Technical Overview of CajunBot (2005)

Team CajunBot

ARUN LAKHOTIA[1], Team Leader
GUNA SEETHARAMAN[2], Faculty Advisor
ANTHONY MAIDA[1], Faculty Advisor
ADAM LEWIS[1], Graduate Student
SURESH GOLCONDA[1], Software Engineer
AMIT PUNTAMBEKAR[1], Graduate Student
PABLO MEJIA[3], Chief Architect

Contact author: Arun@Louisiana.Edu

[1]Center for Advanced Computer Studies (CACS), the University of Louisiana at Lafayette
[2] Air Force Institute of Technology, Wright Patterson Air Force Base
[3]ElPrez.Net

Submitted to

DARPA Grand Challenge 2005

DISCLAIMER: The information contained in this paper does not represent the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA) or the Department of Defense. DARPA does not guarantee the accuracy or reliability of the information in this paper.

Technical Overview of CajunBot (2005)

Arun Lakhotia[1], Guna Seetharaman[2], Anthony Maida[1], Adam Lewis[1], Suresh Golconda[1], Amit Puntambekar[1], and Pablo Mejia[3]
[1] University of Louisiana at Lafayette; Air Force Institute of Technology; [3] ElPrez.Net

I. Abstract

CajunBot, Team CajunBot's entry in the DARPA Grand Challenge 2005 is a radically transformed version of the entry in the 2004 challenge. While it is still built upon a MAX IV ATV, its mechanical frame, sensor system, hardware architecture, software architecture, and algorithms are completely overhauled. In its maximal configuration CajunBot uses five SICK LIDAR sensors, four computers connected over a 1GB switch, a RT3102 INS, and CNAV differential GPS. Its software uses blackboard architecture, even though data mostly flows sequentially in a pipeline form. The obstacle detection and path planning algorithms may have significant breakthroughs and are currently considered proprietary.

II. Introduction

Team CajunBot consists of faculty and students of the University of Louisiana at Lafayette and professionals from the Lafayette and Louisiana communities. The team did its maiden foray in the area of autonomous vehicles in the 2004 Grand Challenge, and is now returning to the 2005 Grand Challenge with some significant innovations in the core computational problems related to obstacle detection, path planning, and dealing with sensor inaccuracies.

While the base vehicle in the two editions is the same six wheeled, skid-steered MAX IV ATV, everything else in the 2005 edition has changed. Figure 1 presents pictures of the 2004 edition of CajunBot and the 2005 edition, currently in production. The student crafted cage using aluminum rods purchased from a local hardware store has been replaced by a student-designed but professionally manufactured aluminum structure. A hand rigged case that served as a rack is now replaced by a MIL-spec rack manufactured by Hardigg. A single, garden variety mother board is replaced by two Dell Power Edge 750 computers and two mini-ITX boards. The single (functional) SICK LMS 221 is augmented by four SICK LMS 291s. The radar and sonar sensors are removed. The POS/MV INS from Applanix has been replaced by RT3102 from Oxford Technologies. The single Honda EU2000 generator now shares a berth with another identical generator. Two linear actuators from Ultramotion still form the electromechanical interface for steering, as does a servo motor for throttle.

The most significant, though invisible, aspect that has changed is the CajunBot software system. While still maintaining blackboard architecture, developed using POSIX shared memory, the software system has been completely overhauled.

The rest of the document is organized as follows. Section III provides details of the vehicle, the automotive aspects. Section IV summarizes the electrical and electronic systems. Section V describes the software system. Section VI describes the methods used to test the vehicle and is followed by a section discussing our conclusions and future plans.

CajunBot 2004

CajunBot 2005

Figure 1: CajunBot pictures from 2004 GC and one getting ready for 2005 GC

III. AUTOMOTIVE

The base of CajunBot is a MAX IV all-terrain vehicle (ATV) manufactured by Recreative Industries. MAX ATVs are extremely maneuverable due to their skid-steering, full-time six-wheel drive and extremely low ground-pressure that allows them to traverse incredibly soft, muddy or swampy terrain. Load capacities range from 600 to 1000 pounds and all units can tow up to 1000 pounds dead weight.

