ION: The Intelligent Off-Road Navigator
The Desert Buckeyes' entry in the DARPA Grand Challenge 2005

Ümit Özgüner, Team Lead

Abstract

This paper describes the Ohio State University entry for the 2005 DARPA Grand Challenge for Autonomous Vehicles. The Team is the Desert Buckeyes, the vehicle ION (Intelligent Off-road Navigator). ION was developed in partnership with the University of Karlsruhe, which developed the vision system. ION is a 6 wheeled vehicle with full drive-by-wire capability. A set of sensors (LIDAR's, radars, cameras and ultrasonic transducers) and a GPS and IMU unit provide extensive sensing capability. A sophisticated sensor fusion system was developed and is used with a complex intelligent analysis, decision and control configuration. The development was in parallel with an education process, where a number of OSU students got involved in learning the issues and solving some of the problems.

Acknowledgement

As the Desert Buckeyes were the team that developed the sensing and intelligence for the 2004 TerraMax, a number of aspects of ION are descendants of technology and approaches we used in 2004 and a number of individuals participated in this endeavor through its development. Our thanks go to all. The full list is indeed too long, so only the core group who contributed to the present report is listed: Q. Chen, J. Martin, K. Redmill, C. Toth and U. Ozguner.

1. Vehicle Description

1.1. Vehicle selected

The vehicle the Desert Buckeyes are using is a 2005 Polaris Ranger 6x6 shown in Figure 1. It is 120 inches long and 60 inches wide and its height is 78 inches.

Figure 1. ION on a practice run at TRC.

The Polaris Ranger was selected due to its agility, off-road driving capability, small turn radius and ease of modification.

1.2. Actuation and physical modifications to vehicle

Drive by wire capability was added to the vehicle so that computer control was possible for throttle, brake, steering control, and transmission gear. The engine fuel tank can be augmented with a second fuel tank during the race to extend the range of the vehicle. A frame was added to the front bumper to provide some safety to fore mounted sensors. Two mounting bars were added to the front of the cab, again for sensor mounting.

The low-level actuator control electronics was mounted under the front hood of the vehicle. The remaining electronics were mounted in a shock-mounted metal enclosure with forced-air ventilation for protection from the elements, dust, and terrain induced vibrations affixed to the cargo bed of the vehicle. A gasoline-powered generator and separate fuel tank were also mounted in the cargo area.

During the race, the vehicle will be equipped with run-flat off-road tires.

