# Technical Review of Team Cornell's Spider
# DARPA Grand Challenge 2005

Team Leader: Ephrahim Garcia
Contact: Ephrahim Garcia, Brian Schimpf, or Team Cornell
eg84@cornell.edu, bws22@cornell.edu, contact@dgc.cornell.edu
(607) 255-4366, (585) 732-3128, (607) 330-2764

Sibley School of Mechanical and Aerospace Engineering
Cornell University
105 Upson Hall
Ithaca, NY 14853

## Abstract

The Cornell DARPA Grand Challenge entry is based on a rugged light strike vehicle more than capable of handling the hazardous conditions of the desert. The solution features a rigorous sensor fusion algorithm based on mixture model density functions and linearized error transformations, Bayesian-optimal inertial navigation via square root information filter, global and local path planners with internal faster-than-real-time vehicle dynamics simulations, and a high level situational awareness system to deal with adverse circumstances not handled by the path planner. These systems and additional critical components in the Cornell entry comprise a robust and natural solution to the Grand Challenge.

## Introduction

The Cornell University DARPA Grand Challenge Team is a group of 30 highly motivated students and faculty at Cornell. Our team was formed in the spring of 2004 with the goal of developing a fully autonomous ground vehicle for competition in the 2005 DARPA Grand Challenge. Cornell engineers are drawn to difficult problems like the Grand Challenge, and we look forward to this unique opportunity to test our ingenuity. The real-world engineering experience this contest offers is unparalleled at most universities and unequaled in any classroom.

Cornell has a long history of successful project teams such as the Formula SAE, Robocup and the Autonomous Underwater Vehicle team, giving us an excellent base of experienced students and a winning reputation. Despite the fact that some competing teams have a one-year head start on us in the Grand Challenge, Cornell engineers have the skills it will take to win. Although no team came close to finishing the course last year, many lessons were learned from the successes and failures of last year's entrants, and unexpected problems became apparent. In entering the 2005 Grand Challenge, we have the added benefit of being able to observe these deficiencies so that we may ensure that our vehicle will not be defeated by the same design flaws.

## Vehicle

The base platform of Cornell's Grand Challenge entry is a Spider Light Strike Vehicle manufactured and donated by Singapore Technologies Kinetics. The Spider is combat-proven and designed to military specifications, so it is more capable in off-road desert conditions than a commercial SUV or truck. Its heavy-gage tube frame chassis provides strength and durability, and the undercarriage of the vehicle is protected by a full-length skidplate. The suspension is fully independent, with two shocks at each front wheel and three at each rear. The vehicle's low center of mass combined with intelligent suspension geometry yields an extremely stable platform, and 35-inch tires and high control arms provided ample ground clearance. A VM Motori 2.8L common rail diesel engine powers the vehicle, outputting a maximum of 163 horsepower and 295 ft-lbs of torque. Figure 1 shows the Spider and its critical systems.

Power for the vehicle's electronics is provided by an Onan CMM 5500 EFI fuel-injected gasoline generator. It outputs 5.5 kW peak, and digital voltage regulator provides clean power for sensitive electronics. Custom aluminum intake and exhaust ducting has been added to ensure an adequate supply of air to the generator for combustion and cooling. Custom dual gas tanks were fabricated and installed in the floor of the Spider to provide more than fourteen hours of continuous full-load operation.