Power is supplied by a 28hp Kohler twin-cylinder engine with electric start, reverse gear, headlights and taillights, and a Borg-Warner designed T-20 Skid Steer transmission. The original 5-gallon fuel tank has been modified to feed two Honda EU2000 Inverter Generators used to power the electronics. A replacement fuel tank with a total capacity of 27 gallons has been added to provide fuel for the vehicle itself.

Table 1: CajunBot Chassis Specification

Length (inches)	**96”**
Width (inches)	**84”**
Height (inches)	**57”**
Weight fully fueled (pounds)	**2000 lbs**
Method of locomotion (wheels, tracks, legs, etc.)	**Wheel**
Minimum ground clearance (inches)	**9”**
Minimum turning radius (feet)	**9ft**
Power source	**Internal Combustion**
Fuel type	**Unleaded Gasoline**
Fuel capacity (gallons)	**37 g.**

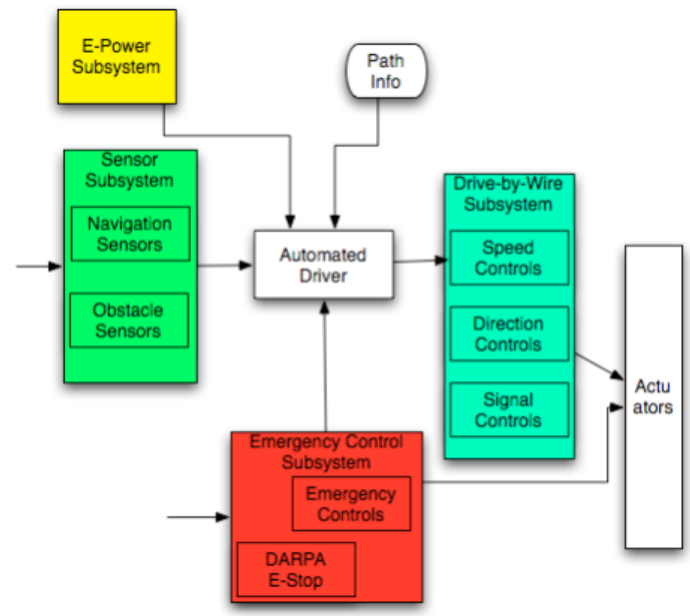

Figure 2: Electrical and Electronic System Topology

IV. Electrical and Electronics

The electrical and electronics components of CajunBot are organized into five major subsystems:

- Sensors Subsystem
- Drive-By-Wire Subsystem
- Emergency Control Subsystem
- E-Power Subsystem
- Automated Driver Subsystem

