underline{v}(k+1) = \underline{z}(k+1) - H(k+1)\bar{x}(k+1)$  is called the “innovation.”

The measurements  $\underline{z}$  used in the update step are extracted from the three sensors that make up the positioning system. Position, velocity, and yaw (heading) are reported directly by the GPS receiver. Speed is reported by the SBS sensor, and a full ECF vector velocity measurement is created from it using vehicle attitude as an exogenous input. Pitch and roll measurements can also be extracted if the vehicle is at rest by using the accelerometers to measure the direction of the gravity vector in the body frame  $\underline{e}_g^B$  and comparing it to a theoretical downward-pointing  $\underline{e}_g^B$ .

$$\underline{e}_g^B = \begin{pmatrix} s\phi \\ -s\theta c\phi \\ -c\theta c\phi \end{pmatrix} \approx \underline{a}_{\text{IMU}}^B \quad (9)$$

from which pitch  $\phi$  and roll  $\theta$  can be extracted, assuming  $-\pi/2 < \phi < \pi/2$ . If the vehicle is moving, a similar calculation can be done using Eq. (4) to solve for  $\underline{g}^{\text{ECF}}$ , with  $\underline{\dot{v}}^{\text{ECF}}$  approximated by a finite difference of GPS velocity. The moving calculation is noisier, however, as it depends on existing attitude estimates to transform  $\underline{g}^{\text{ECF}}$  into  $\underline{g}^B$ .

While many EKF implementations use a single filter to estimate attitude, position, and velocity together, Cornell uses a pair of filters to estimate attitude and position/velocity separately. That is, one filter fuses Eqs. (1) and (2) with measurements of yaw, pitch, and roll to estimate attitude states, and a second filter fuses Eqs. (3)–(5) with position and velocity measurements to estimate position and velocity states. To do so, the attitude estimator treats position and velocity as exogenous noisy inputs, and the position/velocity filter treats attitude as an exogenous noisy input. The motivation for this structure is computational: decoupling the states cuts the sizes of the matrices in Eqs. (7) and (8) almost by a factor of 4, enabling 400 Hz predictions on the 3.4 GHz Pentium IV attitude computer equipped with Windows Server 2003 that match the IMU rates.

The EKFs in Cornell's implementation are both square root information filters (SRIFs), a numerically robust implementation of the Kalman Filter (KF) algorithm (Bierman, 1977). The SRIF has two main advantages over standard KF implementations. First, matrix computations in the SRIF are performed using square root versions of the KF matrices, so numerical precision is effectively double that of the standard Kalman Filter. Second, computations are performed using the inverse of the covariance matrix, which allows a numerically correct representation of infinite covariance during filter initialization. Bierman (1977) describes the SRIF in detail.

#### 4.4. Filter Tuning and Verification

Although the main goal of the filters is to estimate the Spider's attitude, position, and velocity, a second goal of equal importance is to evaluate the accuracy of those estimates. That accuracy, stored in the state error covariance matrix  $P(k)$ , is essential for calculating correct filter updates (Bar-Shalom et al., 2001). In addition, accurate estimates of error statistics are necessary for identifying and localizing terrain in a probabilistically rigorous manner, which serves as the foundation of the Spider's terrain estimation algorithm.

One traditional method for evaluating the estimators' validity and statistical significance is to perform consistency tests (Bar-Shalom et al., 2001). These are statistical hypothesis tests of the assumptions made by the filters: whether the linearizations are valid, the dynamics and sensor error models are correct, and the reported covariance accurately de-

scribes the error statistics of the estimators. If these filter assumptions are correct, one consequence is that the innovations  $\underline{\nu}(k)$  will be zero mean and white, with covariance  $S(k)$  given in Eq. (8). A consistency test statistic for these hypotheses is based on the observed sequence of innovations

$$\bar{\epsilon}_\nu = \frac{1}{N} \sum_{k=k_0}^{k_0+N-1} \underline{\nu}(k)^T S^{-1}(k) \underline{\nu}(k). \quad (10)$$

If the filter assumptions are valid, the test statistic follows a chi-square distribution

$$\bar{\epsilon}_\nu \sim \chi_{Nn_z}^2, \quad (11)$$

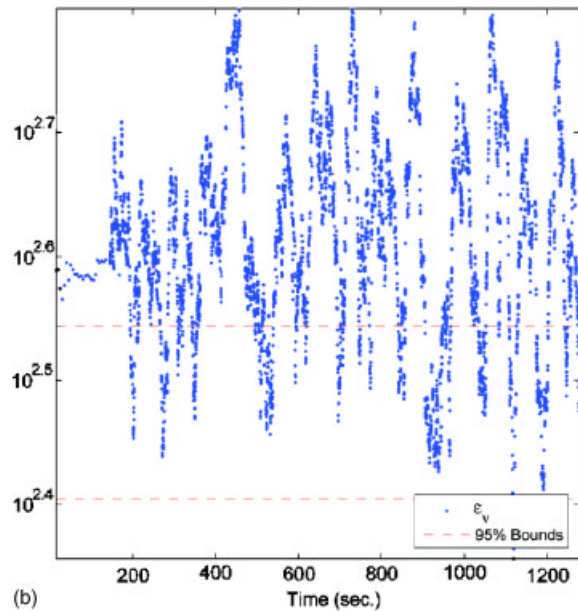
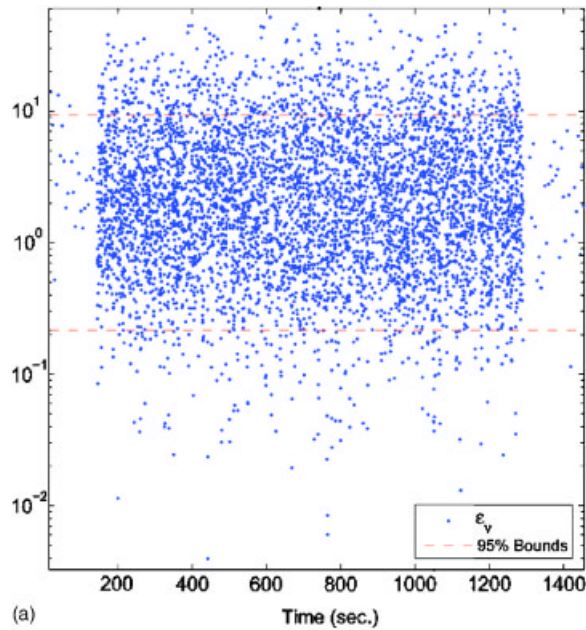
where  $n_z$  is the length of the measurement vector  $\underline{z}(k)$  (Bar-Shalom et al., 2001).

The statistical significance, correctness, and consistency of the Cornell filters are tested using observed innovation vectors taken during a test run in a parking lot. The values of the test statistics for the attitude filter are plotted in Figures 4(a) and 4(b) for  $N=1$  and  $N=100$ . Experimentally, approximately 84.7% of the test statistic values lie in the 95% interval for  $N=1$ , suggesting that the attitude filter makes accurate assumptions and reports statistically significant estimates and covariances for the majority of the test run. The filter shows limitations for the more stringent  $N=100$  test, however, where only 21.4% of the values lie in the 95% interval. The attitude filter is therefore overconfident on average, but consistent for each individual measurement. Time constraints in the Grand Challenge prevented further experimentation to reduce the filter's overconfidence, though the time-correlations of the test statistic in Figure 4(b) suggest the white noise assumption in GPS-derived attitude measurements might be to blame.

The values of the test statistic for the position/velocity filter are plotted in Figures 5(a) and 5(b) for  $N=1$  and  $N=100$ . The results are similar to those of the attitude estimator. With the  $N=1$  single measurement test, 78.4% of the test statistic values lie in the 95% interval. In the  $N=100$  averaged measurement test, only 21.6% of the test statistic values lie in the 95% interval. The results show the position/velocity filter to report valid covariance matrices in most iterations, though it is overconfident on average.

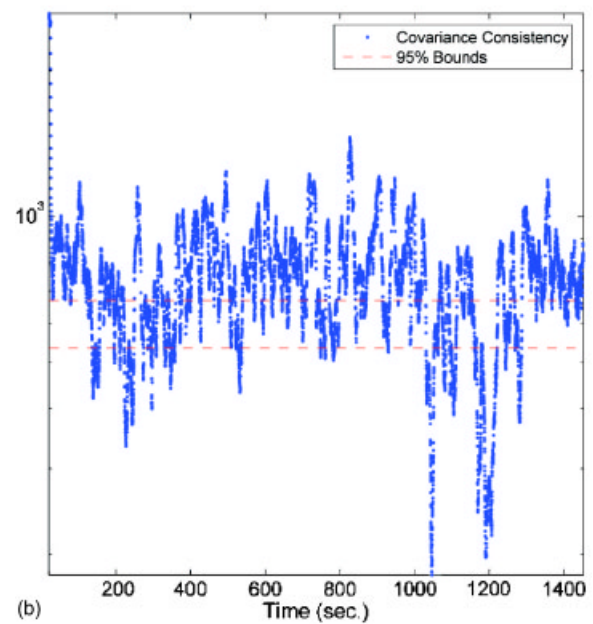
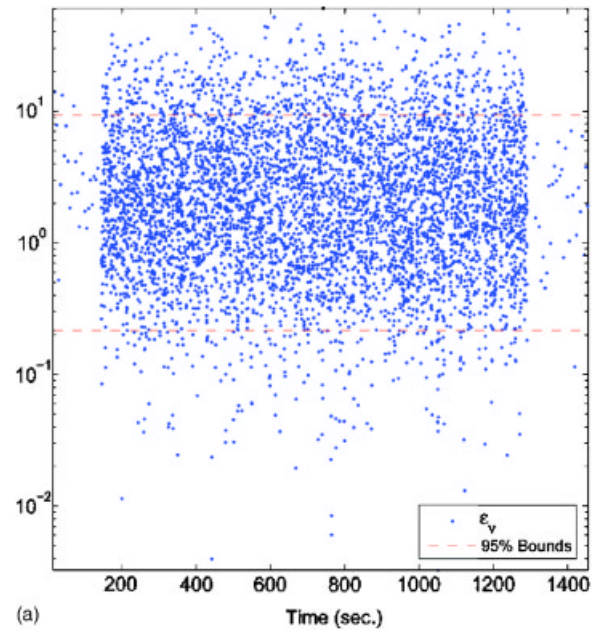
The statistical consistency test described above





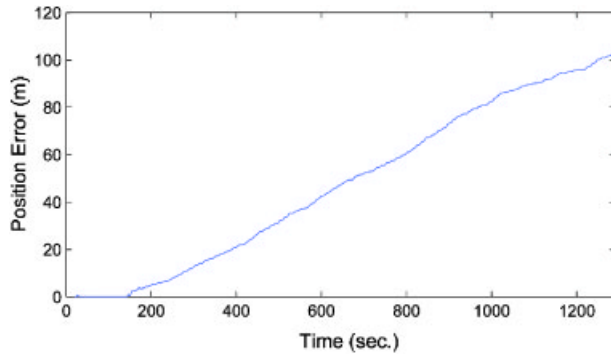
**Figure 4.** (a) Consistency test statistic  $\bar{\epsilon}_v$  and 95% confidence bounds for a calibration run of the attitude estimator, with  $N=1$ . (b)  $\bar{\epsilon}_v$  and 95% confidence bounds during the same trial with  $N=100$ .

is also used to tune the Cornell filters. That is, the process and measurement noise covariance matrices  $Q(k)$  and  $R(k)$  from Eqs. (7) and (8) are tuned using test runs until the filter is as consistent as possible.



**Figure 5.** (a) Consistency test statistic  $\bar{\epsilon}_v$  and 95% confidence bounds for a calibration run of the position/velocity estimator, with  $N=1$ . (b)  $\bar{\epsilon}_v$  and 95% confidence bounds during the same trial with  $N=100$ .

