SICE Journal of Control, Measurement, and System Integration, Vol. 1, No. 2, pp. 129–136, March 2008

Sensing, Control, and System Integration for Autonomous Vehicles: A Series of Challenges

Ümit Özgüner * and Keith Redmill *

Abstract : One of the important examples of mechatronic systems can be found in autonomous ground vehicles. Autonomous ground vehicles provide a series of challenges in sensing, control and system integration. In this paper we consider off-road autonomous vehicles, automated highway systems and urban autonomous driving and indicate the unifying aspects. We specifically consider our own experience during the last twelve years in various demonstrations and challenges in attempting to identify unifying themes. Such unifying themes can be observed in basic hierarchies, hybrid system control approaches and sensor fusion techniques.

Key Words : autonomous ground vehicles, car, intelligent transportation systems, offroad and urban automated vehicles, hierarchical systems, sensor fusion.

1. Introduction

There has been a continuous and gradually increasing interest in developing autonomous ground vehicles for different applications. We classify the types of applications as:

1. Automated Highway Systems (AHS)
2. In-city and urban driving
3. Off-road driving
4. Specialty applications

In this paper we shall consider the first three classes of systems only, as one may attempt to unify relevant sensing, control and integration issues in them. The fourth class includes a series of diverse applications and problems, for example closed deployment environments, tasks requiring special motion, docking, and convoying.

In reviewing the three application areas we shall rely on our experience in developing vehicles for each case. Details can be found in a series of publications. For AHS the reader is referred to our Demo'97 vehicle [1], shown in Fig. 1, and for off-road driving our DARPA Grand Challenge 2004 vehicle TerraMax [2] and 2005 vehicle ION, the Intelligent Offroad Navigator [3],[4], shown in Figs. 2a and b. In the present paper we shall also outline our vehicle ACT, the Autonomous City Transport, designed for city urban driving and developed for the DARPA Urban Challenge 2007 and shown in Fig. 2c.

Studies on the development of AHS usually advocate an ingress-egress pair during which the car will follow the assigned lane on the highway. Early testing and demonstration implementations, for example in Demo'97, assumed that the cars would be following specialized technological aids indicating the precise location of the car with respect to the lane. Figure 1 shows one technology that was advocated for location information with respect to the lane, a radar-reflecting stripe [5] that would indicate the distance from the center of roadway and the relative orientation of the car. We will mention other possible technologies later in the paper. It has to be pointed out that precision GPS and maps were not commonly available at that time. Today, it is assumed that precision maps would be available to the level of identifying individual lanes, and GPS reception would provide precise location information in real time.

* The Department of Electrical and Computer Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210, USA
E-mail: ozguner.1@osu.edu, redmill.1@osu.edu
(Received January 10, 2008)

Fig. 1 Two autonomous vehicles developed by OSU in Demo'97 following a radar- reflecting stripe and undertaking a pass.

Demo'97 was held on a 7.5-mile segment of the highway I-15 in San Diego. This segment was a segregated two-lane highway normally used for rush hour high occupancy vehicle traffic. Traffic flowed in the same direction in both lanes, and there were no intermediate entry and exit points. The curvature of the highway lanes was benign and suited for high speed (70 mph) driving, and other traffic was minimal to nonexistent. A general AHS would presumably have merge and exit lanes, but the single entry-exit aspect of Demo'97 made it a single activity: drive down the lane and possibly handle simple interactions with other vehicles. We shall subsequently call this behavior a meta state. Dealing with interchanges produced by entry and exit lanes would require other meta states.

Fig. 2 Autonomous vehicles developed by OSU: (a) TerraMax at GC 2004, (b) ION at GC 2005, and (c) ACT at UC 2007.

The DARPA Grand Challenges of 2004 and 2005 were both off road races. As such, the only behavior and thus the only meta-state required would be path following with obstacle avoidance from point A to point B. However, since there is no "path" or "lane" that can be discerned from a roadway, the only method of navigation is to rely on GPS and INS based vehicle localization [6] and a series of predefined "waypoints". Obstacle avoidance would be needed, as in an AHS, although in the less structured off-road scenario greater freedom of movement and deviations from the defined path are allowed. The Grand Challenge race rules ensured that there were no moving obstacles and different vehicles would not encounter each other in motion. General off-road driving would of course not have this constraint.