2. Autonomous Operations

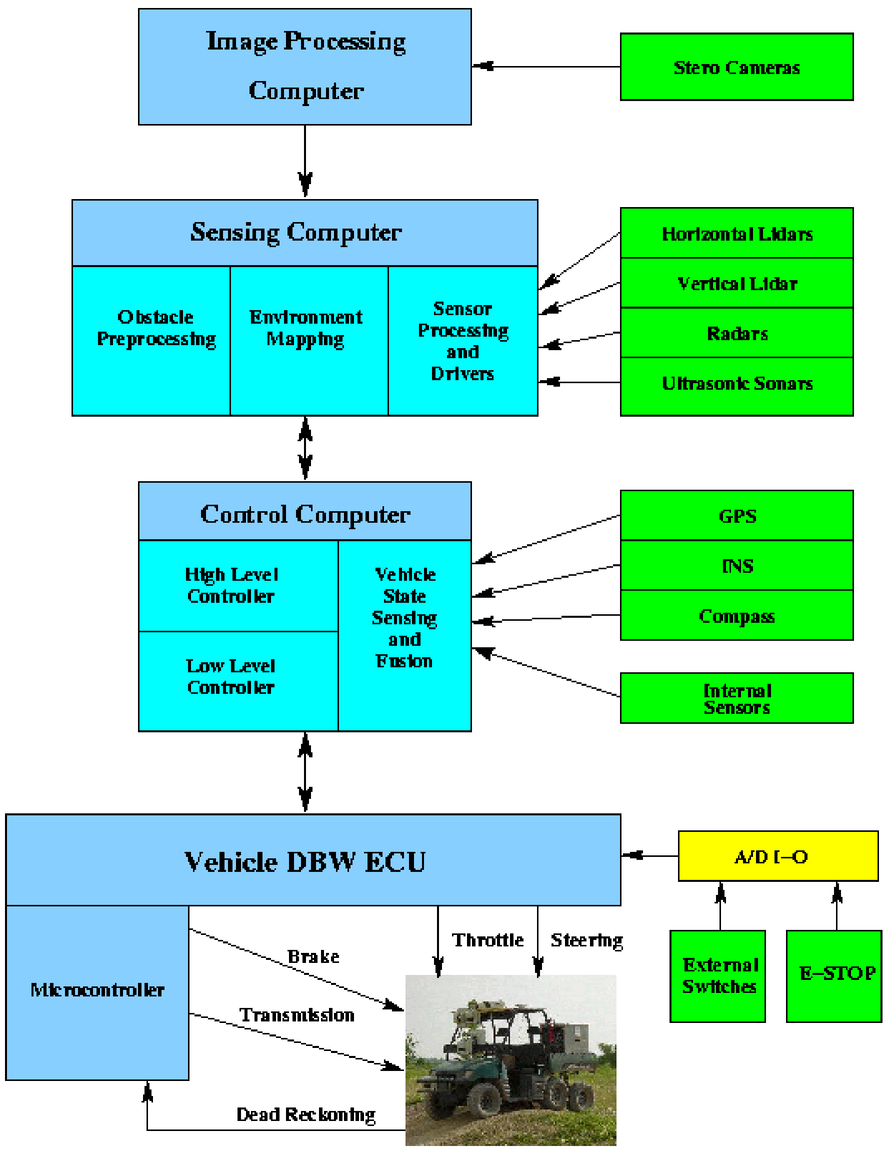

Figure 2, ION System structure

2.1. Processing

2.1.1. The computing system

The computing system is comprised of three separate computers (vision, sensor fusion and control computers). The control computer runs the QNX real-time operating system, the sensing and sensor fusion computer runs Linux, and the vision processing computer runs Microsoft Windows XP. Algorithms are programmed in C and C++. Lessons learned with the TerraMax experience were taken to heart and a simpler, cleaner configuration and interprocessor data communication mechanism was created.

2.1.2. Functional block diagram and processing architecture

The over-all block diagram is shown in Figure 2 above, and the software structure is provided in Figure 3 below.

The main challenge was in making sure the risks of transferring large and small packages of data in inter-processor communication was mitigated. This had been observed in last year’s Grand Challenge where it was the main source of error in TerraMax.

