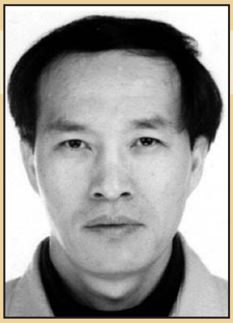

Editor: Fei-Yue Wang
University of Arizona and
Chinese Academy of Science
feiyue@sie.arizona.edu

Ohio State University at the 2004 DARPA Grand Challenge: Developing a Completely Autonomous Vehicle

Qi Chen, Ümit Özgüner, and Keith Redmill, *Ohio State University*

In March 2004, DARPA held its first Grand Challenge (www.darpa.mil/grandchallenge), a 142-mile mostly off-road race for 19 unmanned vehicles through an unrehearsed route across the Mojave Desert. DARPA's goal is to accelerate research and development activities in autonomous ground vehicles, increase collaboration among different organizations and interest groups, and draw widespread attention to the technological issues and benefits of developing completely autonomous off-road ground vehicles.

At Ohio State University, we have extensive experience in developing autonomous ground vehicles. Our team developed three vision- and radar-directed cars to participate in the US Automated Highway demonstration in 1997 and a car that followed a series of predefined routes using GPS in 1999. However, an off-road race such as GC 04 provided a series of new and different challenges. To meet these challenges, we formed Team TerraMax in collaboration with Oshkosh Truck Corporation (www.oshkoshtruckcorporation.com), which provided an MTVR (Medium Tactical Vehicle Replacement), an 8 × 28 ft., 32,000 lb. truck manufactured for off-road military applications. Our goal was to create a completely autonomous vehicle that could negotiate most of the GC 04 route and perhaps even finish the race.

Editor's Perspective

The DARPA Grand Challenge is a field test designed to accelerate R&D in autonomous ground vehicles that will help save lives on the future battlefield. In this issue, Team TerraMax relates its experiences in the first Grand Challenge, which generated great interest worldwide.

Feel free to contact me if you have any comments about this department. I also seek contributions on the status of ITS projects worldwide and on innovative ideas and trends in future transportation systems. Contact me at feiyue@sie.arizona.edu.

Fei-Yue Wang

Technical challenges

To achieve our goal, we needed to conquer three main technical challenges. The first was to install drive-by-wire technology in the vehicle and to modify the vehicle to fulfill DARPA safety requirements. In addition, we added equipment to provide electrical power for computers, actuators, and sensors and a second fuel tank to allow the vehicle to run for over 10 hours and over 300 miles.

The second challenge was to provide sensing and information fusion algorithms. The autonomous vehicle needed to be able to sense both vehicle states such as position, speed, and direction and environment information such as the existence and relative position of obstacles. The autonomous-driving system would collect the information from different kinds of sensors installed on the vehicle. So, to provide accuracy, robustness, and usability, we needed to filter and fuse the information before the vehicle could use it.

The third challenge was control. On the basis of the fused information, the autonomous-driving system would need to make the correct decision, apply the navigation algorithm, and properly control the vehicle through the drive-by-wire capability. Visual road detection, map-based path finding, and real-time communication between computers were also important issues.

Hardware integration

The hardware installed on TerraMax (see Figure 1) consists of the drive-by-wire electronic actuators, local network, computers, vehicle ego state sensors, and several different types of environment sensors. Figure 2 diagrams the TerraMax hardware.

The full TerraMax navigation and control system consisted of six computer systems. All the computers ran a version of Linux except the central computer, which used QNX, a real-time operating system, to provide guaranteed timing for the critical control loops and high-speed Kalman filters.

The connections in Figure 2 are virtual links. The connections between computers are through a local network, and the connections between computers and sensors are through hardware interfaces such as RS-232 and RS-422,

1541-1672/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

and through directly connected analog and digital signals.