The fact that the Cornell filters can be tuned to be consistent means simultaneously that the modeling approximations are valid, the filters are tuned properly, and they report accurate covariance matrices  $P(k)$ . Once tuned correctly, the covariance matrices



**Figure 6.** Position errors from dead reckoning with Cornell's IMU and wheel speed sensor during an artificially created GPS blackout starting at  $t=140$  s.

in the Cornell filters typically report root-mean-square errors of 0.02 m in each axis in position, 0.05 m/s in each axis in velocity, and  $0.2^\circ$  in heading in the presence of the OmniSTAR signal, making them competitive with many off-the-shelf positioning systems available.

#### 4.5. Special Filtering Considerations

Despite filter correctness and consistency, Cornell encountered two field events requiring special consideration: loss and reacquisition of the GPS signal. In principle, filtering theory has no difficulty handling these situations, as the IMU and speed sensor are used for repeated prediction steps until the GPS signal is reacquired. From a practical point of view, however, these events generate difficulties. In particular, numerical integration errors in the absence of a GPS signal grow in time without bound (Sukkarieh, Nebot & Durrant-Whyte, 1999). With no prior information about the course and no absolute measurements during GPS blackouts, Cornell relies on the quality of the inertial sensors to keep estimation errors small. Figure 6 shows the errors accumulated by dead reckoning with the IMU and speed sensor during an artificially created GPS blackout. Position errors reach only 4.8 m after a 60 s blackout, and the rate of growth is approximately constant. With the open desert environment, an average corridor width of more than 6.5 m, and average speeds nearing 9 m/s, this level of blackout tolerance is sufficient for blackout situations likely to be encountered in the Grand Challenge.

While GPS signal blackouts cause errors that grow over time, a more acute event occurs when the GPS signal is reacquired. The problem lies in the fact that the filter update step of Eq. (8) produces discontinuous jumps in state estimates, particularly position. Although these discontinuities are consistent with Kalman Filtering theory, they cause the vehicle to turn sharply if fed back in a path tracking controller. To limit the size of these discontinuities, a "rate limiting" modification is performed in the position filter update step. The strategy works as follows.

1. Calculate the position and velocity discontinuities during a position filter update

$$\Delta \underline{p}^{\text{ECF}}(k) = \hat{\underline{p}}^{\text{ECF}}(k) - \bar{\underline{p}}^{\text{ECF}}(k),$$

$$\Delta \underline{v}^{\text{ECF}}(k) = \hat{\underline{v}}^{\text{ECF}}(k) - \bar{\underline{v}}^{\text{ECF}}(k). \quad (12)$$

Also calculate the magnitude  $d_p$  of the position discontinuity.

2. If  $d_p$  is larger than some tolerable position jump  $d_{\text{max}}$ , calculate the discount factor  $c_0 = d_{\text{max}}/d_p$ .
3. Calculate the rate limited measurement update to limit the position discontinuity

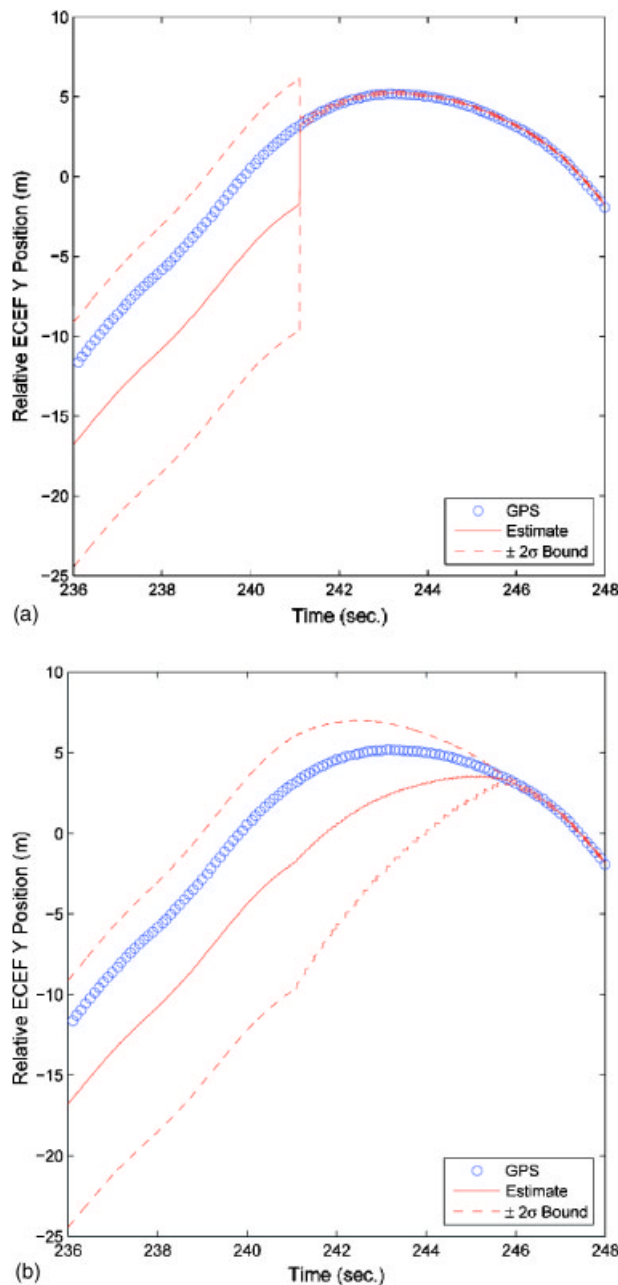
$$\begin{bmatrix} \hat{\underline{p}}^{\text{ECF}}(k) \\ \hat{\underline{v}}^{\text{ECF}}(k) \end{bmatrix} = \begin{bmatrix} \bar{\underline{p}}^{\text{ECF}}(k) + c_0 \Delta \underline{p}^{\text{ECF}}(k) \\ \bar{\underline{v}}^{\text{ECF}}(k) + c_0 \Delta \underline{v}^{\text{ECF}}(k) \end{bmatrix}. \quad (13)$$

4. Calculate the rate limited filter covariance

$$\begin{aligned} P(k) = & [I - c_0 W(k) H(k)] \\ & \times \bar{P}(k) [I - c_0 W(k) H(k)]^T \\ & + c_0^2 W(k) R(k) W(k)^T. \end{aligned} \quad (14)$$

This rate limited covariance update must be used instead of Eq. (8) when position and velocity are rate limited if the filter is to be kept accurate and consistent (Bar-Shalom et al., 2001).

This rate limiting strategy modifies the filter gain  $W(k)$  to ensure smooth reacquisition of GPS signals. It also allows the filter to be tuned to the vehicle controller, as the rate limiting threshold  $d_{\text{max}}$



**Figure 7.** (a) Without rate limiting, the position estimate makes a discontinuous jump of more than 5 m during one GPS signal reacquisition. (b) Rate limiting causes the position estimate and its uncertainty to converge smoothly.

may be set to ensure the maximum distance discontinuity per second does not exceed typical vehicle speeds. Figure 7 shows the effects of rate limiting on reacquisition after an artificially created GPS blackout. Without the rate limiting, the position estimate

undergoes an instantaneous jump of more than 5 m for this example. In the field, a discontinuous jump of that magnitude would cause the vehicle to swerve and possibly roll. The rate limited position estimate, in contrast, smoothly converges toward the accurate GPS signal. The uncertainty in the estimate also decreases slowly.

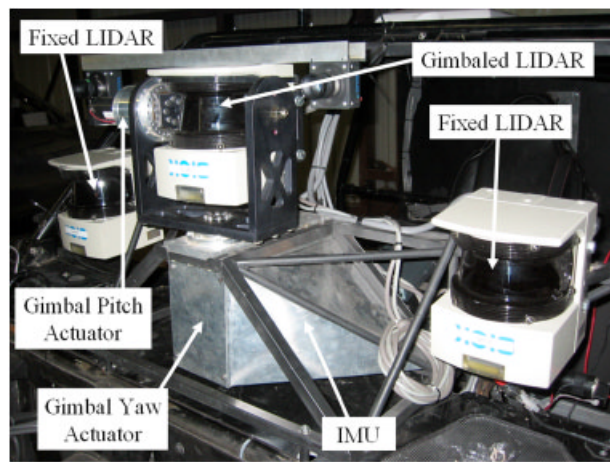
Although effective at solving the GPS reacquisition problem, the rate limiting strategy has several drawbacks. First, the rate limited updates are not true minimum mean squared error (MMSE) estimates, even for estimation on a linear system. Second, the more computationally expensive Eq. (14) must be used to update filter covariance during rate limiting, as the update of Eq. (8) is not valid for arbitrary filter gains (Bar-Shalom et al., 2001). Finally, rate limiting has the potential to smear any absolutely-mapped objects. However, because Eq. (14) maintains accurate filter covariance matrices, the uncertainty in these objects decreases slowly as they converge to more accurate estimates. Without rate limiting, any absolutely mapped objects make large discontinuous jumps on reacquisition of a GPS signal. Such discontinuities violate the slowly-changing world assumption required by Cornell's path planner, resulting in indecision and crashes during testing.

## 5. TERRAIN SENSING AND MAPPING

The 2005 DARPA Grand Challenge places great emphasis on the vehicles' ability to detect unsafe regions and unplanned obstacles inside the route boundaries. To distinguish between safe and unsafe regions of the route, each vehicle must be equipped with terrain sensors or obstacle detectors and an algorithm to localize the vehicle with respect to the sensed objects. Historically there have been a wide variety of approaches to this mapping problem, including maps of discrete obstacles or landmarks, grid-based approaches, statistical and stochastic models, and even polygonal object representations (Thrun, 2002).

Cornell's terrain representation is grid based, similar to the representation in Olin & Tseng (1991). That is, terrain is divided into 40 cm  $\times$  40 cm grid cells in a local ENU plane, with the data in each cell describing that cell's elevation. However, unlike Olin & Tseng (1991), Cornell uses a novel statistical terrain estimation algorithm to generate MMSE terrain estimates within each grid cell. This approach generates





**Figure 8.** The Spider's sensor platform and two-axis gimbal.

not only terrain estimates in real time, but also their uncertainties, enabling paths to be selected based on statistical statements of traversability.

### 5.1. Terrain Sensors

Sensed terrain data for the terrain estimator is provided by three SICK LMS 291 laser rangefinders (LIDARs). Each of these LIDAR units scans along a single line and is configured to return 181 ranges spaced at half-degree increments over a 90° field of view, all at 75 Hz (SICK, 2003). Each LIDAR returns data via 500 kbps serial connection to the terrain computer, an AMD 2 GHz quad Opteron 846 server running Windows Server 2003, where the data are fused into terrain estimates. Figure 8 shows the three LIDAR units, each mounted to the hood of the Spider. The outer two LIDARs are mounted rigidly using a turnbuckle and rod ends to allow their pitch to be set without inducing yaw or roll. Both LIDARs face in the direction of vehicle travel; one is pitched at approximately 4° to scan 15 m in front of the vehicle, and the other is pitched at approximately 6° to scan the ground at 20 m. The central LIDAR sits on a two-axis gimbaled platform. The platform is actuated in the yaw and pitch axes with two MAXON EPOS 70-4 rotary motors and separate Harmonic Drive gear trains, allowing the gimbaled LIDAR to be aimed by the path planner.

Cornell's sensor orientation is driven by two fac-

tors: budget constraints and terrain representation. Budget constraints limited the team to three LIDAR units, so each unit is positioned to sense terrain without risking damage upon collision. For this reason, the LIDARs are all mounted on the hood of the vehicle rather than near the ground. The LIDARs are aimed downward to provide the most useful information for the terrain estimation algorithm, which fuses data from all three LIDARs into a single persistent map. Because the LIDARs reinforce each other in their measurements of the terrain in front of the vehicle, the system is less sensitive to false obstacles than one in which each LIDAR is positioned to sense only one type of obstacle, such as an obstacle in the path of one of the vehicle's wheels.