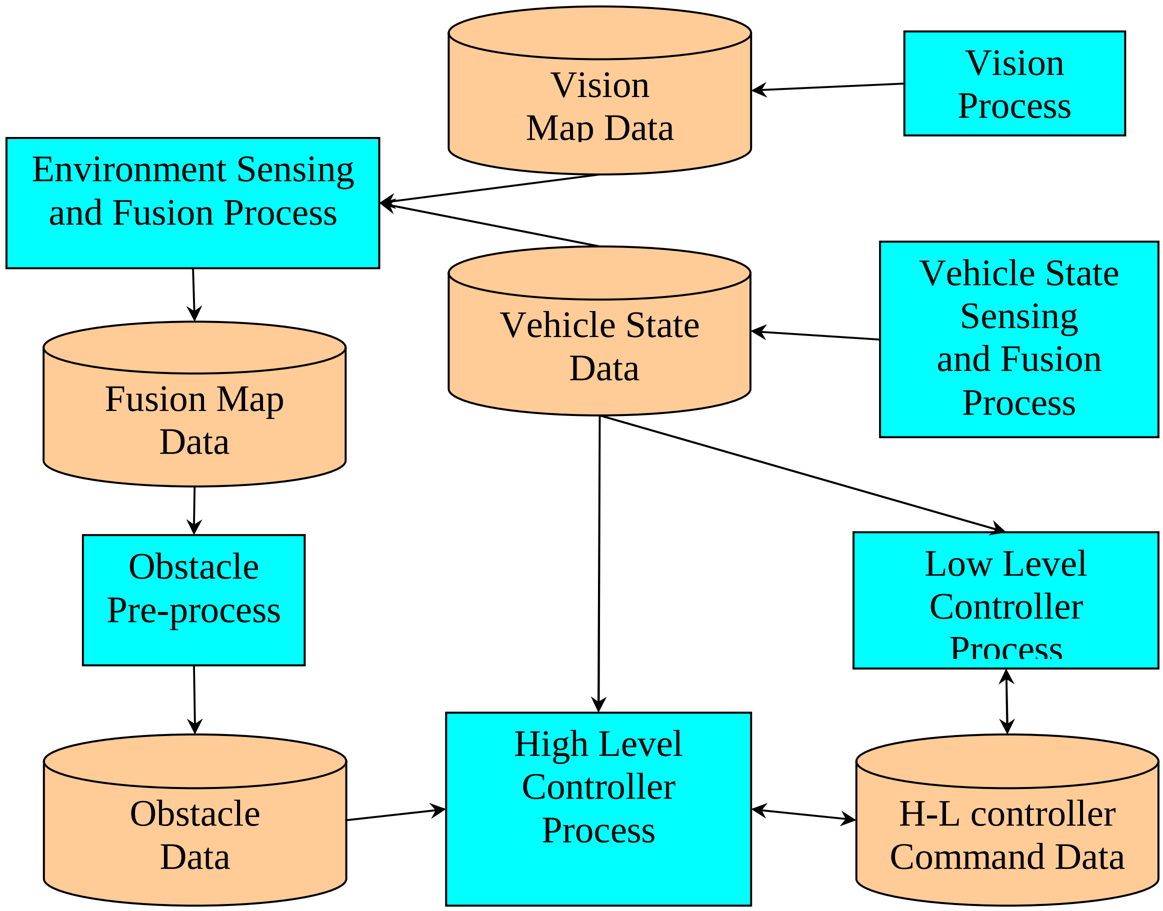

Figure 3. ION Software architecture

2.1.3. Development process

The development was based on the already existing architecture and software from the 2004 TerraMax. A "task hierarchy" had been established and was also followed here. Portions of the software were tested on different simulation and emulation environments. Two specific simulation environments were developed for testing obstacle avoidance. One was a simple, flexible 2-D package for initial testing. The second was based on the Player/Gazebo environment and with the 3-D developments made, could actually include terrain configurations from real data. Two hardware environments (smaller robots with a smaller suite of sensors) were also used both indoors and outdoors for testing during the development cycle.

A two Quarter “Capstone Design Course” sequence was initiated and helped investigate new ideas, apart from inspiring students.

Figure 4. The test robot with GPS and Lidar in the GC Design Course at OSU.

The Vision system was developed entirely separately by the University of Karlsruhe team in Germany, and then integrated with the Sensor Fusion set-up at Ohio State University. The cross-Atlantic cooperative development was similar in nature to the one we initiated with an Italian team in 2004, while developing TerraMax.

2.2. Localization

2.2.1. GPS System

The GPS used is a Novatel Propak LB-L1L2 with Omnistar HP differential correction. The IMU is a Crossbow VG700AA-201.

Software has been developed to provide inertial navigation during brief glitches in GPS and longer, full outages. This uses both the IMU and additional local measurements.

2.2.2. Map data

Map data (as provided from pre-race information) can be considered an integral part of the system in an indirect way.

Before the race, when the waypoint file is supplied, we have developed software to:

- Show the waypoints on satellite photographs of the region
- Add virtual waypoints
- Shift location of waypoints (presumably within the given boundaries)

- Find paths that were pre-run
- Find "optimal" routes

During the race, ION uses the data file of waypoints produced (DARPA given and pre-race generated) as the basic pre-planned path but also generates a local Grid Map. The Grid Map is a map that is the result of sensor fusion and supplements the pre-planned path.