The environment sensors, including six cameras, four ladars, two radars, and 12 sonars, are mounted on TerraMax to detect obstacles around the vehicle (see Figure 3). A fusion algorithm in the environment-sensing-and-fusion computer puts each obstacle into local map cells with a certain confidence degree. This computer sends the local map, together with other parameters, to the central computer, on the basis of which the high-level controller makes decisions. Two GPS devices, a compass, and an inertial navigation system sense the vehicle's ego state, such as position; speed in 3D; and yaw, pitch, and roll and their rates. The central computer collects the raw data and passes it to an extended Kalman filter to estimate the vehicle states. The filtered states are then available to other software modules through virtual-shared-memory buffers.

Figure 1. The TerraMax vehicle.

Software integration

The autonomous-driving system consists of eight processes that work on different computers and two operating systems, QNX and Linux. The data communication among these processes uses the *Real-time Control System Library* (www.isd.mel.nist.gov/projects/rcslib), developed by the US National Institute of Standards and Technology. Through RCSLib, multiple read or write processes on the same processor, across a backplane, or over a network can access a fixed-size buffer of general data. Using the Neutral Message Language that RCSLib provides, we designed shared buffers for the eight processes. Figure 4 shows the relations between these processes.

The processes work asynchronously by sharing the information through the buffers. The mono-vision and stereo-vision processes generate two local maps indicating any roads or paths identified and free space or obstacles detected ahead of the vehicle, and then update the two related buffers. The environment-sensing-and-fusion process obtains the updated map data and fuses the information with data collected from ladars, radars, and sonars to generate the local map around the vehicle indicating each obstacle's position, shape, and size. Then, this process updates the fusion-map-data buffer so that the high-level controller will get the latest local-map information. Combining this information with information from the lat-

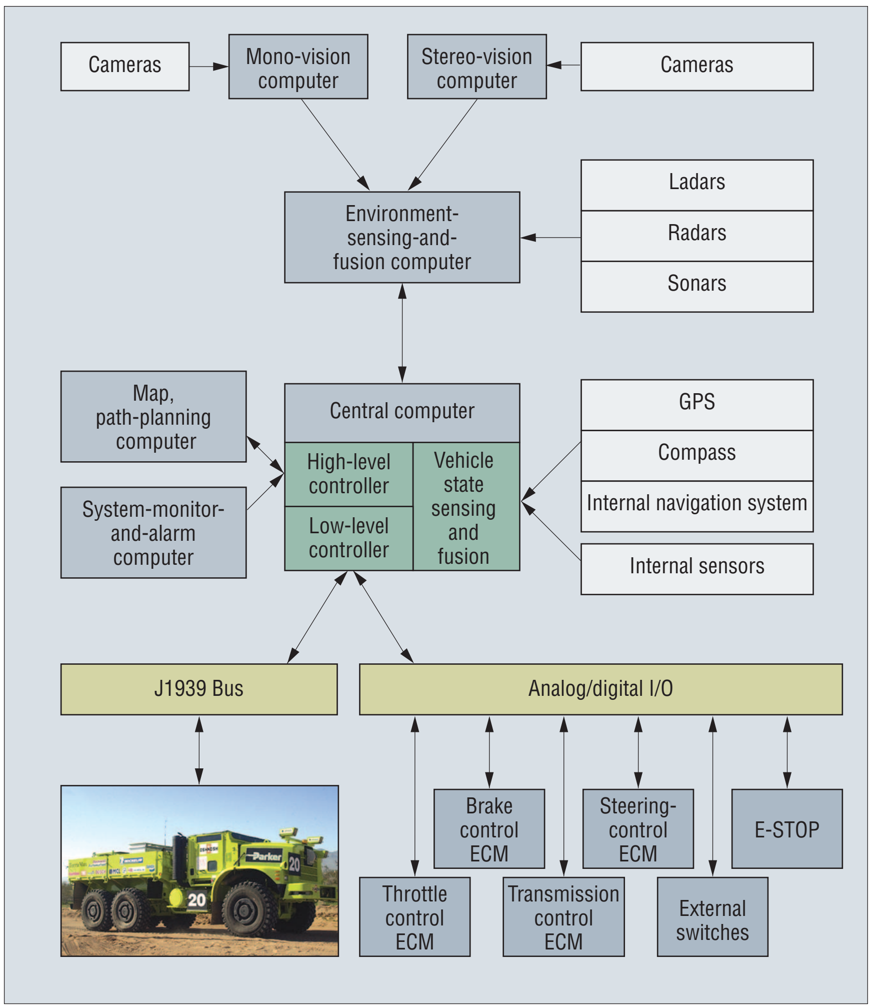

Figure 2. TerraMax's hardware.