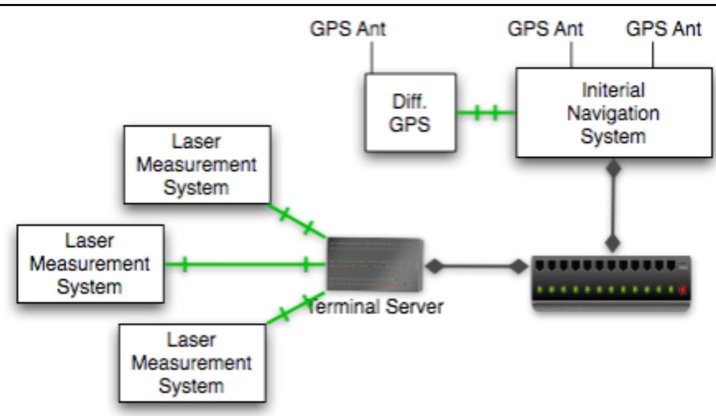

Figure 3: Sensor Subsystem

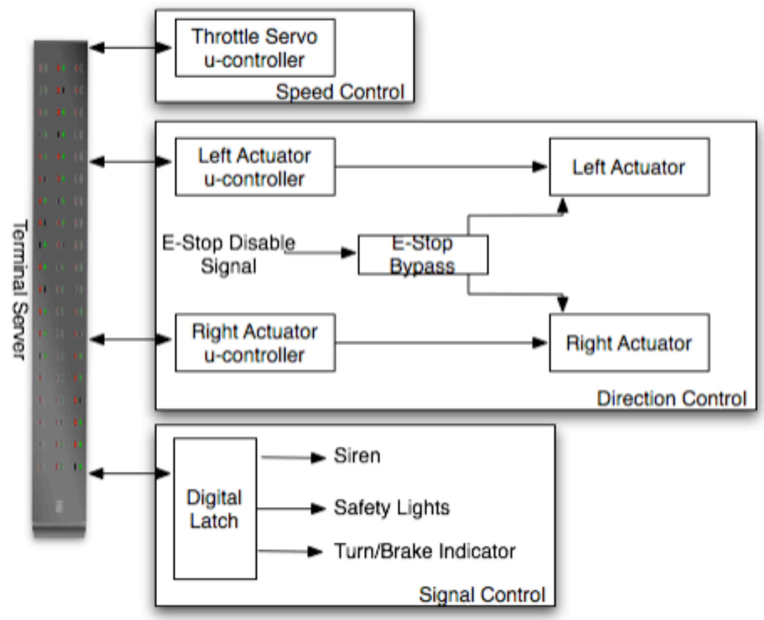

Figure 4: Drive-By-Wire Subsystem

A. Sensor Subsystem

The Sensors Subsystem, Figure 3, of the CajunBot consists of navigation sensors and obstacle sensors. An Oxford Technical Solutions RT3000 inertial navigation system (INS) is the primary navigation sensor. The accuracy of the INS is enhanced by Starfire differential GPS correction signals provided by a C&C Technologies C-Nav receiver.

Four SICK LMS 291 LIDARs and one SICK LMS 221 constitute CajunBot's obstacle sensors. Three of these sensors are mounted on the front of the vehicle looking at a distance of 25m, 29m, and 30m along the direction of the vehicle's motion. The remaining two sensors are mounted on the left and right of the vehicle, scanning from 3m from the 'shoulder' of the vehicle to 30m in front of the 'nose'.

The five SICK LMS configuration is the maximal configuration. CajunBot can also perform well, albeit at a reduced speed, with only one of three SICKs mounted on the front.

B. Drive-By-Wire Subsystem

The Drive-By-Wire Subsystem, Figure 4, provides the electro-mechanical controls to affect change in the state of the vehicle by manipulating current. *Speed Control* is provided by a servo and microcontroller connected to the throttle. *Directional Control* is provided by a pair of linear actuators connected to the left and right steering levers (one actuator per steering lever). The left (right) levers are used to stop the three wheels on the left (right) sides, to make the

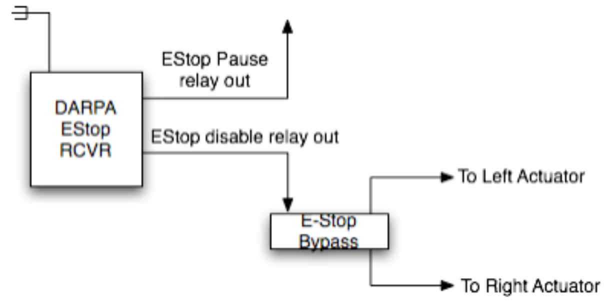

Figure 5: Emergency Control Subsystem

vehicle turn to that side. These actuators also act as the brakes of the vehicles by pulling both the levers at the same time. *Signal Control* is provided by a microcontroller and set of relays connected to the vehicle's siren, strobe lights, and turning/braking indicators.

C. Emergency Control Subsystem

A requirement of the Challenge is that participants must provide a mechanism to apply an emergency stop of the vehicle. The emergency stop mechanism is expected to be implemented completely at the hardware level, bypassing and overriding the software system.
There are three methods to bring CajunBot to an emergency stop. 1) Using the E-stop provided by DARPA, 2) using the E-stop button of RC Control of CajunBot, and 3) pushing stop buttons physically located on each side of the vehicle.