### 5.2. Terrain Estimation Algorithm

The goal of the terrain estimation algorithm is to produce an elevation estimate within each grid cell, optimal in the sense of MMSE. Three steps are performed to generate these terrain estimates. First, a statistical representation of each sensor measurement is formed to account for multiple sources of error in a probabilistically rigorous manner. Second, each measurement is probabilistically assigned or "associated" to one or more terrain cells in which it is likely to belong. Finally, the measurements assigned to each cell are fused in real time into a single MMSE estimate of the elevation within each cell. These three steps are discussed in turn below.

#### 5.2.1. Statistical Treatment of Sensor Measurements

The first step in the terrain estimation algorithm is to form a statistical representation of each LIDAR measurement to account for different errors in the sensing path. These errors include those due to the LIDAR itself, errors due to sensor orientation, errors due to the attitude estimator, and errors due to the position/velocity estimator (Huising & Pereira, 1998). The first of these error types are due to the LIDAR itself, and include range errors and errors due to expansion of the LIDAR beam. Range errors add directly to the ranges reported by the LIDAR, and have typical accuracy of  $\pm 5$  cm out to a range of 80 m when the LIDAR is configured in centimeter mode (SICK, 2003). Beam expansion errors, in contrast, are errors in detection angle due to the fact that LIDAR ranges are generated from reflections of a beam of light that expands as it travels from the



LIDAR. Beam expansion accounts for approximately  $\pm 0.24^\circ$  detection angle error for each LIDAR measurement (SICK, 2003).

The second source of sensing error is uncertainty in the sensors' orientations. For fixed sensors, this type of uncertainty can be eliminated by offline calibration routines against objects of known location. However, for the gimbaled LIDAR, orientation errors arise due to inaccuracies in the angle encoders on the MAXON motors actuating the gimbal. These errors affect the localization of measurements in the vehicle body frame.

The third and fourth sources of sensing error are uncertainty due to the attitude and position estimators. While these errors do not affect the sensor readings themselves, they introduce error in locating measurements on an absolute map. The statistical properties of these errors are captured by the attitude and position estimates  $\hat{x}(k)$  and their covariances  $P(k)$  from Eq. (8).

In order to understand in a statistical sense how these four types of error affect each LIDAR measurement, it is first necessary to define the transformation in which they play a part. For this task, each valid sensor measurement is combined with the sensor's orientation, the vehicle's orientation, and the vehicle's location to express the measurement in the fixed ENU coordinates of the terrain map

$$\underline{r}^{\text{ENU}} = (E^{\text{ENU}} \quad N^{\text{ENU}} \quad U^{\text{ENU}})^T = f(\underline{p}, \underline{r}) \quad (15)$$

with

$$\underline{p} = (\underline{L}_i^T, \underline{x}_a^T, \underline{x}_p^T)^T, \quad (16)$$

$$\underline{r} = (\rho, \theta_D)^T,$$

where  $\underline{L}_i$  is a vector of parameters describing the location and orientation of the  $i$ th LIDAR with respect to the vehicle body frame,  $\underline{x}_a$  is the vehicle's attitude at the time of measurement,  $\underline{x}_p$  is the vehicle's position at the time of measurement,  $\rho$  is a range returned by the LIDAR, and  $\theta_D$  is the scalar angle of the measurement within the LIDAR's scanning plane. Intuitively, these parameters are divided into the vector  $\underline{p}$  of LIDAR orientation parameters, and a measurement  $\underline{r}$ . Note in general the transformation function  $f(\cdot)$  of Eq. (15) is a nonlinear function of these parameters. For the Cornell implementation, it is constructed from a set of  $4 \times 4$  matrix transformations, with each matrix a function of one parameter as in Murray, Li & Sastry (1994) and Moon (1998). A sample sequence of these transformations is as follows.

1. Rotate the coordinate frame about its Z axis by  $-\theta_D$  to express the measurement  $(\rho \ 0 \ 0 \ 1)^T$  in LIDAR-centered coordinates, where  $\rho$  is a range reported by the LIDAR and  $\theta_D$  is its detection angle.
2. Rotate the coordinate frame by  $-S_\theta$  about its X axis, where  $S_\theta$  is the LIDAR's roll angle with respect to the Spider.
3. Rotate the coordinate frame by  $-S_\phi$  about its Y axis, where  $S_\phi$  is the LIDAR's pitch angle with respect to the Spider.
4. Rotate the coordinate frame by  $-S_\psi$  about its Z axis, where  $S_\psi$  is the LIDAR's yaw angle with respect to the Spider.
5. Translate the origin of the coordinate frame by  $(-S_x \ -S_y \ -S_z)^T$ , where  $S_x$ ,  $S_y$ , and  $S_z$  represent the location of the LIDAR with respect to the vehicle body origin, centered at the IMU.
6. Rotate the coordinate frame by  $-\theta$  about its X axis, where  $\theta$  is the vehicle's roll angle.
7. Rotate the coordinate frame by  $-\phi$  about its Y axis, where  $\phi$  is the vehicle's pitch angle.
8. Rotate the coordinate frame by  $-\psi$  about its Z axis, where  $\psi$  is the vehicle's yaw angle.
9. Translate the origin of the coordinate frame by  $(-O_x \ -O_y \ -O_z)^T$ , where  $O_x$ ,  $O_y$ , and  $O_z$  represent the location of the vehicle with respect to the fixed ENU origin.

To augment each LIDAR measurement with an expression of its uncertainty, each of the parameters is modeled as corrupted by an additive measurement noise

$$\underline{p} = \hat{\underline{p}} + \delta \underline{p},$$

$$\underline{r} = \hat{\underline{r}} + \delta \underline{r}, \quad (17)$$

where  $\hat{\underline{p}}$  and  $\hat{\underline{r}}$  are the measured values of  $\underline{p}$  and  $\underline{r}$ , and  $\delta \underline{p}$  and  $\delta \underline{r}$  are measurement noise, assumed to be zero mean, Gaussian, and mutually uncorrelated. Using

this representation, the terrain measurement  $\underline{r}^{\text{ENU}}$  can be written

$$\underline{r}^{\text{ENU}} = f(\underline{\hat{p}} + \delta \underline{p}, \underline{\hat{r}} + \delta \underline{r}). \quad (18)$$

A typical linearization can then be made to characterize the true measurement and the statistics of its uncertainty (Bar-Shalom et al., 2001):

$$\underline{r}^{\text{ENU}} \approx f(\underline{\hat{p}}, \underline{\hat{r}}) + J_p(\underline{\hat{p}}, \underline{\hat{r}}) \delta \underline{p} + J_r(\underline{\hat{p}}, \underline{\hat{r}}) \delta \underline{r}, \quad (19)$$

where  $J_p(\cdot)$  and  $J_r(\cdot)$  are the Jacobians of the transformation  $f(\cdot)$  with respect to the orientation parameters and measurement, respectively,

$$J_p(\underline{\hat{p}}, \underline{\hat{r}}) = \left. \frac{\partial f}{\partial \underline{p}} \right|_{\underline{p}=\underline{\hat{p}}, \underline{r}=\underline{\hat{r}}}, \quad (20)$$

$$J_r(\underline{\hat{p}}, \underline{\hat{r}}) = \left. \frac{\partial f}{\partial \underline{r}} \right|_{\underline{p}=\underline{\hat{p}}, \underline{r}=\underline{\hat{r}}}.$$

Taking the expectation of Eq. (19) gives an approximate MMSE estimate of the terrain detection, accounting for all modeled sources of error in the sensing path (Bar-Shalom et al., 2001):

$$\underline{\hat{r}}^{\text{ENU}} = (\hat{E}^{\text{ENU}} \quad \hat{N}^{\text{ENU}} \quad \hat{U}^{\text{ENU}})^T = f(\underline{\hat{p}}, \underline{\hat{r}}) \quad (21)$$

and the estimate has the associated mean squared error matrix

$$P^{\text{ENU}} = J_p Q_p J_p^T + J_r Q_r J_r^T, \quad (22)$$

where  $Q_p$  and  $Q_r$  are the covariance matrices of the orientation parameters' noise  $\delta \underline{p}$  and the measurement's noise  $\delta \underline{r}$ , respectively. Note  $P^{\text{ENU}}$  is effectively an ENU covariance matrix for the LIDAR measurement estimate, taking into account the modeled sources of error in the sensing path.

### 5.2.2. Measurement Association

The second step in the terrain estimation algorithm is to assign each estimated terrain measurement  $\hat{r}^{\text{ENU}}$  to one or more grid cells from which it is likely to have originated. To accomplish this, the probability

that each measurement belongs to each cell must be computed. Because each measurement's mean  $\hat{r}^{\text{ENU}}$  and covariance matrix  $P^{\text{ENU}}$  are available from Eqs. (21) and (22), the joint ENU probability density  $\mathcal{P}_f(E, N, U)$  of the measurement can be approximated as a multivariate Gaussian with vector mean  $\hat{r}^{\text{ENU}}$  and covariance  $P^{\text{ENU}}$ . The probability that the measurement belongs in a particular cell can then be evaluated by marginalizing the density with respect to  $U$  and integrating the resulting EN density  $\mathcal{P}_f(E, N)$  over the area of the cell. This integral can be approximated as the area of a single Riemann square

$$p(\underline{\hat{r}}^{\text{ENU}} \in \text{CELL}) \approx (E_f - E_0)(N_f - N_0) \times \mathcal{P}_f\left[\frac{1}{2}(E_f + E_0), \frac{1}{2}(N_f + N_0)\right] \quad (23)$$

for the cell defined by  $E_0 \leq E \leq E_f$  and  $N_0 \leq N \leq N_f$ . This gives the desired association probability  $p_i$  that the  $i$ th LIDAR measurement estimate  $\hat{r}_i^{\text{ENU}}$  belongs in a particular cell.

### 5.2.3. In-Cell Terrain Measurement Fusion

The final step in the terrain estimation algorithm is to fuse all measurements assigned to a particular cell into an estimate of the elevation distribution within that cell. First, each ENU measurement  $\hat{r}_i^{\text{ENU}} = (\hat{E}_i^{\text{ENU}} \quad \hat{N}_i^{\text{ENU}} \quad \hat{U}_i^{\text{ENU}})^T$  assigned to a cell is transformed into a posterior estimate of the cell's elevation  $\hat{U}_i$  by conditioning the measurement on the EN location of the cell (Bar-Shalom et al., 2001):

$$\hat{U}_i = \hat{U}_i^{\text{ENU}} + (P_i^{\text{ENU}, U})^T (P_i^{\text{ENU}})^{-1} \times \left[ \frac{1}{2} \begin{pmatrix} E_f + E_0 \\ N_f + N_0 \end{pmatrix} - \begin{pmatrix} \hat{E}_i^{\text{ENU}} \\ \hat{N}_i^{\text{ENU}} \end{pmatrix} \right] \quad (24)$$

with associated conditional variance

$$\sigma_{U,i}^2 = P_i^U - (P_i^{\text{ENU}, U})^T (P_i^{\text{ENU}})^{-1} P_i^{\text{ENU}, U}, \quad (25)$$

where the  $i$ th measurement's covariance matrix  $P_i^{\text{ENU}}$  has been divided up into a  $2 \times 2$  block  $P_i^{\text{EN}}$ , a  $1 \times 1$  block  $P_i^U$ , a  $2 \times 1$  block  $(P_i^{\text{ENU}, U})^T$ , and a  $1 \times 2$  block  $P_i^{\text{ENU}, U}$ :

$$P_i^{\text{ENU}} = \begin{bmatrix} P_i^{\text{EN}} & (P_i^{\text{EN},U})^T \\ P_i^{\text{EN},U} & P_i^U \end{bmatrix}. \quad (26)$$

With this, the task of fusing terrain measurements has been reduced to determining the univariate distribution of elevations within each cell. In this problem, each measurement  $\hat{U}_i$  is effectively a terrain detection: an estimate of a piece of the terrain within a cell. From here it is convenient to assume that each of these measurement estimates corresponds to a different patch of terrain within the cell, so that no two measurements occur at precisely the same location. This assumption is justified by the fact that the terrain is continuous, the location of the terrain sensed by each measurement is uncertain, and the sensing platform is moving. Additionally, each terrain measurement is assumed equally likely; that is, there is no *a priori* terrain information. Finally, each cell is assumed to have one correct or “dominant” elevation to be estimated, and that elevation is represented within the set of terrain measurements assigned to the cell. Under these assumptions, the posterior elevation distribution within the cell is a Gaussian sum or “Gaussian mixture” constructed from the elevation estimates (Bar-Shalom et al., 2001; Bishop, 1995):

$$\mathcal{P}(U|\hat{R}^{\text{ENU}}) = \frac{\sum_{i=1}^M p_i \mathcal{N}(\hat{U}_i, \sigma_{\hat{U}_i}^2)}{\sum_{i=1}^M p_i}, \quad (27)$$

where  $\hat{R}^{\text{ENU}}$  is the set of all LIDAR measurements made so far,  $\hat{U}_i$  from equation (24) is the  $i$ th elevation measurement estimate assigned to the cell,  $\sigma_{\hat{U}_i}^2$  from Eq. (25) is its associated conditional variance,  $p_i$  is the probability that the measurement belongs in the cell, and  $M$  is the number of measurement estimates assigned to the cell.