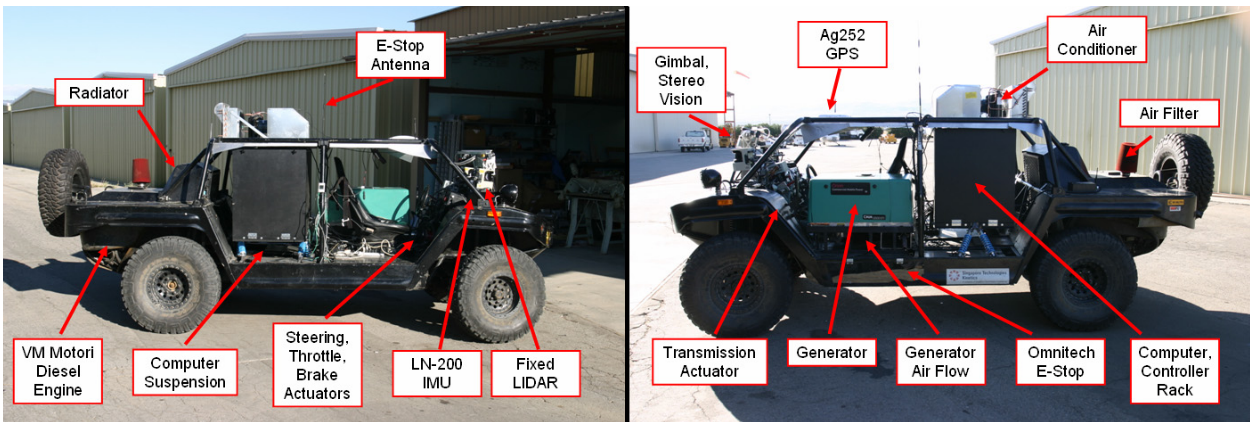


**Figure 1 - Right and left views of the Spider, with critical systems marked.**

## Hardware Interface Layer

Cornell's vehicle features the Actuation Controller, a robust actuation system that is tightly integrated with the vehicle's factory components. A Motorola HCS12 16 bit microcontroller lies at the heart of this unit and uses a CAN (Controller Area Network) Bus to communicate with high level artificial intelligence and low level motor controllers. The Actuation Controller manages the speed of the vehicle with two proportional, integral, derivative (PID) controllers; see, for example, Ref. [Franklin02]. One of these PID controllers is designed for accelerations, and the other is designed for deceleration. Dual sets of integrators, one fast and one slow, are used in these PID controllers to keep integrator windup low while ensuring the vehicle will still apply full throttle if it is stuck. Additionally, the Actuation Controller handles vehicle steering via an external feedback loop with a simple PID controller running at 100 Hz. The closed loop system can steer from lock to lock in less than 0.75 seconds, making the vehicle extremely responsive to artificial intelligence commands. The Spider's three-speed semi-automatic clutchless transmission is also fully actuated for both forward and reverse operation. Several shiftschedules have been created to deal with road conditions such as normal operation, hill-descent, and high-speed driving.

Tremendous consideration has been given to the overall safety of the system. The car is outfitted with three exterior E-Stop switches and two remote E-Stop transmitters: the DARPA issued E-Stop and a custom unit used during testing. The E-Stop system is designed to disable the engine and apply full braking force using a linear actuators mounted to the brake. This brake actuator is not back-drivable, so no power is required to keep the brakes engaged. The E-Stop also disengages the transmission and steering control.

## Inertial Navigation

In order to transform sensor readings from mere ranges to absolute positions useful for path planning, the location and orientation of the vehicle's sensors must be known. For this task Team Cornell fuses three sensors into a smooth Bayesian optimal estimate of the vehicle's position and attitude: a Litton LN-200 Inertial Measurement Unit (IMU), a Trimble Ag252 GPS unit, and a speed brake sensor (SBS) mounted directly to the transmission. These three sensors are combined to estimate a fifteen element vehicle state: x, y, z position and velocity, a bias on each accelerometer, vehicle yaw, pitch, and roll, and a bias on each rate gyro. This choice is motivated by classical satellite attitude estimation and modified for use in a ground vehicle [Lefferts82]. A typical square root information filter (SRIF) is implemented in C++ to fuse these measurements in a Bayesian optimal sense [Bar-Shalom01]. To facilitate computational speed, the filter states are split up into two separate SRIF estimators: an attitude estimator (yaw, pitch, roll, and rate gyro biases), and a position estimator (position, velocity, and accelerometer biases). Both estimators run at 400 Hz, faster than any terrain sensor on the vehicle. The attitude estimator numerically integrates rate gyro measurements with a fourth order Runge-Kutta scheme according to the kinematics of yaw, pitch, and roll [Triantafyllou04], [Wesstein]. Yaw, pitch, and roll measurements are provided by observations of GPS velocity, Earth's gravity vector, and Earth's angular velocity vector. These last two measurements can be made without a GPS fix, so attitude may be determined without GPS. Similarly, the position estimator numerically integrates accelerometer measurements with a fourth order Runge-Kutta scheme. Position and velocity measurements are provided by GPS. An additional velocity measurement is provided by the SBS sensor and the current attitude estimate. The SBS measurement is independent of GPS, and with it the filter can operate for several minutes without a GPS fix.