The Emergency Control Subsystem, Figure 5, receives the DARPA E-stop signal using the *E-stop* radio receiver provided by DARPA. This receiver can receive two commands: pause and disable. Similar disable and pause commands are also generated by the RC Control (used when DARPA's E-Stop is not available). The Emergency Control System also processes the disable signal generated by the kill button just like the disable signal from other means.

The disable signal is directly routed to the drive-by-wire system but the pause signal is routed to the Automated Driver. When the disable signal is received the drive-by-wire subsystem pulls the left and right levers to the brake position, cuts the throttle, and turn on an E-stop flashing light (not required by DARPA).

D. E-Power Subsystem

E-Power Subsystem, Figure 6, provides three power sources—120 VAC, 12 VDC, and 5VDC—for the electrical and electronic components of CajunBot. Two inverter generators in parallel supplies 2300W each at 120VAC powers a 725VA UPS unit. The UPS units clean the power coming out of the generator and also protect against power glitches such as when unplugging from shore power and plugging into generator power. These feed a chain of 12V, 10A regulated power supplies that provide power for DC electronics. A 5VDC regulator feeds off the 12VDC to power low voltage DC components.

E. Automated Driver Subsystem

The Automated Driver Subsystem, Figure 7, is the brains of CajunBot. It consists of the two Dell Power Edge 750 computers and two mini-ITX computers along with the associated software.

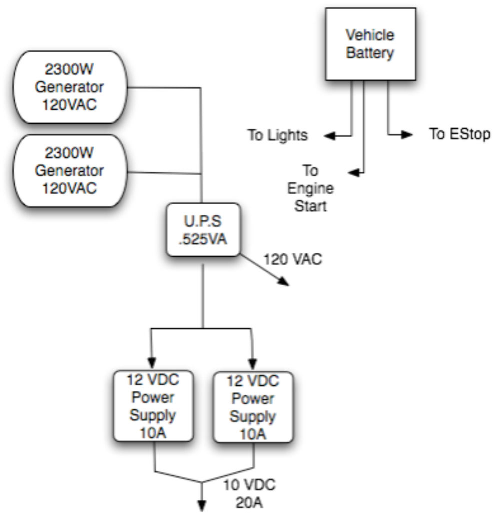

Figure 6: CajunBot E-Power Subsystem

The two Dell computers provide the primary computational power, whereas the mini-ITX computers provide CPU cycles for non-computation heavy tasks. Both the Dell computers are needed when using three or more SICK sensors. When only one or two SICK sensors are used, just one Dell computer suffices. Thus, in the maximal configuration the computers are used as follows:

- Obstacle Detection Server (Dell)
- Path Planning Server (Dell)
- Time Server (mini-ITX)
- Data Logging Server (mini-ITX)

The servers communicate with each other using a custom application UDP protocol through a 1GB switched Ethernet network. Data taken from and sent to the other subsystems is transmitted to networked connected terminal servers. The physical devices in the other subsystems are connected to the terminal server.