The distribution in Eq. (27) is the desired data-driven elevation distribution in the cell that takes into account all sources of uncertainty within the sensing path. However, the model is in general not computationally feasible, because the Gaussian mixture stored in each cell grows with the number of sensor measurements assigned to that cell. For real-time terrain estimates, an approximate model using a small set of information about each cell is desired: a set of data descriptive enough to be useful but small enough to be computationally feasible. It is

proposed that the mean and variance of the elevation distribution are used. The expected value of the Gaussian mixture (GM) elevation distribution yields an approximate MMSE estimate of the characteristic or dominant elevation of the  $j$ th cell

$$\hat{U}_{\text{GM},j} = \frac{\sum_{i=1}^M p_i \hat{U}_i}{\sum_{i=1}^M p_i} \approx E[U_j | \hat{R}^{\text{ENU}}] \quad (28)$$

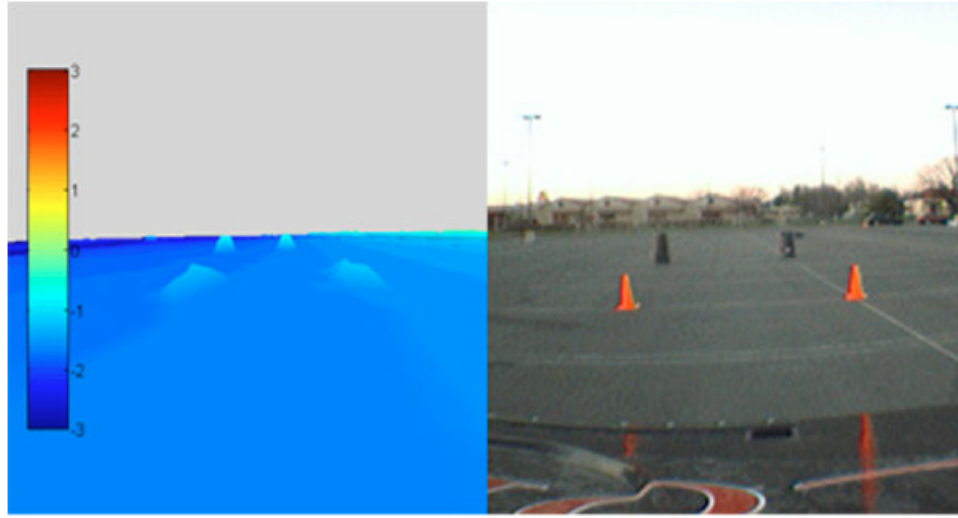
by the linearity of the expectation operator. Note this expectation is conditioned upon all the information available, as the elevation distribution itself is conditioned upon those measurements. Similarly, the second central moment of the Gaussian mixture gives the conditional mean square error of the estimate within the  $j$ th cell

$$\sigma_{\text{GM},j}^2 = \frac{\sum_{i=1}^M p_i (\hat{U}_i^2 + \sigma_{\hat{U}_i}^2)}{\sum_{i=1}^M p_i} - \hat{U}_{\text{GM},j}^2. \quad (29)$$

Equations (28) and (29) give the first two moments of the Gaussian mixture elevation distribution of the  $j$ th cell. Physically, Eq. (28) gives an estimate of the average or characteristic elevation of the cell with respect to the ENU reference origin derived from the available measurements. Mathematically, Eq. (28) is an approximate MMSE estimate of the elevation of the  $j$ th cell when taken with the assumptions discussed above. Equation (29), in contrast, may be interpreted as a measure of the roughness or spread of elevations within the cell, though it also stores information about the confidence of the mean elevation estimate. Equation (29) gives the second central moment of the measured elevations, effectively condensing the cell’s elevation distribution into a Gaussian distribution with first and second moments matching those of the mixture model. The estimates of cell mean and variance can be used to make statistical statements about the elevations in the cell, taking into account all the noise present in the sensing path.

### 5.3. Terrain Estimator Performance

Cornell’s terrain estimation algorithm has several unique advantages over traditional mapping strategies. First, the variance estimate within each cell gives information about the spread of elevations in



**Figure 9.** Sample  $\hat{U}_{GM} + 2\sigma_{GM}$  elevation map (in meters) resulting from the Spider's terrain estimation scheme. Warmer colors represent higher elevations.

that cell, allowing confidence intervals to be created over elevations in that cell. These confidence intervals hold more information than the standard binary obstacle representation of Martin & Moravec (1996), for example, without requiring the additional post-processing and interpolation of elevation-based terrain representations presented, for example, in Lohmann, Koch & Schaeffer (2000) and Arakawa & Krotkov (1992).

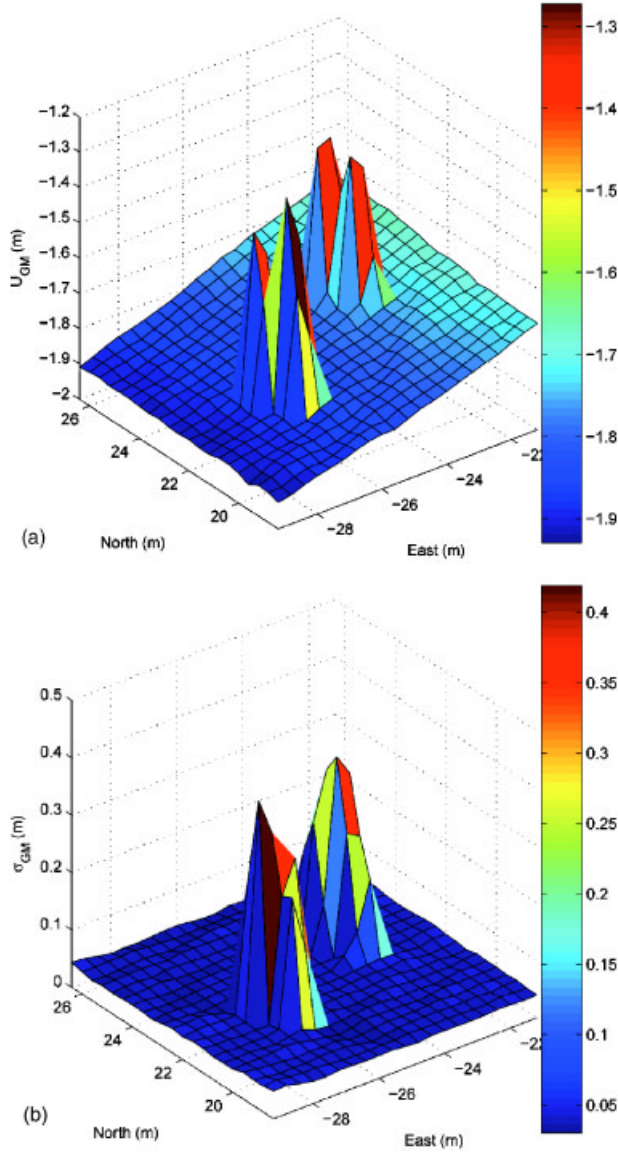
A second advantage of this terrain model is that it can be generated and maintained in real time. Recall from Section 5.2.2 that each measurement estimate is assigned to each cell according to the probability that the measurement lies in that cell. For finite numerical precision, however, only cells near the nominal measurement location derived from  $\hat{r}_i^{ENU}$  will be affected by that measurement estimate. As a result, each measurement estimate need only be applied to cells in a small neighborhood of the nominal measurement. This places limits on the number of cells to which each measurement can be applied, so the computational complexity is reduced to  $O(kN)$  for  $N$  LIDAR measurements, each applied to a maximum of  $k$  terrain cells. Furthermore, if only the first two moments of the elevation distribution are retained, then each measurement can easily be fused with previous measurements. In fact, only the following four quantities are required for each cell:

$$\sum_{i=1}^M p_i, \quad \sum_{i=1}^M p_i \hat{U}_i, \quad \sum_{i=1}^M p_i \hat{U}_i^2, \quad \sum_{i=1}^M p_i \sigma_{U,i}^2$$

where each of these quantities is itself a scalar. Also, because fusing a new measurement with the measurement history only requires knowledge of these four variables, the computational complexity and memory requirements of maintaining each cell are  $O(1)$ . That is, once sensor measurements have been used to update the terrain model, the original measurements can be discarded. The entire terrain map can therefore be maintained without storing a measurement history.

Figure 9 shows a sample terrain map generated from LIDAR measurements as the Spider approaches several objects. In this example, the  $\hat{U}_{GM} + 2\sigma_{GM}$  elevations from Eqs. (28) and (29) (in meters) are plotted relative to an arbitrary ENU origin selected near the test site. These elevations are plotted across a color spectrum, and the axes in Figure 9 are to scale. The  $\hat{U}_{GM}$  elevations for the cells near the 0.8 m tall trash cans are approximately 0.45 m higher than surrounding cells, and the  $\hat{U}_{GM}$  elevations for the cells near the 0.46 m traffic cones are approximately 0.25 m. These low elevation estimates reflect the fact that the cells containing these objects also contain some exposed portions of the ground





**Figure 10.** (a) Final  $\hat{U}_{GM}$  map near the two 0.8 m tall trash cans. (b) Final  $\sigma_{GM}$  map near the two 0.8 m tall trash cans.