Finally, fully autonomous urban driving would introduce a significant number of meta states, situations where different behavior and different classes of decisions need to be made. The DARPA Urban Challenge, although quite complex, did have fairly low speed limits, careful drivers and no traffic lights. Visual lane markings were unreliable, and thus true to life, and the terrain was fairly flat, although some areas were unpaved, generating an unusual amount of dust and creating problems for some sensors.

We will make the claim that the basic problem definition will include a series of waypoints and a concept of "lanes". Although "lanes" are obvious in highway systems and urban routes, it is reasonable to assume that off-road environments also present a set of constraints that indicate the drivability of different areas and thus provide the possibility of defining lanes. A sketch of a vehicle on a path with waypoints and lanes is shown in Fig. 3.

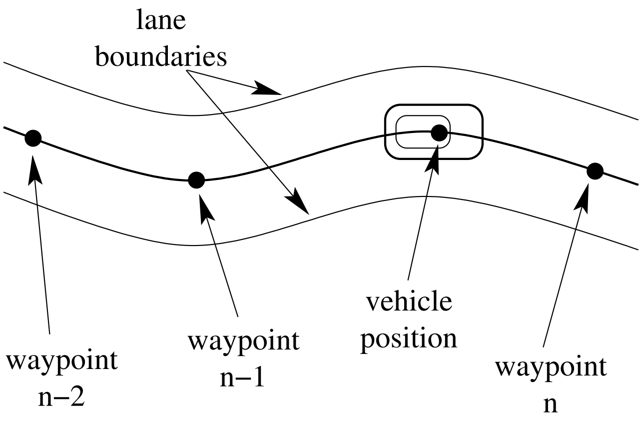

Fig. 3 Roadway with waypoints and lane markings.

In the subsequent sections we shall overview the architecture of the intelligence systems in the car's, sensing and control, specifically stressing the unifying aspects.

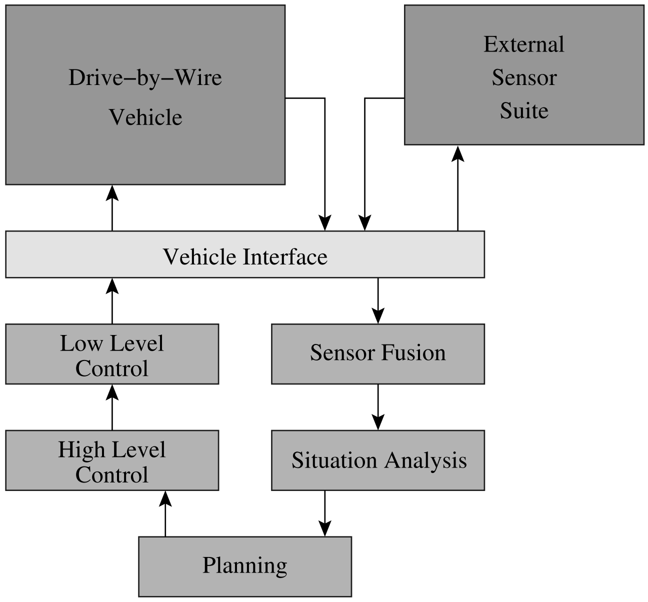

Fig. 4 The generic functional architecture for an automated vehicle.

2. Architecture and Hierarchy Issues

A generic functional architecture for an autonomous vehicle is given in Fig. 4. The "plan" can be simple or complex, but the overall configuration would cover all three application areas under consideration. Indeed Figs. 5 and 6 show the details of the hardware architecture for two different autonomous vehicles we developed in 1996 and 2007, more than ten years apart. Although some technologies have changed, and in spite of one being for AHS and the other for autonomous urban driving, one can note the similarities between the two configurations.