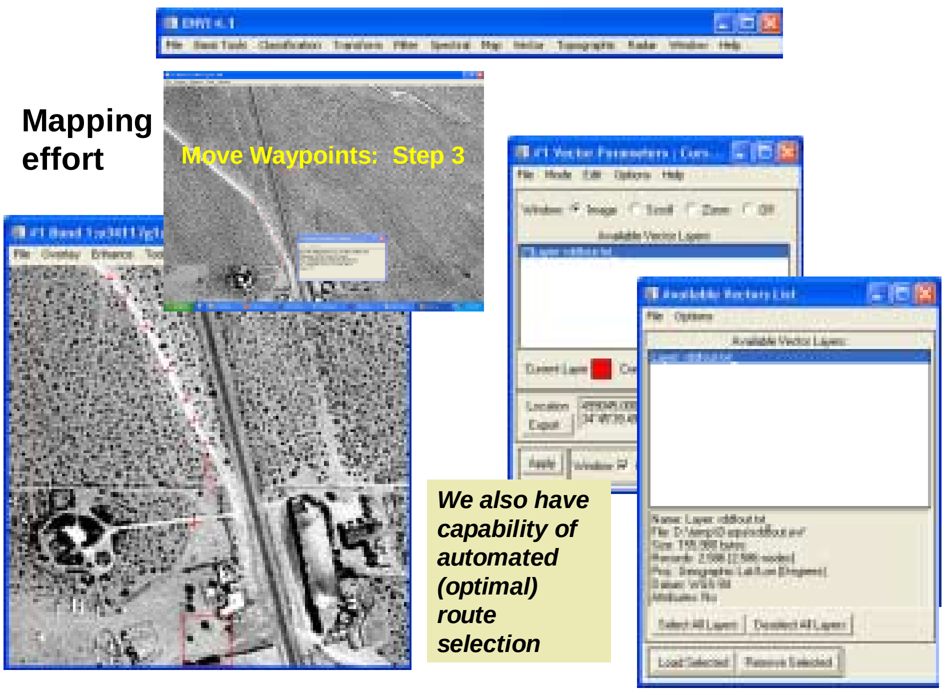

Figure 5. GUI for pre-race path planning.

2.3. Sensing

2.3.1. Location and use of sensors

ION has LIDAR's, radars, cameras and ultrasonic sensors to sense its immediate surroundings. All 4 LIDARS (SICK LMS221-30206 scanning rangefinders) are facing the front. Three of them, mounted at different heights, are scanning in horizontal planes, to provide obstacle detection and height estimation. The forth lidar is scanning in a vertical plane and is used primarily to aid in ground profile estimation. One radar (the Eaton-Vorad 300 EVT) is pointed straight ahead and is mainly for long distance obstacle detection at high speed. The second radar has a slewing dish antenna and is an in-house development. It points forward and slews approximately +-- 60 degrees. The stereo camera system is mounted on the top and again pointed ahead. 8 ultrasonic rangefinders are mounted around the vehicle and are for short distance sensing at low speeds, and in particular for sensing in confined areas when the vehicle is operating at low speeds and potentially without accurate position information. In particular, two rear facing ultrasonics provide the only backwards facing obstacle detection capabilities. Two additional ultrasonic rangefinders are mounted high on the front of the vehicle and angled downward, in an attempt to detect sharp dropoffs on either side of the vehicle.

A sketch of the effective ranges and fields of view is given in Figure 6.