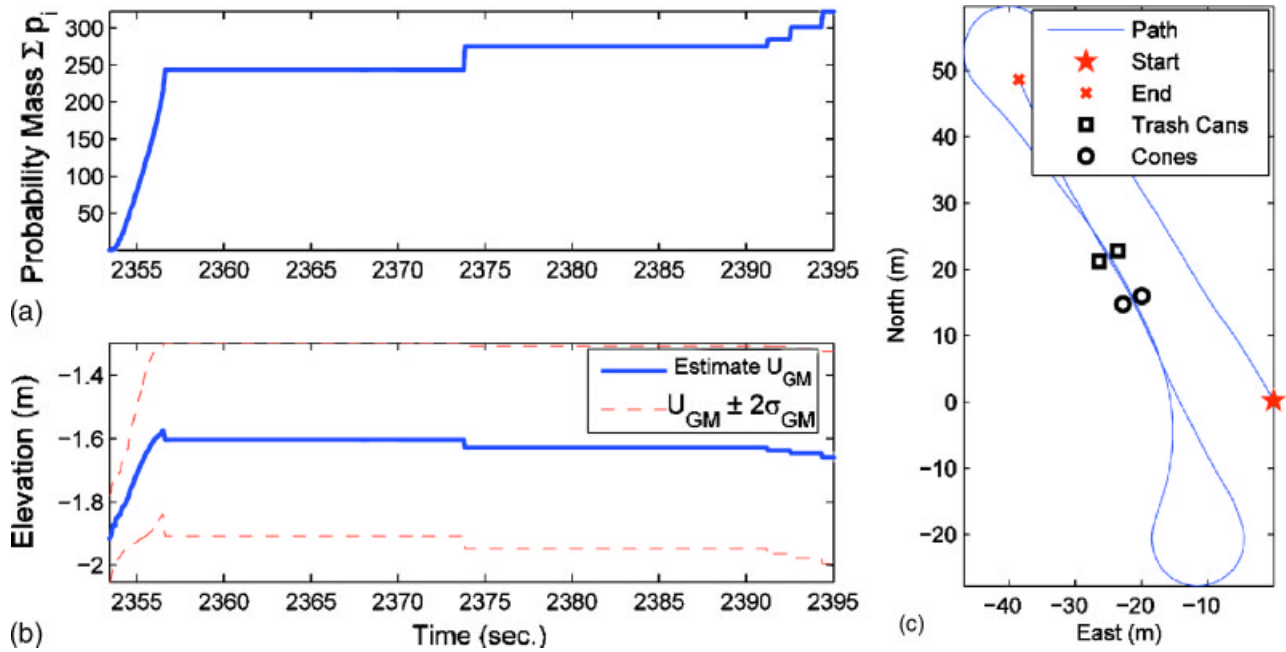
plane. The uncertainties, however, are appropriately large for these cells:  $\sigma_{GM} \approx 0.35$  m for the cells near the trash cans and  $\sigma_{GM} \approx 0.17$  m for the cells near the traffic cones. Figure 10 shows the estimated elevation  $\hat{U}_{GM}$  and estimation uncertainty  $\sigma_{GM}$  near the trash cans in greater detail.

Figures 11(a) and 11(b), in contrast, show the

evolution of the terrain estimate for one cell near one of the trash cans. Figure 11(c) shows that during this test, the Spider passes near the trash cans three times: once with the trash cans in the periphery of the sensors' footprints at  $t \approx 2358$  s, and twice directly between the two trash cans at  $t \approx 2374$  and  $2395$  s. Figure 11(a) shows the total association probability sum  $\sum p_i$  accumulated for the cell over time. Figure 11(b) shows the terrain elevation estimate  $\hat{U}_{GM}$  and the  $\pm 2\sigma_{GM}$  bounds for the same cell. Note that the elevation estimate for the terrain is relative to an arbitrary ENU origin, not the ground plane, as the location of the ground plane is unknown at the outset of the experiment. Figure 12 shows vehicle speed and heading during the test.

The elevation estimate in the particular cell shown in Figure 11(b) fluctuates over time as more measurements are applied to it: some measurements, from the near-vertical face of the trash can, tend to increase the elevation estimate upward. Others, from the surrounding flat ground, tend to decrease the elevation estimate. These different measurements also affect the cell's variance: it is much higher than in surrounding cells, indicating rough terrain or obstacles, and it increases throughout the experiment. Finally, notice that the elevation estimate changes most on the Spider's first pass by the trash can, and that subsequent passes do not cause a substantial increase in total association probability of measurements assigned to that cell. The algorithm is therefore capable of producing accurate estimates in real time, at reasonable speeds, and without revisiting old terrain.

Additional information can be found in the accompanying video files, which show the  $\hat{U}_{GM} + 2\sigma_{GM}$  elevation map generated as the terrain estimation algorithm fuses LIDAR data in real time. The elevations in these video files are in meters, and the colors have the same scale as in Figure 9. Notice that the algorithm produces relatively smooth and dense terrain estimates; this occurs because individual sensor measurements are applied in more than one cell as per Eq. (27). This smoothed and correlated terrain model arises naturally from the estimation algorithm and the continuous posterior distribution of the errors in the sensors. It contrasts sharply with other terrain estimation algorithms that make use of multiple passes or recursion on the terrain estimate to smooth it out, such as those of Olin & Tseng



**Figure 11.** [(a), (b)] Total association probability and  $\hat{U}_{GM} \pm 2\sigma_{GM}$  bounds over time for a particular terrain cell containing a portion of a trash can. (c) Vehicle ground track during real-time terrain experiment. The vehicle intersects the line connecting the trash cans at  $t \approx 2358, 2374$ , and  $2395$  s.

(1991), Arakawa & Krotkov (1992), Hähnel, Burgard & Thrun (2004), and Weingarten & Siegwart (2005).

## 6. GIMBAL AIMING AND CONTROL

As discussed in Section 5.1 and depicted in Figure 8, one of the Spider's terrain sensors is a two-axis actuated gimbal platform supporting a LIDAR unit. This gimballed LIDAR is the Spider's most valuable terrain sensor, as it can scan in any direction. When combined in a feedback loop with the Spider's path planner, the gimballed LIDAR is used to gather data along potential paths as they are generated. In order to point the LIDAR in a desired direction, however, a kinematic relationship must be derived between the actuated gimbal axes and the LIDAR's Earth-referenced orientation. This kinematic relationship can then be used for active gimbal control while simultaneously canceling motion of the vehicle.

To derive this kinematic relationship between gimbal yaw, pitch, and LIDAR orientation, it is first necessary to define a gimbal reference frame. This frame is defined by letting the X axis of the gimbal frame point in the direction of the LIDAR's central

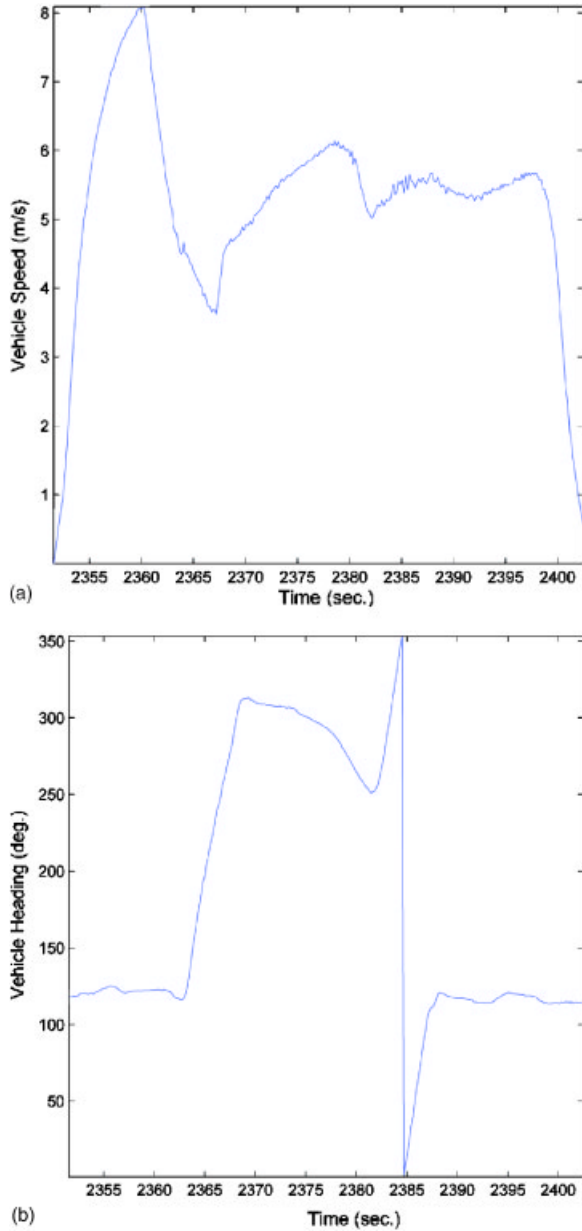
measurement, letting the Y axis be the gimbal pitch axis, and letting the Z axis complete the right-handed coordinate system. In general, the vehicle body frame axes ( $B$ ) may be rotated to the gimbal frame axes ( $G$ ) by applying the following transformations:

1. Rotate by  $\psi_g$ , the gimbal yaw, about the Z axis of the body frame.
2. Rotate by  $\phi_g$ , the gimbal pitch, about the new Y axis.

Next, define the LIDAR pointing direction  $e_l^G$  as the unit vector pointing along the X axis of the gimbal coordinate system. Using the definitions of the gimbal coordinates ( $G$ ) and the vehicle body coordinates ( $B$ ), this unit vector  $e_l$  has the following representation in vehicle body coordinates:

$$e_l^B = \begin{bmatrix} \cos(\psi_g)\cos(\phi_g) \\ \sin(\psi_g)\cos(\phi_g) \\ -\sin(\phi_g) \end{bmatrix}. \quad (30)$$

This unit vector  $e_l^B$  can then be compared to a unit vector  $e_d$  that represents the desired LIDAR pointing di-



**Figure 12.** (a) Vehicle speed during real-time terrain experiment. (b) Vehicle heading during real-time terrain experiment.

rection. The vector  $\underline{e}_d$  may in general be constructed in ENU coordinates as a unit vector that points from the vehicle to some target of interest. When this vector is then expressed in vehicle body coordinates ( $B$ ), define it to have the components  $\underline{e}_d^B = (e_x^B \ e_y^B \ e_z^B)^T$ . In order to point the gimbaled LIDAR in the desired di-

rection  $\underline{e}_d$ , choose  $\psi_g$  and  $\phi_g$  such that  $\underline{e}_I^B = \underline{e}_d^B$ :

$$\begin{bmatrix} \cos(\psi_g)\cos(\phi_g) \\ \sin(\psi_g)\cos(\phi_g) \\ -\sin(\phi_g) \end{bmatrix} = \begin{pmatrix} e_x^B \\ e_y^B \\ e_z^B \end{pmatrix}. \quad (31)$$

With the constraint  $-\pi/2 < \phi_g < \pi/2$ , Eq. (31) has a unique solution

$$\psi_g = \tan^{-1}\left(\frac{e_y^B}{e_x^B}\right),$$

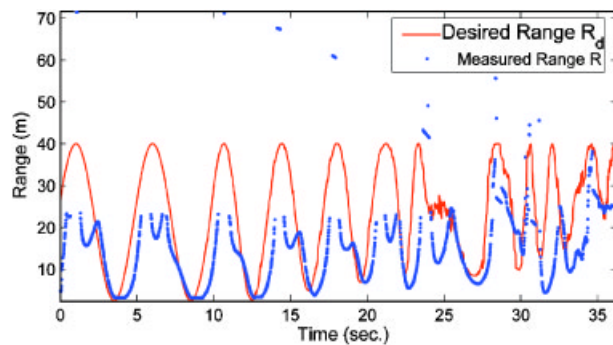
$$\phi_g = \sin^{-1}(-e_z^B). \quad (32)$$

Setting  $\psi_g$  and  $\phi_g$  according to Eq. (32) aims the center of the gimbaled LIDAR in the desired direction  $\underline{e}_d$ . Because this solution accounts for the Spider's position and orientation, it also cancels the effects of vehicle motion.

In addition to canceling the motion of the Spider, this approach is also used by the Spider's path planner to gather terrain data about the path the Spider plans to traverse. The algorithm first walks forward from the Spider along the current planned path to some desired range  $R_d$  to select a point  $(E_C \ N_C \ U_C)^T$  of interest. Because the algorithm may not know the elevation  $U_C$  at  $(E_C \ N_C)^T$ , the Spider's current elevation is used. The desired pointing direction  $\underline{e}_d$  is then computed by normalizing the vector from the LIDAR to this point of interest, and the gimbal control angles are then calculated using Eq. (32). A feedback loop is applied to the point of interest in subsequent iterations to ensure that the central measurements returned by the gimbaled LIDAR are near  $R_d$ :

$$U_C(k+1) = U_C(k) + K_1(R_d - R) - K_2N, \quad (33)$$

where  $U_C(k+1)$  is the elevation of the point of interest at the next iteration,  $U_C(k)$  is the elevation at the current iteration,  $R$  is the range measured at the center of the gimbaled LIDAR,  $N$  is the number of "infinite" ranges reported by the LIDAR, and  $K_1$  and  $K_2$  are control gains. The  $K_1$  term ensures data is gathered at the desired range  $R_d$  by adjusting the commanded elevation until it matches the terrain elevation at the desired range. The  $K_2$  term is added to prevent the gimbaled LIDAR from aiming above the horizon, where



**Figure 13.** Measured and commanded range of the center 20 measurements of the gimbaled LIDAR.

it returns no terrain measurements. The form of the control law in Eq. (33) allows the gimbaled LIDAR to track the top of the terrain at the desired range along the currently planned path.