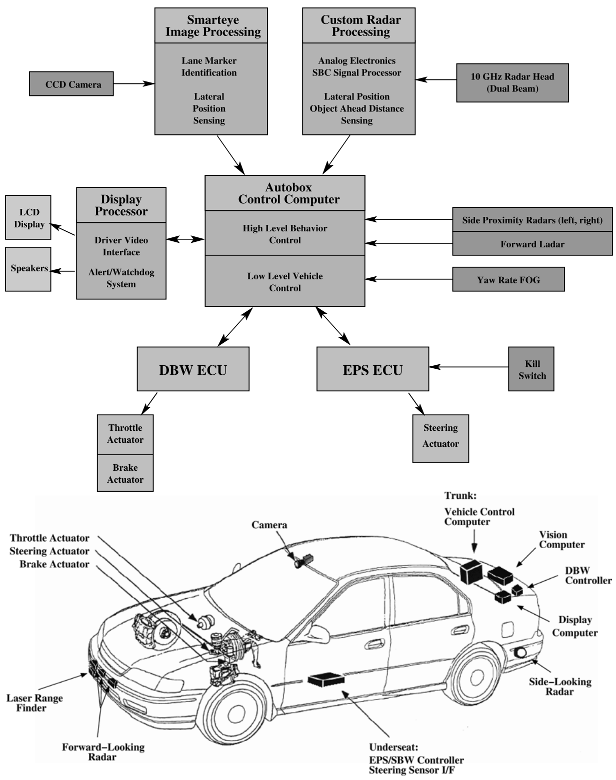

Fig. 5 The architecture for the OSU Team vehicle used in Demo'97.

Note that the Demo'97 car does not have a GPS system and relies totally on infrastructure based queues [5],[7] to find its position with respect to the roadway. The car has both a special (stereo) radar system that senses the radar-reflective stripe it straddles on the lane and a vision system that senses the white lane markers on both sides of the lane. Sensing of other cars on the roadway is accomplished with the radar and a separate LIDAR unit.

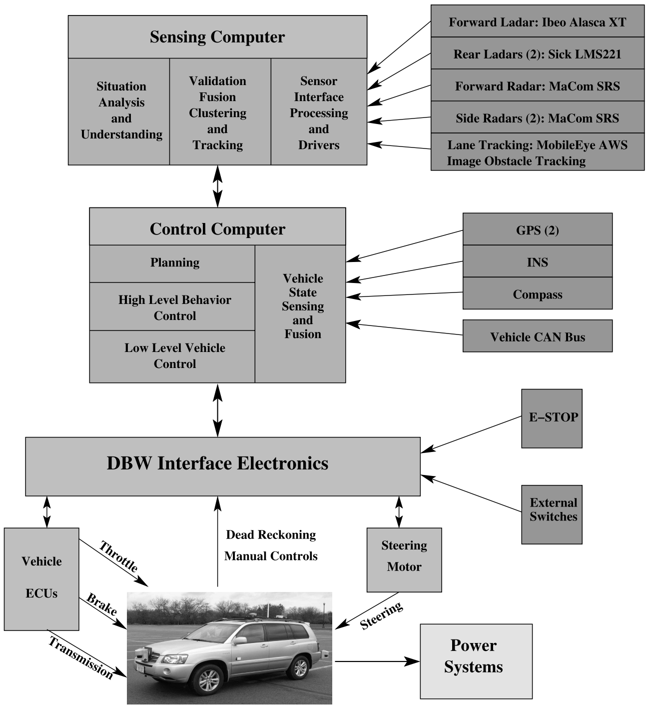

Fig. 6 The architecture of ACT, the OSU urban driving car used in the DARPA UC.

In the Urban Challenge car developed by OSU, shown in Fig. 6, the overall architecture is very similar. In this case, direct sensor based lane detection was developed but not fully integrated into the vehicle, although the sensor suite utilized would have allowed this. The vehicle, based on the DARPA definition of the Challenge, relied on high precision GPS signals and inertial and dead reckoning positioning technologies.

3. Sensing

3.1 Sensing the surroundings

We shall assume that "internal sensing" of the car, i.e. determination of vehicle speed, steering angle, engine and braking wheel torques, etc. can be accomplished and is similar for all three application domains. We shall thus concentrate on the external sensing, which would be different in each case.

The sensors we used in the three applications, and their effective coverage footprints, are given in Fig. 7 (a, b and c).