## Sensor Hardware

The Spider's sensor suite and IMU are mounted on a custom-built steel truss designed to withstand expected loadings during offroad-operation. The truss is built from 4130 steel tig for extra strength, and the configuration was designed to reduce the number of members in bending. Figure 2 shows the sensor suite and sensor platform. This platform is mounted to the front hood of the vehicle, allowing the sensors to see the ground far in front of the vehicle without being vulnerable in a collision. The platform supports two rigidly-mounted SICK LMS-291 LIDAR units, nominally capable of returning up to 80 meter ranges at half degree increments over a $90^{o}$ field of view, all at 75 Hz. Each of these LIDAR units is mounted using a turnbuckle and rod ends to allow its pitch to be adjusted without inducing yaw or roll in the LIDAR. The LN-200 IMU is rigidly mounted between these LIDAR units, enclosed with a 3003 aluminum skin. The inside of this enclosed area is cooled with a standard computer fan that maintains positive pressure to keep out dust.

To mitigate the effects of limited field of view and vehicle orientation, a two-axis gimbal platform was designed to house the third LIDAR unit and Team Cornell's stereo vision pair. Active cancellation of vehicle motion is performed by the gimbal at 100 Hz using harmonic drives in order to aim the LIDAR and stereo vision pair at target directions dictated by the AI. Gimbal yaw and pitch commands are calculated to point the center of the LIDAR in the desired direction by solving the inverse kinematics problem between the ground plane and the target location, taking into account the vehicle's current attitude. The gimbal's behavior is set to transition from scanning to long distance lookahead as vehicle speed increases.

**Figure 2 - The Spider's sensor platform and suite: two fixed LIDAR units, one gimbaled LIDAR unit, the gimbaled stereo vision pair and a Litton LN-200**

In addition to the three LIDAR units, team Cornell also uses a pair of cameras to create a depth map of the surrounding world using stereo vision. The Cornell stereo vision camera pair is a pair of Basler A311FC cameras equipped with 14.5$^{o}$ field of view long-distance lenses carrying a maximum range of approximately 40 meters. The stereo pair has a baseline of approximately 18.5 inches and operates at approximately 8 Hz.

The Cornell stereo algorithm is based on the Sum of Absolute Differences (SAD) algorithm and consists of five steps [Scharstein01]. As images are taken from the cameras, they are first transformed and rectified to eliminate distortion using the Intel OpenCV library [OpenCV]. Next, Sobel edge detection is performed to identify useful sections of the images [Scharstein01]. Third, stereo correspondence is computed on ¼ scaled copies of the images using modified SAD metrics to find matching regions in both images. Fourth, the stereo correspondence on the full-size images is computed using the correspondence from the scaled images to seed the search [Birchfield98]. Finally, stereo correspondence and geometry are used with OpenCV's transformations to reconstruct the sparse view of the world that is passed to the sensor fusion algorithm.

The Cornell vehicle also makes use of digital elevation models (DEMs) provided by Digital Globe to initialize its maps. The DEMs have errors less than 10m, and can be used to detect roads and large negative obstacles like cliffs. At the beginning of the race, this map data is used as low confidence sensor data to plan an initial path from the start line to the finish.

## Sensor Fusion

The Cornell sensor fusion system creates a complete and persistent map of the world around the vehicle. This map is kept in reference to an East-North-Up (ENU) local tangent plane, centered at some arbitrary latitude and longitude near the middle of the RDDF path. The persistent height map stored by the vehicle is divided into two different Cartesian maps, the sensor world model (SWM) and the artificial intelligence world model (AIWM). The SWM is a small map centered near the current vehicle location. It contains all the information captured within the current sensor horizon, valid over a very small window of time. The heavy computations for sensor fusion take place in the SWM over a relatively small set of sensor data, where the fusion for all the vehicle's sensors can be completed in real time. Once completed, the sensor data is reduced to a minimal set of information necessary to maintain recursive estimates of the height and variance within each cell. This simplified view of the world is passed to the AIWM where it is maintained as a global map for path planning.