An example of the feedback loop's tracking capabilities is given in Figure 13, which shows the average of the central 20 ranges  $R$  measured by the gimbaled LIDAR plotted against a sinusoidal commanded range  $R_d$  as the Spider drives on a paved surface. The system tracks the commanded ranges accurately up to 25 m, the approximate range limit of the LIDAR on pavement. Beyond this threshold the LIDAR typically returns ranges of infinity when measuring pavement, so the  $K_2$  term of Eq. (33) causes the gimbal to pitch down. The LIDAR also occasionally receives one or two measurements from an object at a great distance; these objects cause the LIDAR to pitch down rapidly due to the  $K_1$  term. The total effect of the  $K_1$  and  $K_2$  terms cause the gimbaled LIDAR to spend most of its time measuring the dense, informative terrain in front of the vehicle. In contrast, little time is spent aiming above the horizon or gathering sparse data far from the vehicle.

Figure 13 also shows that the tracking loop has a significant time lag between the desired range and the measured range. This lag is not due to actuator dynamics: a first order lag is placed on the controller outputs to create smooth gimbal motion. Although the gimbal actuators are capable of tracking ranges at a higher bandwidth, it was found that higher frequency tracking induced vibrations in the sensor platform that produced invalid measurements from the other LIDAR units.

## 7. ROBUST GRID-BASED PATH PLANNING

One of the most difficult tasks for any autonomous ground vehicle is integrating vehicle capabilities, actuator states, position and orientation information, and sensed terrain maps to plan a path to traverse. The time-limited format of the Grand Challenge makes the task especially difficult, because critical path decisions must be made on-the-fly in unknown obstacle fields at average speeds of 7 m/s or more. To function under these constraints, a good path planner must have an efficient data representation, and it must be able to strike a balance between path quality and reaction time.

These two design attributes, data representation and search preference, generally categorize the behavior of path planners. The first attribute, data representation, describes how a path planner ranks potential paths. Existing path planners generally fall between two extremes: cost-based representations, and constraint-based representations. Cost-based representations, such as the one used in Stentz (1994), integrate all environment information into a scalar cost value for each potential path. Constraint-based representations, such as the one used in Spenko, Iagnemma & Dubowsky (2004), store environment information as a set of constraints that must be satisfied for a path to be selectable. The other attribute, search preference, describes the types of paths a planner investigates. Global planners, such as the approach presented in Stentz (1994), choose paths over the entire course to be traversed. Local or reactive planners, in contrast, may select circular arcs or cubic splines that are only a few meters in length (Rosenblatt, 1997; Urmson et al., 2004).

Cornell's Grand Challenge entry lies in between the extremes for each of these two attributes. Terrain information is represented as a scalar cost, while vehicle capabilities, position, and orientation are combined in a physics-based skidding constraint. Paths considered are cubic splines approximately 15–80 m in length; they direct the Spider from its current location to a point that smoothly connects with the nominal route, beyond the Spider's sensor horizon. In practice these design choices allow the Spider's path planner to react quickly to obstacles, yet the paths selected are smooth and human-like.

### 7.1. Cost Representation for Path Planning

The terrain estimation algorithm from Section 5 provides two pieces of environment information for



path planning: elevation estimates  $\mu_{GM}$ , and uncertainties  $\sigma_{GM}$  for each  $40\text{ cm} \times 40\text{ cm}$  grid cell of terrain. In the Cornell planner, these terrain data are used to assign a measure of traversability, in the form of a scalar cost, for each cell. The grid-based storage scheme and dense elevation and uncertainty estimates of the terrain estimation algorithm are ideal for evaluating the traversability of pieces of terrain according to the Spider's physical capabilities. In particular, the elevation estimates can be used to calculate finite difference gradients  $\nabla\mu_{GM,i}$  for the  $i$ th cell

$$|\nabla\mu_{GM,i}| \approx \frac{\max_{j \in N(i)} (|\mu_{GM,i} - \mu_{GM,j}|)}{dx}, \quad (34)$$

where  $N(i)$  is the set of cells neighboring cell  $i$ , and  $dx$  is the width of the cell. These cell gradients are then converted to a gradient cost  $C_{\nabla}$  for the  $i$ th cell based on the Spider's physical limits

$$C_{\nabla,i} = 32\,000 \cdot \left( \frac{|\nabla\mu_{GM,i}|}{\nabla U_{\max}} \right)^2, \quad (35)$$

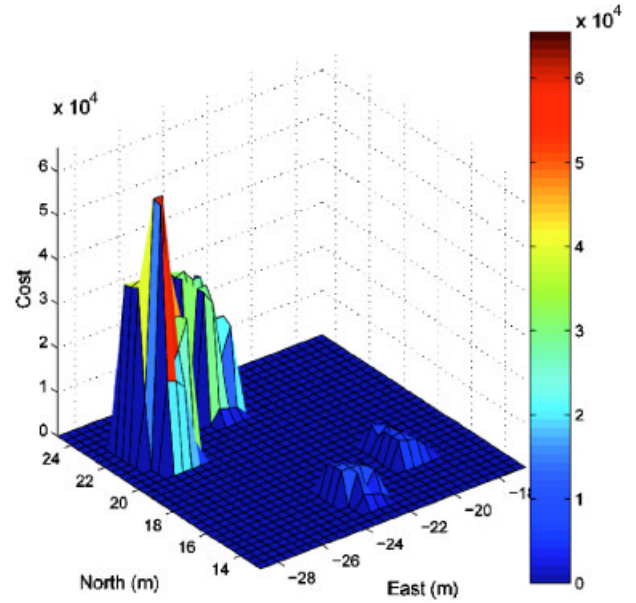
where  $\nabla U_{\max} = 1.095$  is the Spider's maximum safe traversable gradient, calculated as an elevation change of  $0.43\text{ m}$  across a single  $0.4\text{ m}$  cell. The gradient cost in each cell is also multiplied by a scaling factor  $32\,000$  and converted to an integer to avoid additional floating point operations during path evaluations.

The uncertainty in each cell's elevation is also used to calculate a component of cost based on the Spider's physical limits

$$C_{\sigma,i} = 2000 \left( \frac{\sigma_{GM,i}^2}{\sigma_{\max}^2} \right)^2, \quad (36)$$

where  $\sigma_{\max} = 0.2\text{ m}$  is chosen so the  $2\sigma_{GM}$  elevation uncertainty in each cell is penalized if it exceeds the Spider's maximum traversable obstacle height of  $40\text{ cm}$ . As with the gradient cost, the uncertainty cost is also multiplied by a scaling factor  $2000$  and converted to an integer.

Figure 14 shows a cost map calculated from the sum of the gradient and uncertainty costs of the elevation estimates generated in Figure 9. This sum is used as a measure of the traversability of each cell, with less traversable or more uncertain cells having

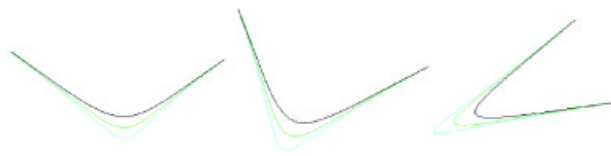


**Figure 14.** Sample cost map generated from the parking lot elevation estimates from Figure 9 near the large trash cans and smaller traffic cones. Warmer colors indicate higher cost.

amplified costs. In addition, the quadratic form of Eqs. (35) and (36) help attune gradients and uncertainty due to sensor noise. In the Cornell planner, these costs are used both to eliminate intraversable paths and to establish preferences if multiple paths are traversable.

## 7.2. Searching Preference in Path Planning: The Cubic Planner

While grid-based terrain estimates fit naturally into a cost framework, the remainder of the Spider's environment, including hardware constraints and vehicle position, velocity, and orientation, are not well represented as costs. It is possible, for example, to penalize paths that violate vehicle constraints, turn sharply, or decelerate rapidly. These cost penalties do not guarantee compliance, however, so paths chosen to minimize this cost may not be physically realizable. Furthermore, with the cost function changing as fast as the vehicle's position, velocity, and orientation, the lowest-cost path may change drastically from one planning iteration to the next. In high-speed, high-dynamics tasks like the Grand



**Figure 15.** Candidate cubic paths generated for various turns. The paths are constructed from pairs of control points at the beginning and end of the path. Families of cubics are generated by varying the spacing of the two center control points.

Challenge, however, it is more desirable to find near-optimal local solutions that are consistent with vehicle dynamics. These points motivate Cornell's planner, which generates path segments from a small set of cubic splines.

Cornell's cubic planner uses several simplifying assumptions to generate a rich variety of paths using only three free parameters. First, it is noted that the Grand Challenge course is often narrow with respect to vehicle size, allowing little lateral motion. Second, the course must be drivable by commercial vehicles, so there is guaranteed to be a path from the start of the course to its end (DARPA, 2004). These two assumptions impose basic constraints on the radius of curvature and space required by candidate paths. These constraints can be checked in real time with approximations to vehicle physics, allowing many candidate paths to be eliminated without lengthy numerical simulation. As a result, this verification-based technique can search among complicated paths that are more likely to be traversable over long distances. Figure 15 shows some of these candidate paths considered by the Cornell cubic planner.

### 7.2.1. Spline Construction

The Cornell path planner uses two-dimensional cubic Bézier curves to represent its candidate paths. Two-dimensional cubic Bézier curves are defined by four 'control points'  $\{A, B, C, D\}$ ; see, for example, Weisstein (1999). Useful properties of the cubic Bézier for path planning are as follows.

1. The Bézier always passes through its first control point  $A$ . This point is set to the center of the Spider's back axle to represent the current planning origin.
2. The Bézier always ends at its last control

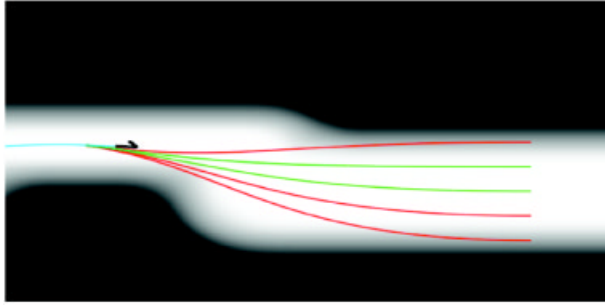
point  $D$ . Point  $D$  therefore represents the end of the cubic spline, and it should be inside the course boundary at all times.

3. The Bézier is always tangent to the line segment  $\overline{AB}$  at point  $A$ . Line segment  $\overline{AB}$  is therefore set in the current direction of motion, according to the Spider's yaw angle  $\psi$ . Control point  $B$  is therefore fully defined by a scalar distance: the forward distance from  $A$  in the direction of travel.
4. The Bézier is also always tangent to the line segment  $\overline{CD}$  at point  $D$ . The Cornell planner chooses the line segment  $\overline{CD}$  such that it is parallel to the DARPA-specified path at the end of the cubic Bézier. As a result, control point  $C$ , like  $B$ , is entirely specified by a single parameter: its distance from  $D$  toward the vehicle.