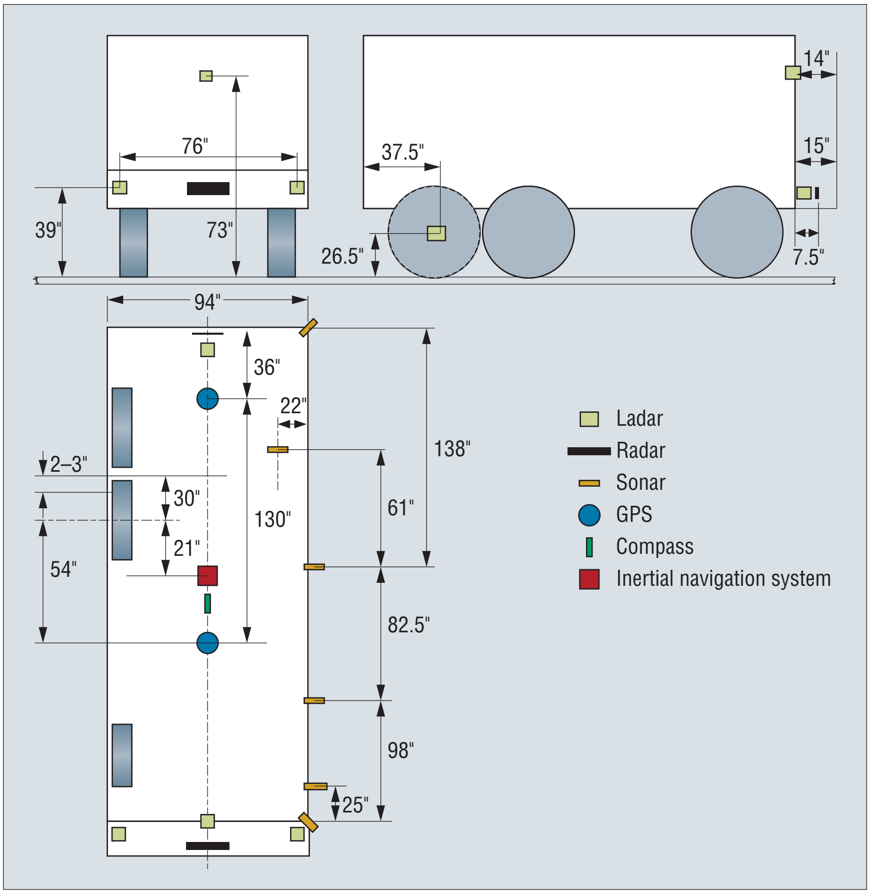

Figure 3. The mounting of the sensors on TerraMax.