Each new measurement in the SWM has a nominal ENU location with uncertainty approximated by a 3 x 3 covariance matrix and corresponding Gaussian density function. The covariance matrix is obtained by transforming and propagating errors through Jacobian calculations to incorporate errors due to the attitude / position estimator and the sensors themselves. Once the nominal location and covariance matrix of each measurement is known, the measurement can be added to the SWM. Because the location of the measurement is uncertain, the measurement is fused into the SWM cells according to the probability that it belongs in each cell. The measurement is only applied to cells in a small region around the nominal measurement location, though, to keep the computational complexity of adding a new measurement at O(1).

The measurements within each SWM cell are all fused via a recursive estimator, similar to one mentioned in Ref. [Bar-Shalom01]. Each height measurement in a cell has a height variance (the lower right 1 x 1 block of the covariance matrix) and a weight equal to the probability that the measurement belongs in the particular cell. At this level of fusion, all measurements are assumed to be independent, so the total probability density function for the height in a particular SWM cell is effectively a sum of Gaussians or Gaussian mixture model, with each Gaussian corresponding to exactly one height measurement [Bishop95]. The mean and variance of this distribution may be shown to be:

$$m = \frac{\sum_i p_i h_i}{\sum_i p_i}$$

$$s^2 = \frac{\sum_i p_i \left(s_i^2 + h_i^2 - m^2\right)}{\sum_i p_i}$$

where $p_i$ is the probability that height measurement $h_i$ belongs in this cell, and $s_i$ is the standard deviation of measurement $h_i$. For recursive estimation of the height and variance of each SWM cell, only the sums of the above equations need to be retained. These terms are the only data sent to the AIWM, yet they allow it to maintain a concise summary of the world around the vehicle.

### Computer Platform and Networking

The rough desert terrain can easily damage sensitive electronics required to operate a vehicle autonomously for extended periods of time. To prevent damage, the computers are stored in a pair of opposite-facing rack mounts situated across the back seat of the Spider, as shown in

Figure 3. The rack mount is vibration isolated from the floor of the vehicle by a six spring/damper suspension system. The suspension system is designed to constrain all six degrees of freedom of the computer rack and to keep displacement and force transmission low at the frequencies experienced by normal driving [Inman01].


**Figure 3 - The Spider's vibration isolated computer system.**

AMD Opteron servers provided by AMD and Seneca Data are used for all computationally intensive tasks. The sensor world model and stereo vision platforms each operate on separate quad-processor Opteron servers, and the artificial intelligence is run on a dual-processor Opteron server. The attitude / position estimator runs on a more standard single-processor Pentium IV computer. These computers were selected based on the expected computational needs of each system: both the sensor world model and stereo vision can be efficiently parallelized, while the path planner and the attitude / position estimator gain less from parallelization. For stability, all computers in the Spider run Microsoft's Windows Server 2003 Standard Edition. Inter-server communication between computers is done via user datagram protocol (UDP) over Ethernet connections. The highest-bandwidth links, from stereo to the sensor world model, and from the sensor world model to artificial intelligence, are done via point-to-point connections. The attitude / position estimator also has its own a dedicated network to distribute vehicle state estimates as fast as possible. This network architecture minimizes collisions and keeps latencies below 0.7 ms 98% of the time. Figure 4 shows the Spider's network architecture and data flow pathways.