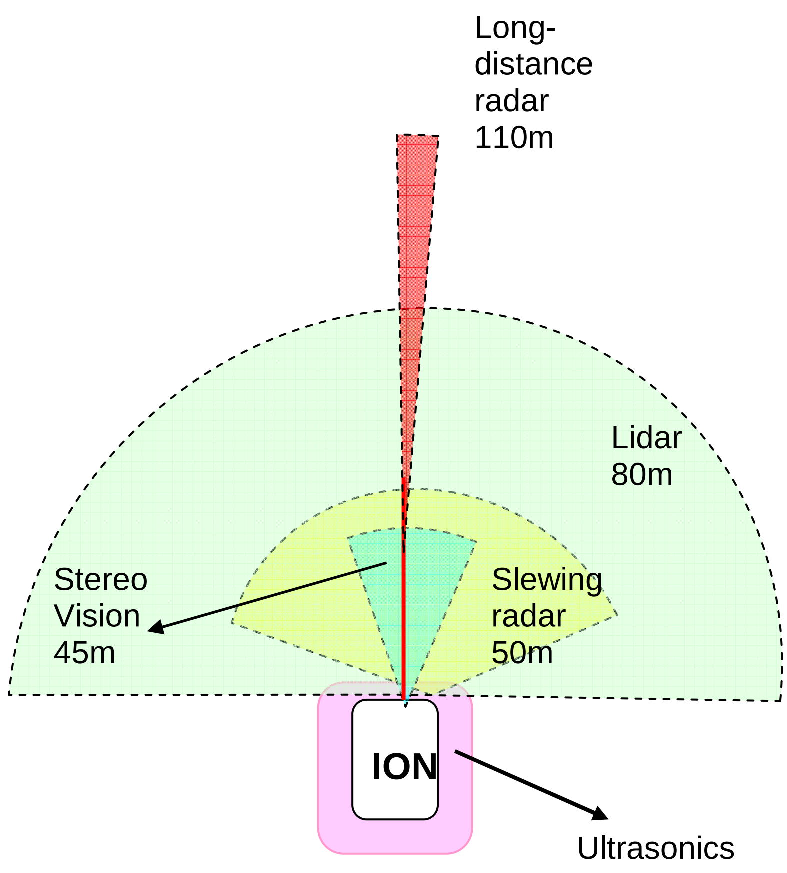

Figure 6. Sensor coverage.

Compensation for vibration and other vertical motion is done in software, using the IMU data, specifically generating a “ground plane” that can be referred to, while doing sensor fusion. The cameras and vision system adjust for light levels. Cameras are housed in a special casing that has a wiper.

2.3.2. Sensing architecture

The sensing architecture/sensor fusion is established by developing a grid map of the vehicle surroundings. All external sensors feed this map with obstacles sensed and related confidence level. The map is maintained internally in vehicle centered world coordinates. This means that the map doesn't rotate with the vehicle, but it does translate. A portion of this map is supplied to the high level control module at 10 Hz. The portion supplied is in vehicle coordinates (map rotates and translates with the vehicle). A display in the vehicle cab can show the map during testing.