With these constraints, the cubic search space is fully defined by five pieces of information: the vehicle's current location, its current yaw angle  $\psi$ , the location of the terminal point  $D$ , the offset of control point  $B$  from  $A$  in the direction of vehicle travel, and the offset of the control point  $C$  from point  $D$  backward toward the vehicle. Note that of these five parameters, only the two scalar offset lengths and the path termination point are free parameters.

In Cornell's implementation, the location of the terminal point  $D$  is determined by selecting a look-ahead distance along the nominal DARPA route based on desired vehicle velocity. The look-ahead distance is set at a minimum distance of 15 m so the vehicle does not consider a turn it cannot achieve. The terminal point  $D$  is then generated by projecting the vehicle's location onto the nominal DARPA route and walking forward by the look-ahead distance. In this implementation, four additional terminal points are also created from equidistant lateral projections off the original terminal point. An example of paths generated to each of these five lateral offsets is shown in Figure 16.

In addition, more complicated paths can be created by varying the locations of points  $B$  and  $C$  by changing the "offset lengths": the lengths of segments  $\overline{AB}$  and  $\overline{CD}$ . Intuitively, changing the offset length of the control point  $B$  determines when and how severely the vehicle reacts as it approaches a turn, as shown in Figure 17(a). In contrast, Figure 17(b) shows that changing the offset length of the control point  $C$  affects the duration of a turn. In the



**Figure 16.** The five lateral offsets considered for a single terminal point.

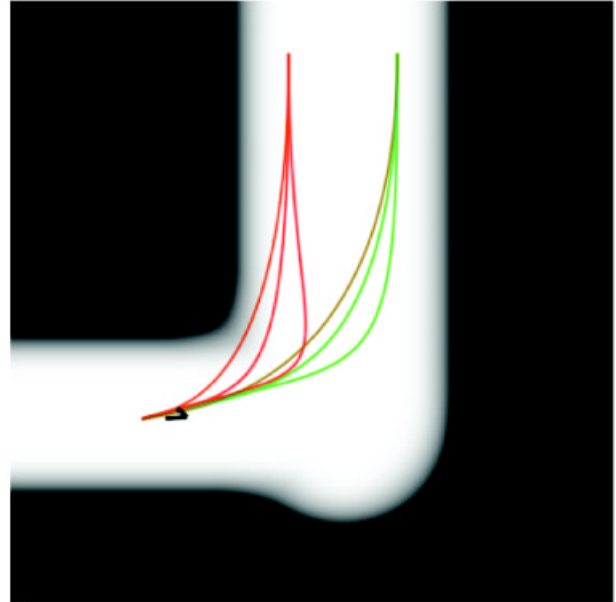
Cornell planner, three *B* offsets, four *C* offsets, and five terminal points are considered at each planning cycle, for a total of 60 paths.

### 7.2.2. Path Evaluation and Selection

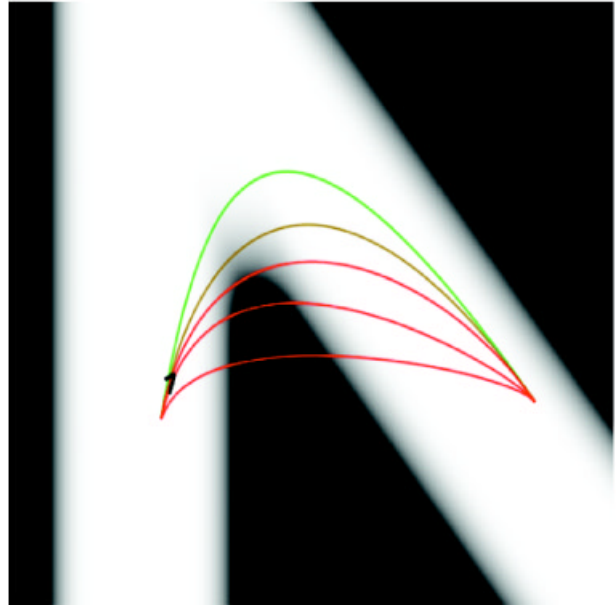
In order to select a final path, the planner determines the best candidate spline in a thresholding step and an ordering step. The first step, thresholding, applies cost and physics-based constraints to eliminate infeasible paths. The planner applies cost constraints by walking each candidate spline in the cost grid, computing a line integral  $C_{LI}$  of cost and eliminating any path traversing impassable obstacles. For the remaining candidates, the planner evaluates four potential vehicle speeds  $v_{des}$ : emergency deceleration ( $v_{des}=v_{cur}-6$  m/s), gradual deceleration ( $v_{des}=v_{cur}-3$  m/s), no change ( $v_{des}=v_{cur}$ ), and gradual acceleration ( $v_{des}=v_{cur}+3$  m/s). The paths and speeds are then evaluated using a lateral skidding test, which imposes a minimum curvature constraint based on speed (Spenko et al., 2004):

$$\min(R_c) > \frac{v_{des}^2}{\mu g}, \quad (37)$$

where  $\min(R_c)$  is the smallest radius of curvature along the path,  $\mu$  is the coefficient of friction, and  $g$  is the acceleration of gravity. Although Eq. (37) approximates lateral skidding assuming constant vehicle speed and turning angle, it is inexpensive to test. In particular, the path's radius of curvature can be evaluated in each grid cell it crosses at the same time the cost line integral is calculated, so the computational expense of applying both cost and



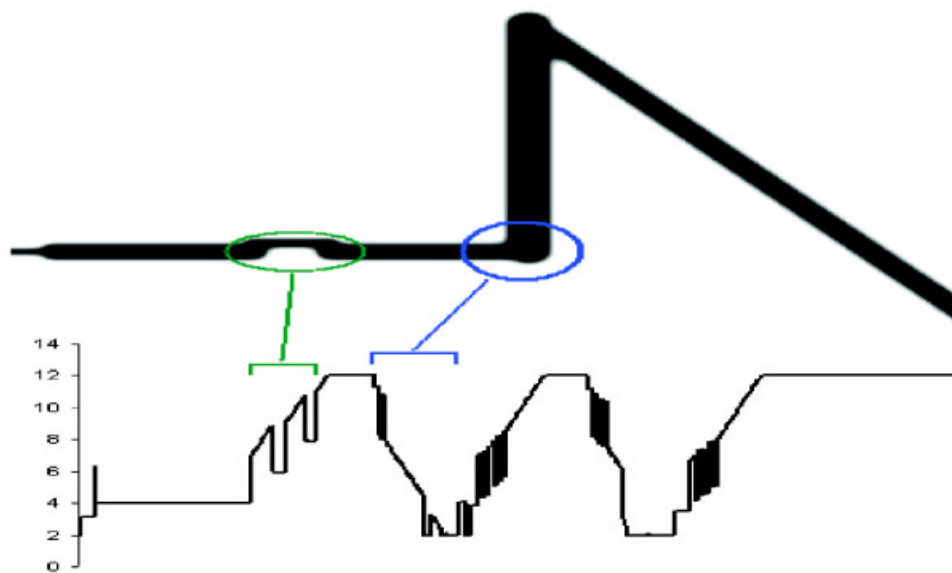
(a)



(b)

**Figure 17.** (a) Variations in the *B* offset affect the lag in the start of a turn. (b) Variations in the *C* offset affect the duration of a turn.

physics-based constraints grows linearly with the length of the path. In practice, the 60 paths and four speeds evaluated in each iteration of the Cornell planner are completed in less than 40 ms on the path



**Figure 18.** Speed selection on a simulated course. The bottom plot shows desired vehicle speed selected by the path planner, in m/s. Note how the Spider decelerates slowly for the soft turn in the beginning and faster for the two sharp turns later on.

planning computer equipped with dual 1.8 GHz AMD Opteron 244 processors and Windows Server 2003.

The skidding constraint of Eq. (37) is used to remove paths that are intraversable due to sharp turns. It also affects the Spider's macroscopic preference toward smoother, straighter, and more human-like paths. The Cornell implementation uses a conservative  $\mu=0.15$ , for example, to discourage last-minute turns to avoid obstacles.

After all intraversable paths have been eliminated, the final selection is made by optimizing a performance metric over the remaining traversable paths. First, the Cornell planner retains only the fastest traversable speed for each cubic path, as finishing time is most important when all paths are traversable. The remaining paths are then ranked according to a weighted combination  $C_{LI}+600\Phi_{LI}$  of total cost  $C_{LI}$  and total steering effort  $\Phi_{LI}$ , calculated during the cost line integral as the one-norm of changes in steering wheel angle from cell to cell along the path. The final path chosen minimizes this weighted combination, representing a balance between avoiding small changes in cost and preferring paths with less steering effort. If no paths are deemed traversable, the Cornell planner chooses the

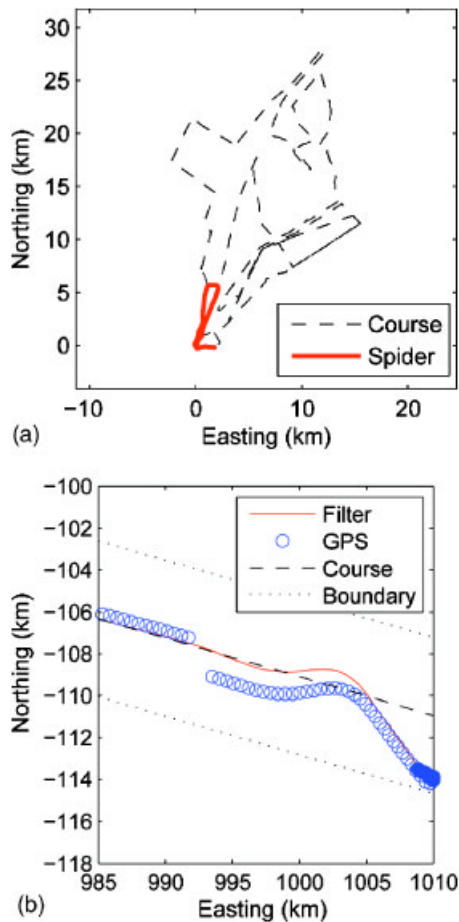
path with the least worst cost while performing a sharp deceleration. This contingency behavior permits the Spider to gather more terrain data, so a better path can be found in the next planning cycle.

Examples of the path planner's final speed selections are shown in Figure 18 for a simulated course. Notice that the planner slows the vehicle down while taking turns, picking up speed as the vehicle exits the turn. Although speeds are tested in discrete 3 m/s increments, Figure 18 shows that they form a feedback loop with the Spider. Sudden speed changes are effectively smoothed out by the 10 Hz replan rate of the path planner and the lag in the Spider's engine.

## 8. GRAND CHALLENGE PERFORMANCE AND FAILURE ANALYSIS

The Spider began the 131.8-mile Grand Challenge course along with the other competitors in Primm, Nevada, in the morning of October 8, 2005. Team Cornell faced several mechanical difficulties just in bringing the Spider to the starting line, including a failing torque converter and a generator that broke its bearings the day before. The Spider made a successful





**Figure 19.** (a) Grand Challenge course, with the Spider's final progress marked. (b) Magnified view of the Spider's point of failure.

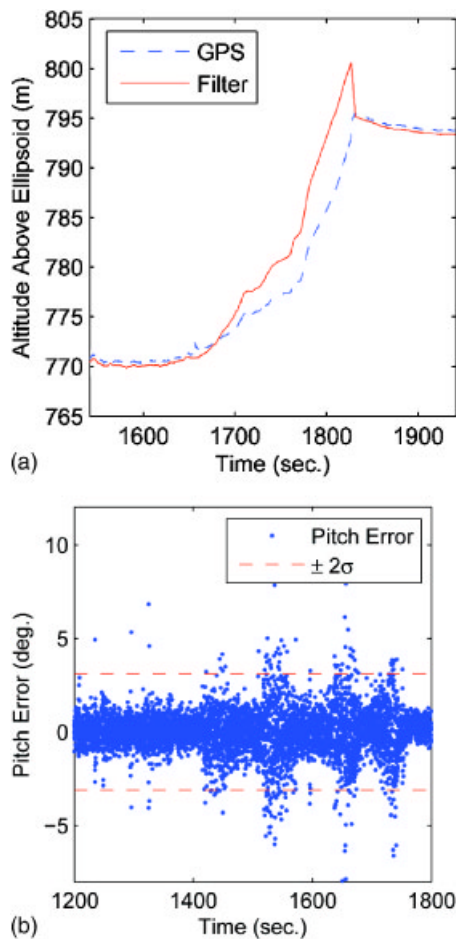
start despite the mechanical difficulties, traveling 9.0 miles before hitting a concrete wall on an overpass. The vehicle was subsequently stopped by DARPA officials. Figure 19(a) shows the total Grand Challenge course, with the Spider's path marked.