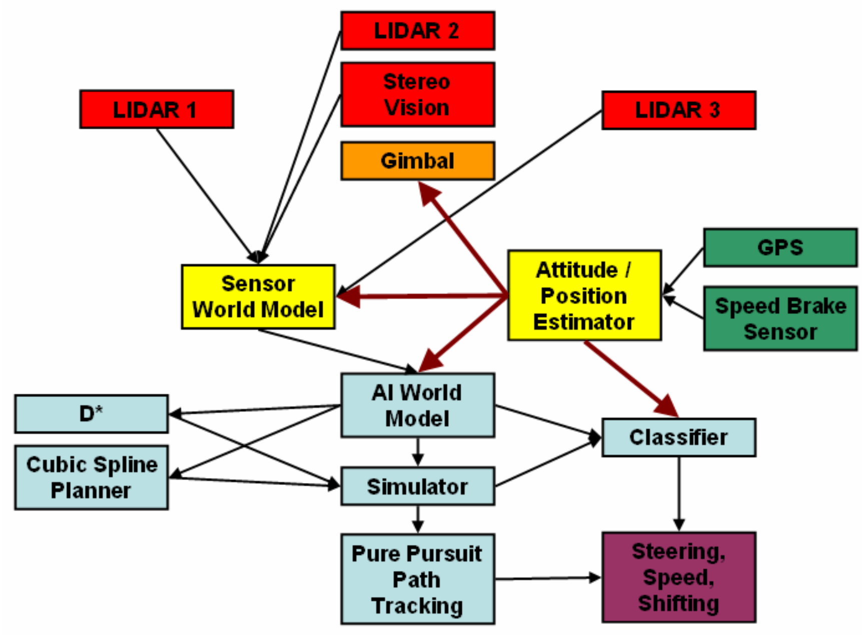


**Figure 4 - Pathways of information flow within the Spider. Dark red arrows indicate the dedicated attitude / position network, and arrows indicate the direction of information flow.**

## Artificial Intelligence

The path planning system in Team Cornell's Spider is designed around the concept of tracking a discrete Cartesian x-y grid with absolute elevations referenced from the WGS-84 ellipsoid stored in each grid cell. The gradient and variance of these heights are combined with the course route description data file (RDDF) waypoints and a priori map data to produce a cost of traversal for each cell. Multiple path planners then process the cell data and generate vehicle trajectories. The AI world model (AIWM) stores the cell data used by these path planners—it is the globally persistent half of the sensor fusion architecture described above, but unlike the sensor world model (SWM), the AIWM never processes raw sensor data. Instead, it receives the already-fused sensor measurements from the SWM. In essence, the AIWM is a complete map of the entire Grand Challenge course, while the SWM is just a sliding window that updates a small part of that map. Because the AIWM only maintains the minimal set of data required to form a height and variance estimate of each cell, it is feasible to store height data and costs for the entire course and access it quickly. This allows for globally optimal path planning, and it also provides a straightforward way to fuse sensor data with a priori data. Figure 5 shows a sample screenshot of the AIWM's post-processed cost.

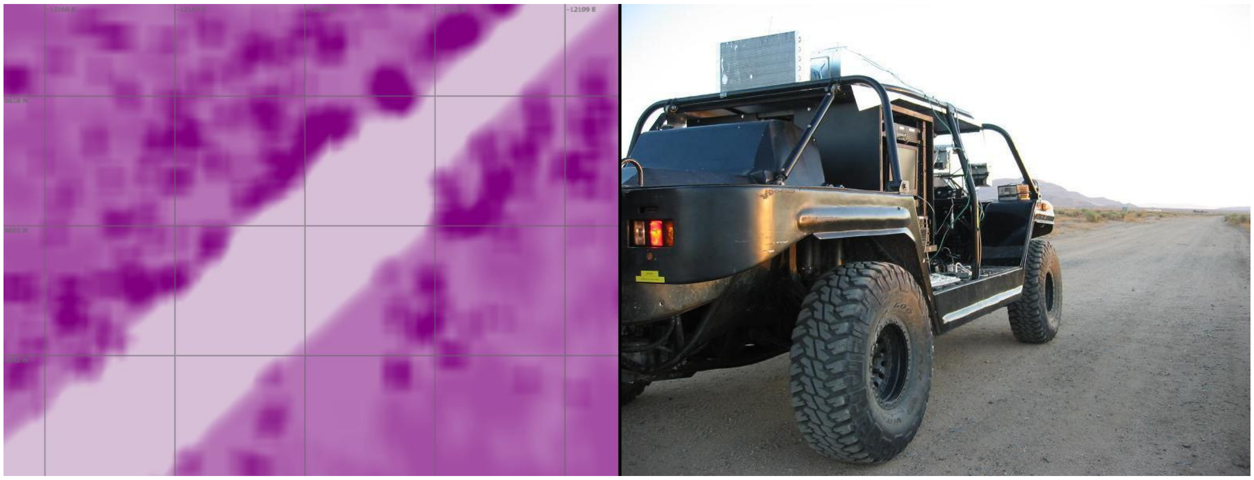

**Figure 5 - A desert road, as seen by the Spider's sensor fusion scheme and AI world model.**

## Path Planning

Team Cornell's Spider uses two complementary path planners: one that finds a globally optimal path but must be rectified with vehicle dynamics, and one that performs a greedy local search but is consistent with vehicle capabilities. The choice of path planner depends on several factors, including current speed, terrain type, and width of the corridor defined by the RDDF.