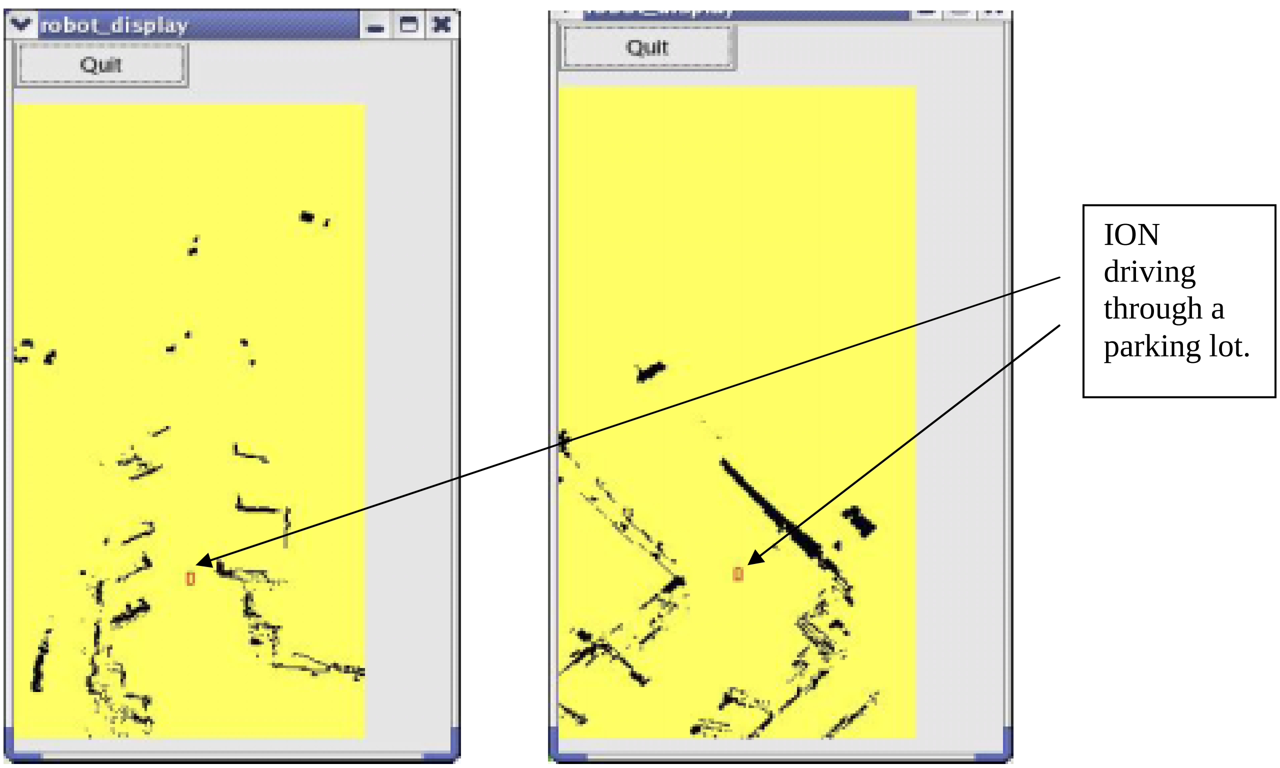

Figure 6. The Grid Map as demonstrated with one LIDAR.

2.3.3. Internal state sensing

Localization, vehicle motion status, and internal state sensing is accomplished using the Novatel GPS with Omnistar HP differential corrections, the Crossbow IMU, wheel speed sensors that were added to the vehicle, engine speed measurements, and brake and throttle position information. These sensors are monitored for changes in their operating state, validated using both dynamic and rule based tests, and finally fused using a Kalman filter based approach to provide continuous position and orientation information even the presence of individual sensor dropouts, reduced accuracies, or complete failures.

2.3.4. Sensing to actuation

Different control approaches and algorithms have been used in the different control loops. These are basically the same as those we have used for TerraMax 2004, although with different gain settings. They can be found in our papers:

- U. Ozguner, K. Redmill, A. Broggi, "Team TerraMax and the DARPA Grand Challenge: A General Overview," Proc. *IEEE Intelligent Vehicle Symposium,* June 2004, Parma Italy.
- H. Yu, Q. Chen, U. Ozguner, "Control System Architecture for TerraMax," *Proc. 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, July 2004, Lisbon Portugal.

As appropriate PI controllers and sliding mode controllers have been designed for lower level speed and steering control. Speed set point selection is situation dependent. Acceleration or deceleration is done with pre defined profiles (using braking if needed). Steering control is done with a small set of way-points (two to four) being passed on to the lower level, with the lower level steering to a look-ahead point.

2.4. Vehicle Control

2.4.1. Common contingencies

Vehicle high-level control can best be explained in terms of the state diagram provided in Figure 7 below. Various contingencies are within the different states in the diagram.