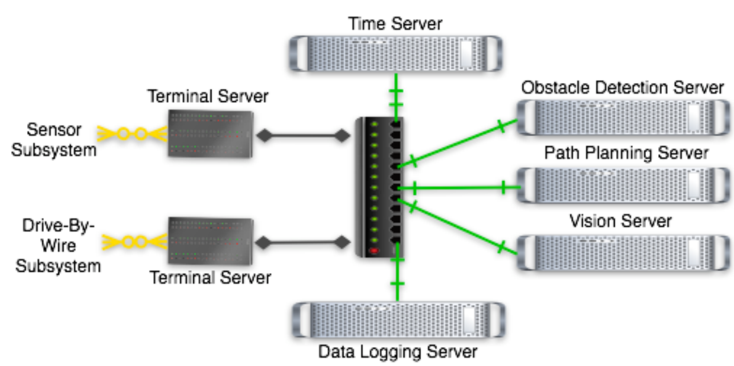

Figure 7: CajunBot Automated Driver

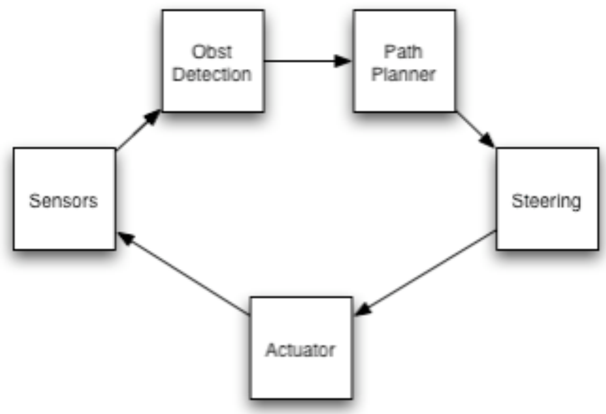

Figure 8: The CajunBot Control Loop. The feedback from Actuator to Sensors is through the environment.

V. SOFTWARE

The CajunBot Autonomous Guidance System is written in C++ on the Fedora Core 2 distribution of the Linux operating system. The system is decomposed into several processes organized using the blackboard architecture. Communication between processes on the same machine is achieved using queues in shared memory. The queues from one machine are distributed to another machine using a specialized publish-subscribe protocol. The distribution of shared memory is transparent to the processes. Thus, the system can be scaled from using a single computer to multiple computers without making any changes to the software system.

The software processes of CajunBot can be grouped along the control loop shown in Figure 8, and described below.

1) Sensor Drivers

A sensor driver reads data from a physical sensor device, formats the data into the internal form, and publishes the data in the inter-process communication queues. Each sensor driver is a separate process.

2) Terrain Modeling and Obstacle Detection

The LIDAR sensors provide range data—the distance at which a laser beam hits something. The coordinates of every beam reflected from the target are converted into global X, Y, and Z coordinates using the position of the vehicle reported by the GPS, the orientation of the vehicle reported by the INS and the offsets of the sensor with respect to the INS. Triangular grids are formed out of the points that are close in time (as the Z value of the GPS is good only in limited time interval). A cross product of the two sides of the triangle yields a normal vector that describes the orientation of that particular triangular plane with respect to the global axis. This orientation is associated to the centroid of the triangle. The coordinates of the centroid and their respective orientations are mapped onto a Digital Terrain Map made up of square cells. The DTM is then analyzed for discontinuities to identify obstacles. A small portion of the DTM, called the Local DTM, which is geographically close to the vehicle is extracted out and passed to the path planner.

The team has developed a highly efficient method to perform the above operations. The algorithm does not produce any significant false positives or false negatives, even with a single LIDAR sensor. As multiple sensors are added the overall computational cost increases by a fraction, and the time to produce the results decreases rapidly. Thus, the algorithm scales well with increased number of sensors.

A special property of the algorithm, which can also deal with moving obstacles such as vehicles and gates, is that it does not require that the sensors be stabilized to reduce the shocks and vibrations they experience. This reduces the cost of developing the system since we do not need to use a gimble to stabilize the sensors. In addition, we also do not need any specialized equipment to move a sensor so as to scan more than a single plane. Instead we utilize the motion of sensors induced by normal motion of the vehicle to increase the span that can be seen by the robots. This also cuts the cost of our solution by turning its weakness (absence of suspension) into a strength.

The algorithm is being evaluated by the University for its IP value. Hence, its details are not being disclosed in this public document.

3) Path Planning

The Path Planning module uses the location information from INS and the terrain and obstacle model from the local DTM to compute a path around obstacles or other non-navigable areas.

Planning algorithms may be broadly classified into two classes: discrete and differential. The first class of algorithms can be used in robots with unlimited mobility traveling at slow speed on a flat terrain. The second class of algorithms take into account the kinematics and dynamics of a vehicle and are used for autonomous vehicles traveling at high speed on uneven surfaces. The discrete algorithms are computationally less expensive than differential algorithms.

The Path Planning module of CajunBot uses a combination of discrete and differential planning techniques. It first uses techniques used by discrete algorithms to compute costs of traveling along a grid world. However, once the costs are calculated it uses a differential algorithm to extract paths that take into account the kinematics and dynamics of the vehicle.