At slow speeds, in cluttered areas, and when the lateral boundary offset is large enough to provide an open area for path planning rather than a small corridor, the vehicle relies on D* path planning [Stentz94]. The discrete grid of cells with costs that makes up the AIWM is a natural environment for D*'s graph-traversal approach. This planner allows the Cornell vehicle to navigate complicated fields of obstacles in situations where the lateral boundary offset is substantially larger than the vehicle's width. When coupled with the persistent world model, D* also plans global paths out of difficult hazards such as dead ends. That is, it decides when to put the vehicle in reverse and try another path. D* has no sense of vehicle dynamics, though, so its paths require post-processing before they can be used. To solve this problem, the raw D* paths are decimated by removing unnecessary vertices in the path and instead taking straight lines if they are significantly more expensive. This removes the artifacts of grid resolution, and causes the planner to drift around obstacles more gently. Decimation also merits a second level of post-processing to prevent the vehicle from cutting corners. It is sometimes necessary for the vehicle to turn wide coming into a corner, as in the case when the vehicle approaches an underpass from the side. To make this possible, circles are fit to the decimated path such that each corner becomes three partial circles. The central circle describes the path to be taken around the outside of the corner, and the first and last circles are used to generate a continuous path. The algorithm maximizes the radius of the smallest of the three circles to generate smooth vehicle motion even in the presence of obstacles.

At higher speeds and in tight corridors, D*'s lack of vehicle dynamics becomes dangerous, and its globally optimal nature is unnecessary. In these situations, smooth paths are much more important than the ability to find a way out of dead-ends. To handle these circumstances, a simple greedy search spline-based path planner is used, similar to that presented in Ref. [Komoriya89]. This planner generates a number of cubic splines extending from the vehicle's current position and heading to a new heading along the RDDF path some distance ahead of the vehicle. This distance is set to the vehicle's sensor horizon, so the vehicle plans from its current location as far out as it can see. Using on-line simulation, the vehicle is then evaluated traversing each of these splines, and the lowest cost spline is chosen for the vehicle's path. Figure 6 compares the output of the two planners.