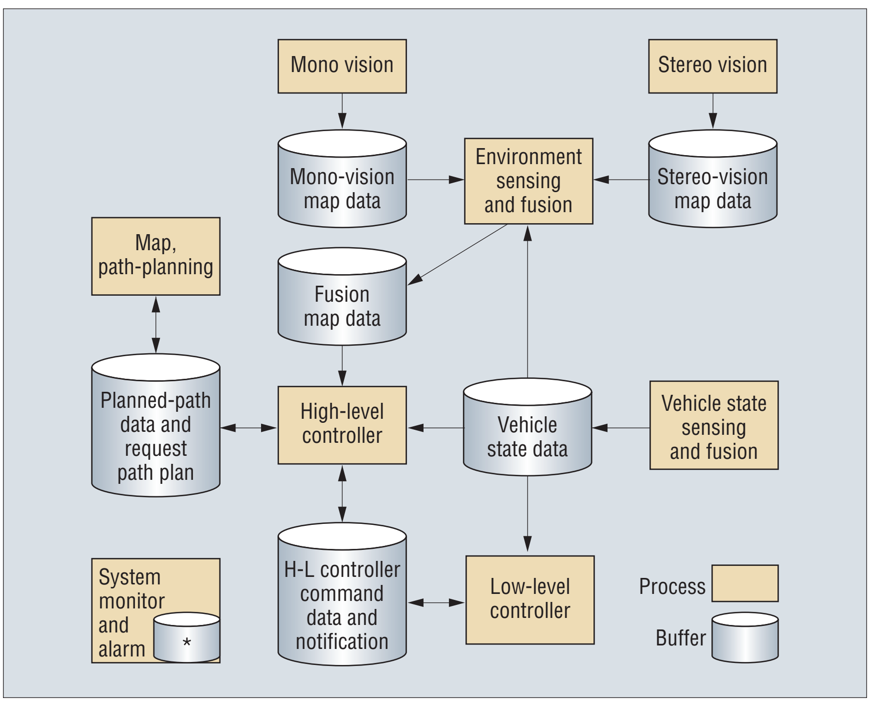

Figure 4. Data communication between autonomous-driving-system processes. Every process in the system has read and write access to the buffer with the asterisk.

est updated vehicle-ego-state-data buffer, the high-level controller selects the drive mode according to a set of rules. On the basis of this mode, the high-level controller generates commands for the low-level controller to follow—for example, the speed set point and a set of GPS route points, or the specification of a sequence of preplanned motions that we called *robotic unit operations*. The high-level controller sends the commands to the low-level controller through another shared buffer, which also stores the notification from the low-level controller to the high-level controller.

The map-based path-planning process generates a sequence of road points when two given checkpoints are far away. Acting as a server, this process answers the request from the high-level controller by sending back the planned-path data through the shared buffer.

The system-monitor-and-alarm process manages a shared buffer that collects polling messages from all the other processes. This lets the process detect and broadcast alarms concerning malfunctions and failures without undue delay.

Defining behavior using a finite-state machine

Because TerraMax is autonomous, central decision-making is a key issue.

Our system uses a finite-state machine to decide which drive mode the high-level controller should select. Figure 5 shows the eight modes (states) and the finite-state machine's corresponding transitions.

The basic and most important mode is the path-point-keeping mode, which we expected our system to employ for the most of the race. In this mode, the vehicle can fully implement the techniques we demonstrated in 1997 and 1999. If the vision systems detect a road, this mode can transition to the road-following mode, which provides better navigation. Detection of an obstacle will trigger the obstacle avoidance mode, which not only dodges obstacles but also maintains the vehicle's approach to the goal point. To deal with narrow paths and sharp turns, the robotic-operation mode implements robotic reasoning to generate a sequence of unit operations to handle these complicated situations. The rollback mode is used when the vehicle gets stuck because it is unable to identify a course to follow from its present circumstances. The vehicle-following and vehicle-passing modes activate when there's a slow vehicle ahead (but this situation never occurred in GC 04). The alarm response mode takes effect when the system-monitor-and-alarm process detects known failures or emergencies, such as sensor failures or out-of-bound control commands.

The QID event

Before the race, all challenge vehicles were inspected, tested, and required to demonstrate their abilities in the Qualification, Inspection, and Demonstration event. The one-and-a-half-mile QID course consisted of most of the terrain types and obstacles that an autonomous vehicle would likely meet along the Grand Challenge course, including dirty hills, ditches, sand pits, tunnels and underpasses, cattle guards, boulders, towers, and even moving obstacles.

Because we detected a network malfunction before the QID event, we decided not to connect the two vision computers to the whole system. So, the road-following mode was unavailable.