Figure 19(b) shows a magnified view of the Grand Challenge course and the Spider's path as it drifted from the center of DARPA's route into the concrete wall. This reveals two odd features of the Spider's final moments in the race. First, the solution reported by the GPS receiver experienced a jump of approximately 2 m, and thereafter it reported an apparent error of more than 1 m until the Spider hit the wall. This offset is evident from the fact that the Spider's filtered position closely followed the center of

DARPA's route, which was confirmed by the GPS solution prior to the anomaly. A further inspection of race data reveals that the GPS jump was due to reacquisition of the OmniSTAR HP signal, which the Spider had lost approximately 175 s earlier. Rate limiting was in effect, however, so the Spider did not swerve to follow the incorrect HP signal. In fact, the Spider turned gently as the rate-limited filter converged to the HP signal.

The second anomaly at the Spider's point of failure is the fact that it chose to turn into the wall. The only explanation for such behavior is that the Spider detected an obstacle in front of it, though postrace investigation revealed no obstacles on the overpass. This behavior suggests obstacle detection errors, and Figure 20 reinforces that conclusion. Figure 20(a) shows that the Spider developed nearly 8 m of error in its altitude estimate over the last few minutes leading up to its decision to hit the wall. Figure 20(b) shows the likely cause with a comparison of the attitude estimator's pitch estimate to pitch derived from GPS velocity. The difference between these two signals shows four neighboring instances just prior to the Spider's failure during which pitch error consistently strayed outside the  $\pm 2\sigma$  line for a short period of time: at 1440, 1540, 1655, and 1735 s into the race. The first two of these instances occurred in the presence of the OmniSTAR HP signal, while the last two occurred after the Spider lost the signal. The fact that these four instances are spaced so closely together suggests the pitch estimate became incorrect during this time interval. This conclusion also explains the Spider's decision to stray from the center of the DARPA route, because pitch errors would cause the Spider to localize LIDAR measurements incorrectly: even flat terrain in the center of the route would appear as obstacles. This problem could have been mitigated either through online hypothesis testing of GPS measurements as in Sukkarieh et al. (1999), online estimator health monitoring as in Bar-Shalom et al. (2001), or by implementing the mapping algorithm of Section 5 relative to the vehicle and independent of absolute position. Unfortunately, the problem did not manifest itself with such severity until the Grand Challenge itself, where it could not be repaired.

The location of the Spider's failure was unfortunate. Reverse driving was not implemented due to time constraints, so the Spider could not recover from its collision. Instead, it was disabled as it struggled to turn free from the wall. If it had chosen to turn in a place where there were no concrete walls, it would



**Figure 20.** (a) Several minutes before it failed in the Grand Challenge, the Spider developed significant errors in its altitude estimate. (b) The altitude errors are believed to have been caused by four instances of significant pitch error at  $t=1440$ ,  $1540$ ,  $1655$ , and  $1735$  s into the race, just prior to the Spider's failure.

have recovered without problem. Even if it strayed from the course boundary, the DARPA officials may have let it continue.

## 9. CONCLUSION

This paper has presented a detailed overview of Cornell University's entry in the 2005 DARPA Grand Challenge. Cornell's approach to the Grand Challenge divided the problem into four main tasks: designing and building a robust platform, localizing the

platform, sensing its environment, and navigating through that environment. This approach had the benefit that each subsystem could be developed independently once the information flow between them was defined.

Cornell's entry in the Grand Challenge was based upon a Spider Light Strike Vehicle, a robust military platform able to tolerate mistakes made by its autonomous components. The platform was localized with dual square root information filters for attitude and position/velocity that fused separate inertial and navigation sensors. This filtering scheme was shown to be as accurate as many of the more expensive off-the-shelf positioning systems. A rate limiting scheme was also presented to handle GPS signal loss and reacquisition, two difficult situations for path tracking and obstacle detection in autonomous ground vehicles. A novel terrain estimation algorithm was also presented to combine the Spider's three laser rangefinders into accurate terrain estimates and their uncertainties. This terrain estimate provided a real-time statistical interpretation of the Spider's environment, permitting robust path planning without heuristic obstacle identification. A method for creating a feedback loop between these terrain estimates and the path planner was presented, and the feedback loop was shown to be able to gather terrain data at the desired range along the planned path. A cubic spline-based path planner was also presented to utilize these dense terrain estimates to generate paths consistent with the physical capabilities of the Spider. It was shown that simple path constraints based on skidding, radius of curvature, and terrain gradient macroscopically affected the final path chosen, making the resulting path smoother and more human-like.

Cornell's Spider was one of the 195 original entrants into the Grand Challenge, and one of only 23 to make it to the starting line of the final Grand Challenge event. Despite experimental validation of each subsystem and the fully integrated vehicle, the Spider's performance in the Grand Challenge was disappointing. Last-minute generator and torque converter failures tested the reliability of the system, while faulty navigational estimates in a highly constrained portion of the course ultimately brought about the Spider's failure in the Grand Challenge. These failures stress the importance of additional on-line health monitoring and contingency behaviors that would have been added to the Spider if there were more time.

## ACKNOWLEDGMENTS

The authors acknowledge the hard work and dedication of all members of Team Cornell from the summer of 2004 through the spring of 2006. The authors also thank the sponsors of Team Cornell, especially Singapore Technologies Kinetics, for their support of this research.

## REFERENCES

- Arakawa, K., & Krotkov, E. (1992) Fractal surface reconstruction with uncertainty estimation: Modeling natural terrain (Tech. Rep. No. CMU-CS-92-194). Pittsburgh: School of Computer Science, Carnegie Mellon University.
- Bar-Shalom, Y., Rong Li, X., & Kirubarajan, T. (2001). Estimation with applications to tracking and navigation: Theory, algorithms and software. New York: Wiley.
- Battin, R. (1999). An introduction to the mathematics and methods of astrodynamics (rev. ed.). Reston: American Institute of Aeronautics and Astronautics.
- Bierman, G. (1977). Factorization methods for discrete sequential estimation. New York: Academic.
- Bishop, C. (1995). Neural networks for pattern recognition. Oxford: Oxford University Press.
- DARPA. (2004). Darpa Grand Challenge 2005 rules. Defense Advanced Research Projects Agency. Retrieved from [http://www.darpa.mil/grandchallenge05/Rules\\_8oct04.pdf](http://www.darpa.mil/grandchallenge05/Rules_8oct04.pdf).
- Franklin, G., Powell, D., & Emami-Naeini, A. (2002). Feedback control of dynamic systems (4th ed.). Upper Saddle River, NJ: Prentice-Hall.
- Hähnel, D., Burgard, W., & Thrun, S. (2004). Learning compact 3d models of indoor and outdoor environments with a mobile robot. Elsevier Science Special Issue Eurobot '01 (pp. 1–16). Retrieved from <http://www.informatik.uni-freiburg.de/burgard/postscripts/haehnel.ras-03.pdf>.
- Huising, E., & Pereira, L. (1998). Errors and accuracy estimates of laser data acquired by various laser scanning systems for topographic applications. ISPRS Journal of Photogrammetry and Remote Sensing, 53, 245–261.
- Lohmann, P., Koch, A., & Schaeffer, M. (2000). Approaches to the filtering of laser scanner data. International archives of photogrammetry, remote sensing and spatial information sciences, 33 (Working Group III-2).
- Martin, M., & Moravec, H. (1996). Robot evidence grids (Tech. Rep. No. CMU-RI-TR-96-06). Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University.
- Moon, F. (1998). Applied dynamics with application to multibody and mechatronic systems. New York: Wiley.
- Murray, R., Li, Z., & Sastry, S. (1994). A mathematical introduction to robotic manipulation. Boca Raton, FL: CRC Press.
- Northrop Grumman Navigation Systems Division. (2000). LN-200 fiber optic inertial measurement unit (Brochure No. 22939/02-06/2000/Crawford). Retrieved from <http://www.nsd.es.northropgrumman.com/Html/LN-200/brochures/ln200.pdf>.
- Ohlmeyer, E., Pepitone, T., Miller, B., Malyevac, D., Bibel, J., & Evans, A. (1997, August). GPS-aided navigation system requirements for smart munitions and guided missiles. In AIAA guidance, navigation, and control conference collection of technical papers (pp. 954–968). Reston: American Institute of Aeronautics and Astronautics.
- Olin, K., & Tseng, D. (1991). Autonomous cross-country navigation: An integrated perception and planning system. IEEE Expert: Intelligent Systems and Their Applications, 6(4), 16–30.
- Rosenblatt, J. (1997). Damn: A distributed architecture for mobile navigation. Unpublished Ph.D. thesis, The Robotics Institute, Carnegie Mellon University.
- Savage, P. (1998). Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. Journal of Guidance, Control, and Dynamics, 21(1), 19–28.
- SICK. (2003). LMS 200, LMS 211, LMS 220, LMS 221, LMS 291 laser measurement systems technical description (Technical Description No. 8 008 970/06-2003).
- Singapore Technologies Kinetics. (2006). Spider lsv 1.04 (Brochure). Retrieved from <http://www.stengg.com/upload/306MJIVTWhk0eRIE2kH.pdf>.
- Spenko, M., Iagnemma, K., & Dubowsky, S. (2004). High speed hazard avoidance for mobile robots in rough terrain. In G. Gerhart, C. Shoemaker, & D. Gage (Eds.), Proceedings of the SPIE (Vol. 5422, pp. 439–450). Bellingham: The International Society for Optical Engineering.
- Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 3310–3317). Los Alamitos: Institute of Electrical and Electronic Engineers.
- Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Transactions on Robotics and Automation, 15(3), 572–578.
- Team Cornell. (2005). Technical review of Team Cornell's Spider: DARPA Grand Challenge 2005 (Technical Review). Retrieved from <http://www.darpa.mil/grandchallenge05/TechPapers/TeamCornell.pdf>.
- Thrun, S. (2002). Robotic mapping: A survey (Tech. Rep. No. CMU-CS-02-111). Pittsburgh: School of Computer Science, Carnegie Mellon University.
- Triantafyllou, M., & Hover, F. (2004). Maneuvering and control of marine vehicles (Maneuvering and Control of Surface and Underwater Vehicles OpenCourseWare Website). Retrieved from <http://ocw.mit.edu/NR/rdonlyres/Ocean-Engineering/13-49Fall-2004/C8072E70-A4F4-466A-98DF-A3EC2865A835/0/lec1.pdf>.
- Trimble. (2004). Aggps 252 receiver user guide version 1.00, revision A (Technical Manual).
- Urmson, C., Anhalt, J., Clark, M., Galatali, T., Gonzales, J., Gowdy, J., et al. (2004). High speed navigation of unrehearsed terrain: Red Team technology for Grand

- Challenge 2004 (Tech. Rep. No. CMU-RI-TR-04-37). Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University.
- Weingarten, J., & Siegwart, R. (2005). EKF-based 3d slam for structured environment reconstruction. In Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (pp. 3834–3839). Los Alamitos: Institute of Electrical and Electronic Engineers.
- Weisstein, E. (1999). Bézier curve. (MathWorld—A Wolfram Web Resource). Retrieved from <http://www.mathworld.wolfram.com/BezierCurve.html>.