The set of sensors used on roadway vehicles depends on the infrastructure available. Basic lane detection can be accomplished by vision, assuming clear detection opportunity [7]. However, roadway installed aids with different technologies are certainly useful. These could be magnetic nails [8]–[10] or radar reflective stripes [1],[5]. Offroad vehicles, on the other hand, don't expect infrastructure, but do need more sensor capability, especially information regarding ground surface level detection, so as to compensate for terrain, bumps and holes in

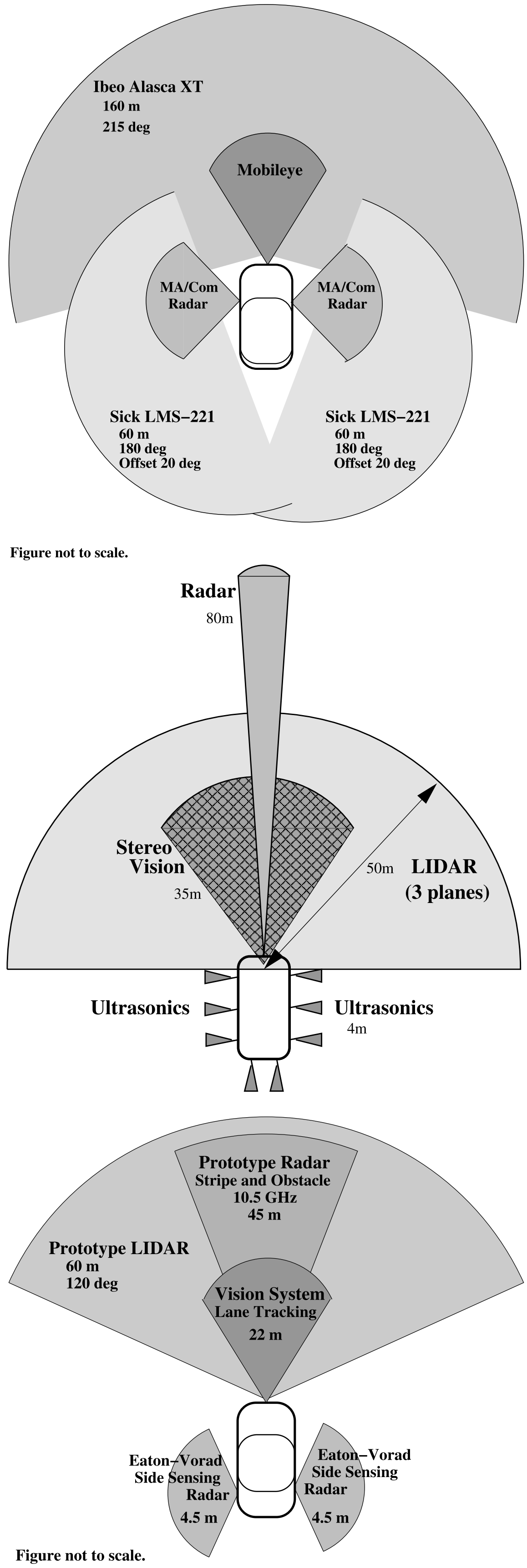

Fig. 7 Sensor suite and footprints: (a) OSU-ACT, (b) OSU-ION, (c) OSU Demo'97 cars.

the ground.

3.2 Sensor Fusion

There are some distinctions in considering highways, urban and off-road applications. We list some noteworthy items below:

- For pure highway applications, a very restricted approach may be appropriate. Since there are a number of distinct issues of concern, for example lane edge location and static or moving obstacles in relevant locations, one can simply fuse data related to specific tasks and not necessarily provide a complete and integrated representation of the world.
- For off-road applications, compensation for vibration and other vertical and rolling motions needs to be provided in software or hardware, for example using the IMU and sensor data to specifically generate a "ground plane" that can be referenced for sensor validation and fusion. Sensor adjustments are also required to deal with dust, rain, and changing lighting conditions.
- For domains where there are many moving obstacles (i.e. urban applications) one may need to "track" individual obstacles all the time.
- Specific operations (parking, dealing with intersections, entering/exiting highways, etc.) may use totally separate sensing and sensor architectures. This may include information provided by the infrastructure or other vehicles through wireless communication.

In general, we identify two approaches to sensor fusion: the grid, or occupancy map and a track identification approach.

In the grid map the sensing architecture/sensor fusion is established by developing a discrete map of the vehicle's surroundings. All external sensors feed into this map with obstacles sensed and related confidence levels. 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. The sensor fusion algorithm implemented on OSU's 2005 DARPA Grand Challenge vehicle used such a grid occupancy approach [4].