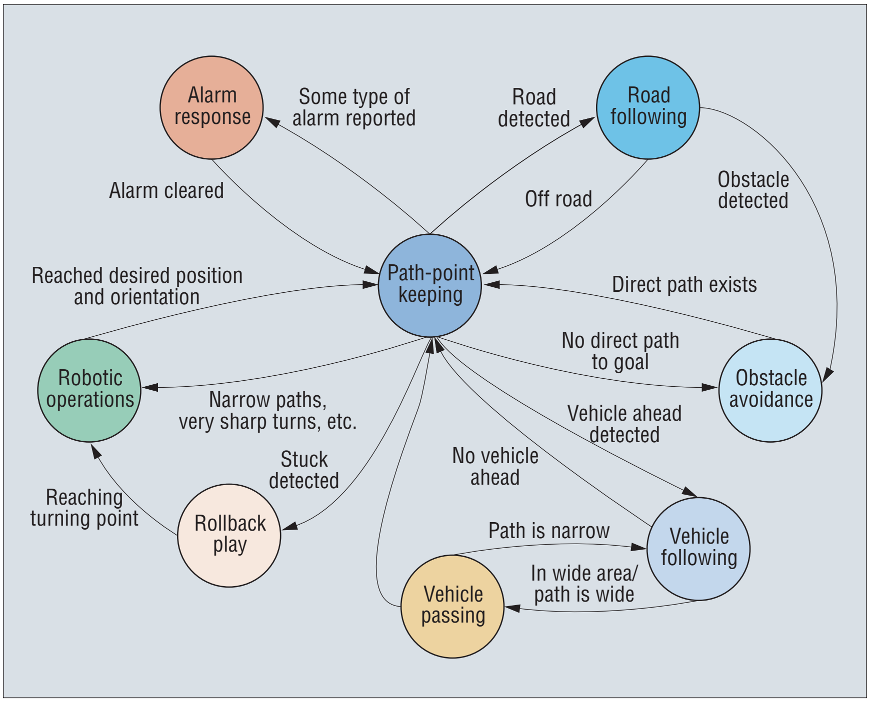

Figure 5. The finite-state-machine diagram of our system.

During the event, the robotic-operation mode triggered when the truck faced a concrete fence. After driving backward twice, the truck corrected its heading and successfully continued the course. This made TerraMax the only GC 04 vehicle that clearly demonstrated reverse driving.

TerraMax turned out to be one of just seven vehicles that completely traversed the QID course. It was the heaviest vehicle in the first Grand Challenge. Its huge size and robust construction allow it to negotiate any terrain very well. However, it had difficulty negotiating narrow paths or gates and making sharp turns, especially in the QID course. So, we had to build a special situation analyzer that let this giant make repetitive motions back and forth to deal with difficult sharp turns.

Performance in GC 04

Roughly four hours before the race, DARPA gave the participants a file of coordinates of 2,586 waypoints defining the vehicle's route. Following the route, TerraMax covered 1.2 miles within the required corridors and got stuck before a bush. The race log indicated that the communication network had malfunctioned and that the environment-sensing-and-fusion process didn't update the fusion map. So, the map indicated that there was always a bush before the truck, which prevented TerraMax from moving on. However, none of the other vehicles finished the course, either.

Through TerraMax's performance in the QID event and in GC 04, our system has proved that it can detect and avoid obstacles. Moreover, the path-point-keeping mode and part of the robotic-operation mode proved effective.

As GC 05 approaches, we are attempting to upgrade and stabilize the network and interprocess communication and are working on reasoning in more complicated environments. To increase navigation accuracy, we have also increased the fusion map's resolution. In addition, we are investigating new sensing technologies such as scanning radars. We also expect to test our system on a smaller platform than a large truck. ■

Acknowledgments

A large group of staff and students participated in TerraMax's development. We acknowledge Zhiyu Xiang, Charles Toth, Rong Xu, Hai Yu, Varuna Dilip, Andy Chien, Mark Soda, and Shokouh Shafiei for their contributions. We thank Ohio State University and the Oshkosh Truck Corporation for all their support and assistance.

Qi Chen is a PhD student at the Department of Electrical & Computer Engineering at Ohio State University. Contact him at chenq@ece.osu.edu.

Ümit Özgüner is a professor of electrical and computer engineering and holds the TRC Inc. Chair on Intelligent Transportation Systems at Ohio State University. Contact him at ozguner.1@osu.edu.

Keith Redmill is a research scientist in the Department of Electrical & Computer Engineering at Ohio State University. Contact him at redmill@ece.osu.edu.