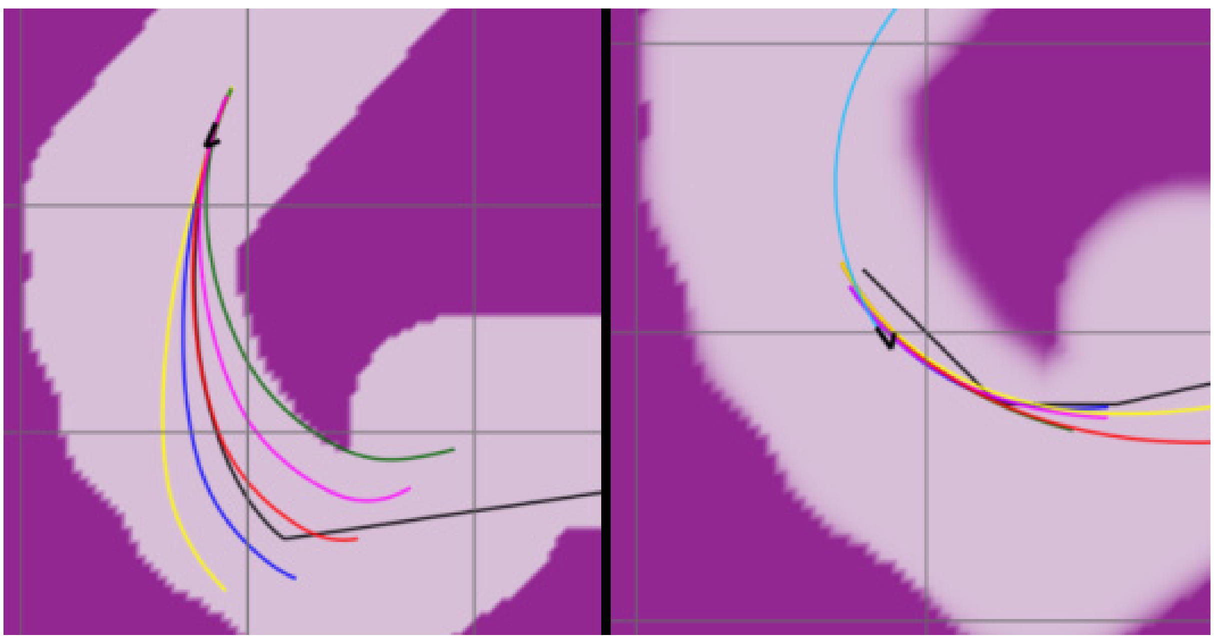

**Figure 6 - Sample planned cubic spline paths (left) and D* paths (right).**

### Simulation

Once one or more basic paths have been determined, AI must choose one of these paths and set the vehicle to follow it. At the lowest level of this software / hardware interface, vehicle waypoint tracking is accomplished using the Pure Pursuit algorithm described in Ref. [Coulter92]. This algorithm picks a steering wheel angle so the vehicle intersects the desired path at some lookahead distance in front of the vehicle. Vehicle speed is controlled independently, and must be determined by other means. There are therefore three types of decisions left to be made: the lookahead distance, vehicle speed, and the path to follow. The Cornell AI segment chooses between these parameters by simulating the vehicle in software. The vehicle simulation is a simple fourth-order Runge-Kutta numerical integration scheme [Weisstein]. The dynamics model chosen for the vehicle is a simple set of planar dynamics described in Ref. [Wit00], augmented by first order models of engine and steering wheel lag:

$$
\begin{aligned}
\dot{x} &= v\cos(\theta) \\
\dot{y} &= v\sin(\theta) \\
\dot{\theta} &= \frac{v\tan(\phi)}{a} \\
\dot{\phi} &= bp(u_p - \phi) \\
\dot{v} &= bv(u_v - v)
\end{aligned}
$$

where x and y are the vehicle's location in the planar projection of the world, v is the vehicle's speed, $\theta$ is the vehicle's heading, $\phi$ is the vehicle's front wheel angle with respect to the heading, a is the length between the vehicle's two axles, bp and bv are time lag constants for the steering and engine control loops, respectively, and $u_p$ and $u_v$ are commanded front wheel angle and speed. Integration time is adjusted to encapsulate the time and distance it takes the vehicle to come to a complete stop at its current speed, so AI will have plenty of time to react if all the simulated paths produce undesirable or hazardous results. The simulations are performed over a wide variety of commanded speeds, lookahead distances, and paths. Costs for the actual path the vehicle traverses are accumulated in simulation from the AI world model, and these costs are used to determine which path and vehicle parameters produce the best results. Dynamic checks for rollover, side slip, and front slip similar to those in Ref. [Spenko04] are also used to penalize or eliminate paths that are more hazardous. The best path is then passed to the actual vehicle controller.

### Situation Classification and Situational Awareness

The highest level of decision-making is handled by a hand-crafted decision tree called the Classifier. The Classifier turns components of artificial intelligence on and off and redirects their output according to the vehicle's current situation. During the autonomous drive, the Classifier dictates the transitions between D* and the cubic spline planner based on speed, environment, and each planner's success in finding a traversable path. It also checks for special vehicle situations such as getting stuck, operating in reverse, or executing a three-point turn. The Classifier also attempts to correct failures in the system by monitoring sensor and vehicle health.

## Reliability Testing

To prepare the Spider for the 2005 DARPA Grand Challenge, Team Cornell has spent two months honing the system at the Mojave Airport in Mojave, California. The airport features miles of dirt roads, bushes, and obstacles indicative of those found in the Grand Challenge environment. In addition, the vehicle has been tested with permission at Jawbone Canyon, an off-road vehicle recreation site just north of Mojave. The Jawbone site features difficult desert terrain, sweltering heat, high voltage power lines, valleys, steep hills, cliffs, and washboards that are some of the most intense terrain navigable by a commercial vehicle.