Due to the traffic situations where an environment is highly dynamic OSU-ACT has moved to an approach in which the sensor fusion algorithm is responsible for clustering and tracking all objects that are seen by the sensors.

The sensor fusion algorithm first uses information about the position and orientation of the sensors with respect to the vehicle to transform the returns into a vehicle centered coordinate system. Out primary sensors, the suite of LIDARS, provide a cloud of points representing each reflection from some surface of the targets in the world. Once the returns from the LIDARs are in vehicle centered coordinates, the position and orientation of the vehicle with respect to the world are used to transform the LIDAR returns into world coordinates. After the LIDAR returns have been transformed into world coordinates, they are clustered into groups of points. The clustering algorithm places the laser returns into a disjoint set data structure using a union find algorithm. Ultimately, clusters of laser returns are found whose members are not further than some maximum distance from each other.

Once the LIDAR returns have been clustered, the clusters are analyzed and those that can be identified as vehicles, based both on shape and motion, are classified as such and their centroids are estimated. All resulting clusters must be tracked using dynamic filters. Vehicle detections that are returned by the vision system or the radar sensors are matched to a LIDAR generated cluster by looking for a LIDAR cluster within some distance threshold. If no suitable matching cluster is found, the detections may update or initialize a track without a corresponding LIDAR cluster. The output of the sensor fusion algorithm is a list of tracks. Each of the resulting tracks has a position and velocity, and the general size and shape of the point cluster supporting the track is abstracted as a set of linear features.

4. The Hybrid System Formulation

A Hybrid States System (HSS) consists of two distinct parts, the Discrete State System (DSS), in which the state assumes only a finite and discrete set of values, and the Continuous State System (CSS), where the system state varies continuously. We have pursued a HSS approach for modelling autonomous ground vehicles in all three application domains considered here. The reason for selecting the HSS approach for the design of OSU-ACT in the Urban Challenge was the natural affinity of the system towards a situation-dependent solution for different scenarios. However even in the simpler off-road set-up of the Grand Challenge we claim HSS's provide a unifying modelling architecture. (Modelling examples and analysis tools for the general HSS architecture can be found in [11] and [12]. The OSU-ACT HSS can be seen in [13].) In order to be able to capture the situation-dependent nature of the challenge, the controllers can be layered into a Low-level Controller (LLC), and a High-level Controller (HLC). The "human driver" analogy is helpful in visualizing the distinct layers. The HLC is responsible for the conscious-level decisions such as lane changes,

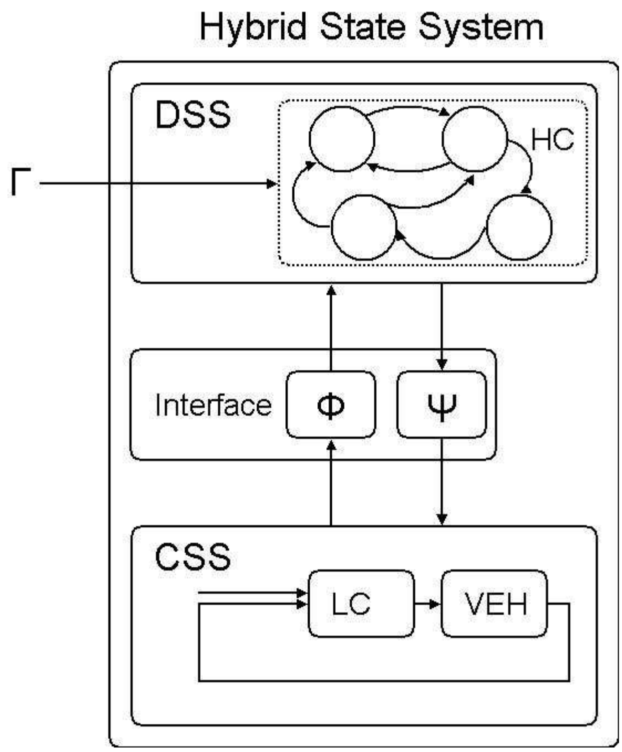

Fig. 8 HSS layout. The HLC is in the discrete-state system and the LLC is in the continuous-state system.

obeying the speed limits and handling intersections, while the LLC handles the subconscious control of steering to stay in the lane and throttle/brake control to maintain the speed. This autonomous vehicle structure is comparatively easy to model in a HSS representation. In Fig. 8, the discrete-state nature of the HLC is connected to the continuous-state system, which includes the LLC, through an interface Ψ and the events generated in LLC is communicated back to HLC through the second interface Φ.