The RT3102 can produce sub-meter accurate data in ideal conditions. However, when the conditions deteriorate, the accuracy can drop to 20 meters. The Path Planning algorithm deals with the inaccuracies in the INS data by (a) expanding the size of the lateral boundary and (b) searching for ‘road’ in this expanded region.

The algorithm is being evaluated by the University for its IP value. Hence, its details are not being disclosed in this public document.

4) Steering

Once the decision about what path needs to be taken around obstacles has been made, the steering process generates commands to steer the vehicle. The steering process is independent of the steering mechanism of the vehicle. The steering commands consist of two floating point values: turn and speed. A positive turn value indicates left turn and a negative value indicates right turn. Similarly, a positive value of speed implies throttling up and a negative value indicates slowing down, or active braking.

5) Actuator Drivers

The actuator drivers interface the CajunBot software to the physical devices, such as actuators, lights, sirens, etc. There are two such drivers, one for interfacing with the two actuators and servo and the other for managing the light and sound signals. The drivers read command generated by the steering module and convert them into commands for the specific devices.

VI. TESTING

The testing methodology used in development of the CajunBot operates on four levels:

1) Simulated unit level testing

A vehicle simulator is included in the CajunBot software suite. The simulator provides a test environment that emulates the physical environment in which the vehicle operates. Daily builds of the software are tested against a collection of test cases gathered from the real world. Developers perform unit level testing of changes to the software using the combination of the vehicle simulator and visualization tools included in the software suite.

2) Vehicle integration testing

Mechanical and electrical design changes are tested in combination with the software during vehicle integration testing. This testing occurs in a controlled physical environment setup on test courses at our development shop, the University's athletic field, and the University's former Horse Farm. The vehicle is operated in both manual control and autonomous control in these locations.

3) Vehicle system testing

Vehicle system testing occurs at the University's former Horse Farm. In these tests, the vehicle operates on courses that simulate expected conditions at the NQE and the race. Endurance testing of the vehicle occurs during this testing phase.

4) Vehicle field trials

Vehicle field trials test the capability of the vehicle to run for long periods in real world conditions. In these trials, autonomous runs occur on public roads and along the local levee system in the Lafayette area and on public roads, ranch trails, and desert terrain in the desert areas of West Texas. The autonomous runs vary in length from 1 mile to 10 miles depending upon terrain and safety conditions. Over 100 miles of autonomous running has occurred during our vehicle field trials.

VII. FUTURE WORK

Besides a skid-steered ATV based autonomous vehicle, in future challenges Team CajunBot plans to develop autonomous vehicles based on front-wheel steered vehicles. One such vehicle, Ragin'Bot, based on a 2004 Jeep Wrangler Rubicon, is currently under development. Our intent is to use the same software, except actuator drivers, to drive the two types of robots. The use of such drastically different vehicles as the basis for the two robots, while increasing the complexity of the task, has also helped in identification and parameterization of the vehicular differences. The choices also increase the possibility of at least one vehicle doing well, for they make different trade-offs between speed and agility. CajunBot, with a five feet turning radius, is agile. But its top speed is around 25 miles/hour. In contrast, the turning radius of Ragin'Bot is 20 ft, and it can easily reach speeds upwards of 100 miles/hour.

We also have prototype implementations of camera-vision based terrain and obstacle modeling algorithm. The algorithms are not completely tuned to be used for the 2005 Grand Challenge, and will be fielded in the next challenge (if there is one).

VIII. SPONSORS

We are thankful to the following companies and individuals for their support of the project: C&C Technologies, Lafayette, LA; Ray Majors and family, Melville, LA; MedExpress Ambulance Service, Alexandria, LA; Oxford Technology Solutions and Brendel Associates; SOLA Communications, Lafayette, LA; Lafayette Motors, Lafayette, LA; BEGNAUD Manufacturing, Lafayette, LA; Louisiana Department of Transportation; Recreative Industries, Buffalo, NY.; FireFly Digital, Lafayette; and Pixus Printing, Lafayette.