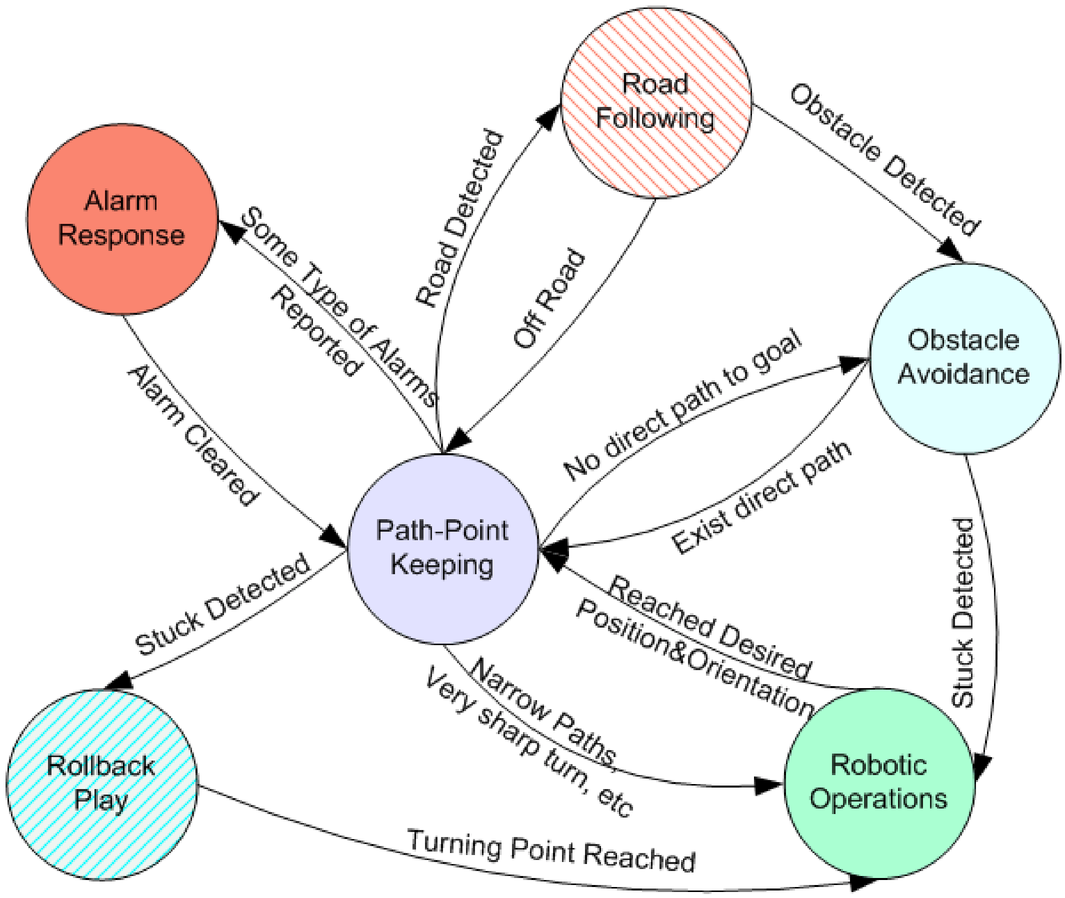

Figure 7. State diagram for ION operation.

For missed waypoint: We consider a waypoint is reached if the distance to the waypoint is smaller than the boundary offset, or if the distance to the next waypoint is smaller than the distance of next section and the vehicle is inside the corridor of the next section. Thus, we can skip one waypoint, if something bad happens, and we won't skip to the next section at sharp turns.

For vehicle-stuck: We record and monitor the movement of the vehicle. We consider the vehicle is stuck if it doesn't move 3 meters from previous position in 2 minutes. When the vehicle is doing backup, the time is reset. If paused, we don't consider it stuck until up to 4 minutes after we restart from pause. When stuck is detected, depending on the distance of obstacles in the front and in the back of the vehicle, we decide if we should go ahead or backwards with a relatively high throttle command for a short period of time.

For vehicle-outside-lateral-boundary-offset: We check if we are outside of the lateral boundary offset. If so, we will pick a low speed and go toward a reasonable point which lies in the middle of the corridor and keep doing obstacle avoidance.

For obstacle-detected-in-path: Once an obstacle is detected in the path, we might reduce the speed a little bit according to the distance to the obstacle. If there is enough space for the vehicle to pass through, we generate a new path that avoids the obstacle. If the obstacle is blocking the path, we will slow down and stop before the obstacle. If the obstacle exists for a long time, like longer than 2min, the stuck situation will be detected and will go backwards to have another try.

2.4.2. Maneuvers

Braking, starting on a hill:

The speed set point is generated regardless of the slope of the ground. The speed controller has the "integration" part that keeps increasing the throttle if the vehicle is slower than the speed set point so that we can climb a hill. In order to stop short in some situations, the vehicle applies the maximum brake pressure.