The external input to the DSS, Γ, consists of discrete events generated by the sensing and situation analysis system of OSU-ACT and by the externally prescribed destinations and goals for the vehicle.

5. High Level Control

We indicated in the Introduction that we shall assume a basic setup with waypoints and lane boundaries. The off-road situation where the feasible path is understood to be a "lane" provides the simplest illustration of "meta-states" (see Fig. 9). A highway configuration would have multiple lanes all headed in the same direction, with standard lanes with equal width.

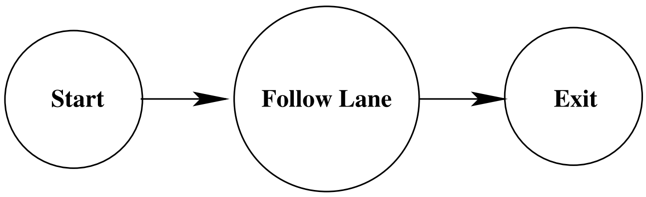

Fig. 9 A three meta-state model of transporting from start to end (exit).

On the other hand, both AHS and urban automated driving scenarios need concepts/tasks related to changing lanes. One possible meta-state configuration is shown in Fig. 10.

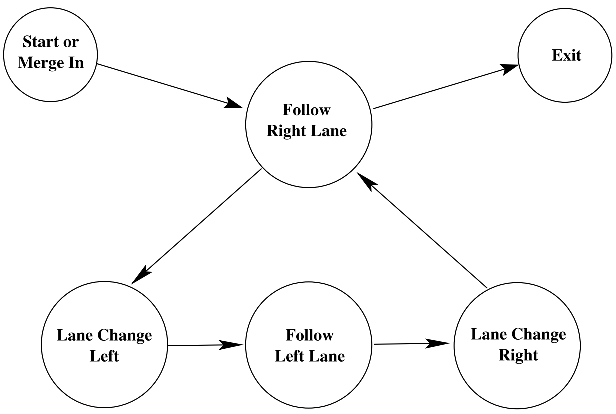

Fig. 10 A two-lane highway example with lane change.

The urban environment requires a much more complex state machine to represent the situations and control transitions. Figure 11 illustrates the meta-states used in OSU-ACT. It has to be pointed out that Fig. 11 hides many more substates underneath, as compared to the meta-states of Fig. 10. (See [13]).

6. Smooth Path Generation

One of the core problems in designing control for autonomous vehicles is the generation of a smooth path. The approach to the solution of this problem depends on how the heading correction needs to be accomplished. There are basically three approaches:

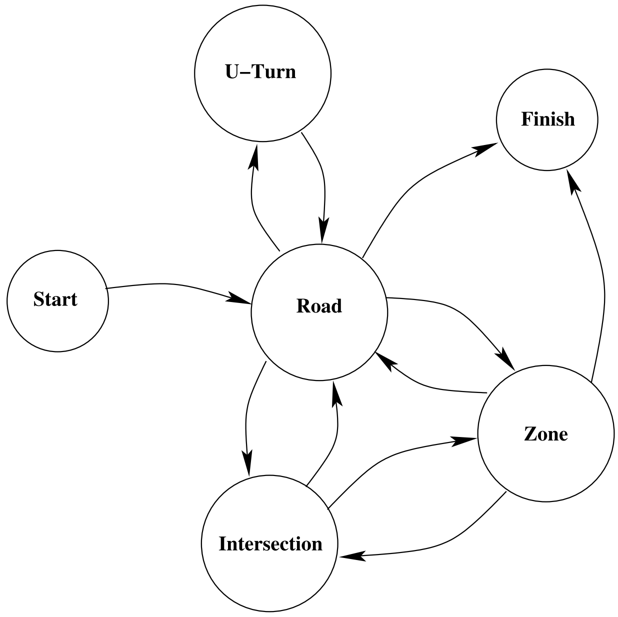

Fig. 11 Meta-states for the 2007 DARPA Urban Challenge situation.