The Spider performs reliably in the difficult environment of the desert. All software systems on the vehicle are equipped with automatic recovery, so processes may be restarted if they encounter error. All computers are monitored by watchdog-like mechanisms, so any computer can be entirely restarted in the event of a more serious system failure. In addition, the artificial intelligence has the ability to detect and repair sensor failures by restarting any of the sensors on the Spider. Special filters are also added to each sensor at the lowest level to ensure that it is ignored if it consistently reports faulty information. The Classifier also monitors vehicle health, and has the capability of adjusting vehicle behavior based on engine and generator temperatures, as well as several other vehicle health metrics. With these systems, the Spider has been proven in many miles of autonomous drives without system failure.

## Conclusion

Team Cornell's Spider has many unique features that make it a reliable and sensible entry in the 2005 DARPA Grand Challenge. The vehicle employs a reliable attitude / position estimator, as well as a rigorous sensor fusion algorithm to model and handle sensor uncertainty. The artificial intelligence segment generates paths with consistent vehicle dynamics and global characteristics. It also incorporates a detailed vehicle dynamics simulation to predict the actual path the vehicle will take through the world. A classification system governs the entire architecture, incorporating fault detection and situational awareness far beyond the level of a simple path planner. The system has been tested extensively in the deserts of the Southwestern U. S., and it has been found to perform reliably even in the harshest conditions. Team Cornell's Spider has what it takes to be a serious competitor in the 2005 DARPA Grand Challenge.

## References


Bar-Shalom01: Bar-Shalom, Y., Rong Li, X., Kirubarajan, T. Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software. New York: John Wiley & Sons, Inc., 2001.

Birchfield98: Birchfield, S., and Tomasi, C. "Depth Discontinuities by Pixel-to-Pixel Stereo." Proceedings of the Sixth IEEE International Conference on Computer Vision, Mumbai, India (1998): 1073 – 1080.

Bishop95: Bishop, C. Neural Networks for Pattern Recognition. Oxford: Oxford University Press, 1995.

Coulter92: Coulter, R. C. "Implementation of the Pure Pursuit Path Tracking Algorithm." Carnegie Mellon University Technical Report CMU-RI-TR-92-01, January, 1992.

Franklin02: Franklin, G., Powell, D., and Emami-Naeini, A. Feedback Control of Dynamic Systems, Fourth Edition. Upper Saddle River: Prentice-Hall, 2002.

Inman01: Inman, D. Engineering Vibrations, Second Edition. Upper Saddle River: Prentice-Hall, 2001.

Komoriya89: Komoriya, K., and Tanie, K. "Trajectory Design and Control of a Wheel-Type Mobile Robot Using B-Spline Curves." IEEE / RSJ International Workshop on Intelligent Robots and Systems, Tsukuba, Japan, September, 1989.

Lefferts82: Lefferts, E., Markley, F., and Shuster, M. "Kalman Filtering for Spacecraft Attitude Estimation." Journal of Guidance, Control, and Dynamics. 5 (1982): 417-429.

OpenCV: "Intel Open Source Computer Vision Library." Intel Technology and Research Website. <http://www.intel.com/technology/computing/opencv/>

Scharstein01: Scharstein, D., Szeliski, R., and Zabih, R. "A Taxonomy and Evolution of Dense Two-Frame Stereo Correspondence Algorithms." Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision, Kauai, HI, Dec. 2001.

Spenko04: Spenko, M., Iagnemma, K., and Dubowsky, S. "High Speed Hazard Avoidance for Mobile Robots in Rough Terrain." Proceedings of the SPIE Conference on Unmanned Ground Vehicle Technology VI, Orlando, FL, Vol. 5422, April 2004.

Stentz94: Stentz, A. "Optimal and efficient path planning for partially-known environments." In Proceedings of the IEEE International Conference on Robotics and Automation, May 1994.

Triantafyllou04: Triantafyllou, M., and Hover, F. "Maneuvering and Control of Marine Vehicles." Fall, 2004. Maneuvering and Control of Surface and Underwater Vehicles MIT OpenCourseWare Website. < http://ocw.mit.edu/NR/rdonlyres/Ocean-Engineering/13-49Fall-2004/C8072E70-A4F4-466A-98DF-A3EC2865A835/0/lec1.pdf>

Weisstein: Weisstein, E. "Runge-Kutta Method." From Mathworld – A Wolfram Web Resource. http://mathworld.wolfram.com/Runge-KuttaMethod.html

Wit00: Wit, J. Vector Pursuit Path Tracking for Autonomous Ground Vehicles. University of Florida, 2000.