Making a sharp turn without leaving boundaries:

Out vehicle has a small turning radius. When we approach a waypoint, the trajectory we plan shifts from the center of the corridor that makes the sharp turning easier. When we encounter very sharp turns, we might stop and go back a little to adjust the vehicle orientation without leaving the boundaries.

2.4.3. Integration of navigation and sensing information

Again, this is easier to understand in terms of the state diagram of Figure 7. In general, the vehicle will be following the pre-planned path while monitoring for obstacles. It is however possible that a road may be detected *within the boundaries* and then the vehicle could follow the roadway. (It is possible, for example, for the vehicle to take the middle of the road, even if the waypoints are at the edge.) If an obstacle is detected, the sensing information will dominate and a path will be selected in looking at the local grid-map.

2.4.4. Non-autonomous operation

ION is controlled through its drive-by-wire hardware by the use of a joystick.

2.5. System Tests

2.5.1. Testing

Before the Desert Buckeyes Team came to the testing stage various simulation environments were used. We developed four simulation/emulation environments. (See Section 2.1.3. above.)

Initial testing was done at the OSU campus at the Center for Automotive Research and Intelligent Transportation (CAR-IT) facilities. Extensive full systems were performed at two tracks within the TRC Inc. site and off road just outside TRC.

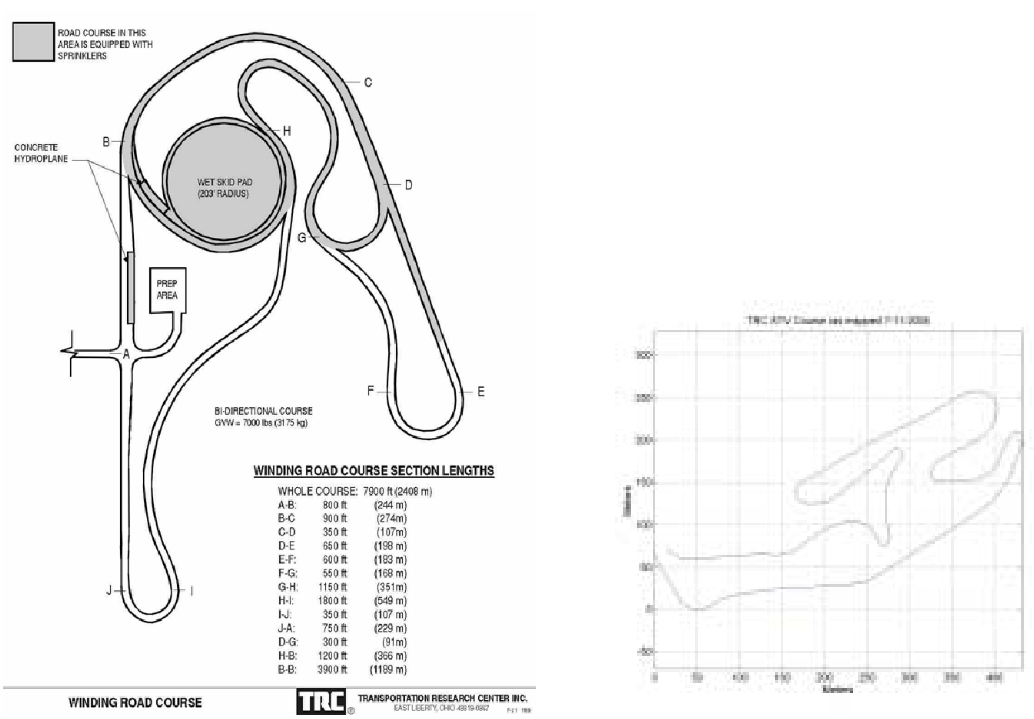

Figure 8. The paved “winding course” and off-road ATV course at TRC.

2.5.2. Results and challenges

The key challenges are

- Maintaining knowledge of “ground plane” for all sensors to refer to (in spite of off-road conditions)
- Keeping all subsystems mechanically working
- Covering all possible “special situations” and scenarios
- Graceful degradation and removal of systems that are not working