6.1 Paths with respect to local measurements

Historically, in Automated Highway System studies, heading correction has been done based on lane detection. Lanes could be detected by vision technologies relying on lane edge markers [7], magnetic nails [8]–[10], radar reflective tapes [1],[5] or other such technologies where the infrastructure would support the sensing either actively or passively. There has been a preference to deal with so called Clothoid models, in which the road curvature C, which is the inverse of the road's radius of curvature, is the basic parameter

$$C = \frac{1}{R} \tag{1}$$

and is assumed to changed linearly with arc length

$$C = C_0 + \frac{dC}{dl} * l == C_0 + C_1 * l. \tag{2}$$

Thus with constant speed and steering wheel turn rates, ideally there will be no deviation from the ideal path [14]. It can be shown that $C_1 = 1/A^2$ is piecewise constant and A is the clothoid parameter.

For small angular changes we can get an approximation for y

$$Y = C_0 \frac{l^2}{2} + C_1 \frac{l^3}{6} \tag{3}$$

Again, for small deviations, this corresponds to a spline representation. This approach assumes the need for an independent representation of the roadway. Such a representation is not necessarily needed in implementation. Our activity in Demo'97 has been based on a look-ahead system simply measuring a heading error, and feeding back the error signal through a PID or P^2PID (squared error feedback) system without explicit modeling of the roadway in the controller.

6.2 Paths with respect to a map and GPS

In off-road and urban driving (especially based on the DARPA Urban Challenge definition), direct use of GPS and knowledge of a preplanned path has become the approach of choice. The preplanned path would be generated from knowledge of a map and a small set of selected waypoints from the map, as shown in Fig. 3. A smooth path, whose curvature is continuous, is generated to connect the waypoints.

The map database may be assumed to provide enough points to define the shape of the roadway, but these points are unevenly spaced and in straight sections may be widely separated. Thus, a local representation of the shape of the road is created and then subsampled to extract evenly and closely spaced (3 meter) waypoints for navigation and control purposes. Since we desire the vehicle to actually pass through each map given waypoint P_0, P_1, P_2, and P_3, an interpolating Catmull-Rom spline [15]–[17] is created as

$$f = 0.5((-t+2t^2-t^3)P_0 + (2-5t^2+3t^3)P_1 + (t+4t^2-3t^3)P_2 + (t^3-t^2)P_3), \quad 0 \le t \le 1 \tag{4}$$

Special situations, for example the absence of points at the beginning or ending of a road section or truly linear segments, must be identified and treated as special cases.

6.3 Paths with respect to desired objectives and constraints

For open areas (zones) and obstacle avoidance situations in the Urban Challenge we generated paths constructed using non-interpolating cubic Bezier splines, although other splines are certainly possible. Typically, two-dimensional cubic Bezier splines are defined by four points P_1, P_2, P_3, and P_4. The Bezier curve always passes through the first (P_1) and the last (P_4) points. Moreover, the Bezier curve is tangent to line segment P_1P_2 at point P_1 and similarly tangent to line segment P_3P_4 at point P_4. Thus, control points P_2 and P_3 can be chosen to change the slope of the end points. The parametric equations for the Bezier curve are given by

$$R(t) = At^3 + Bt^2 + Ct + P_1, \qquad 0 \le t \le 1 \tag{5}$$

$$A = P_4 - P_1 + 3P_2 - 3P_3 \tag{6}$$

$$B = 3(P_1 + P_3 - 2P_2) \tag{7}$$

$$C = 3(P_2 - P_1) \tag{8}$$

Using these properties of cubic Bezier splines, the end point P_1 is defined by the current vehicle position and the control point P_2 is selected to be of distance D_1 in the direction of the heading from the current position of the vehicle (P_1). The end point P_4 is defined by the destination position, and the control point P_3 is selected to be of distance D_2 towards the vehicle such that the vehicle will have an appropriate yaw angle at the destination. By changing the offset distances D_1 and D_2, more complicated paths can be formed. For global paths, offset distances are taken to be about third of the total linear distance from P_1 to P_4, while for local paths offset distances are fixed.

7. Vehicle Localization

A key element of autonomous vehicle work is vehicle localization. All aspects of the system, from sensor processing and fusion to navigation and behavioral decision making to low level lateral and longitudinal control require accurate vehicle position, velocity, and vehicle heading, pitch, and roll information at a fairly high update rate. Providing this information requires the use of multiple sensors, including multiple

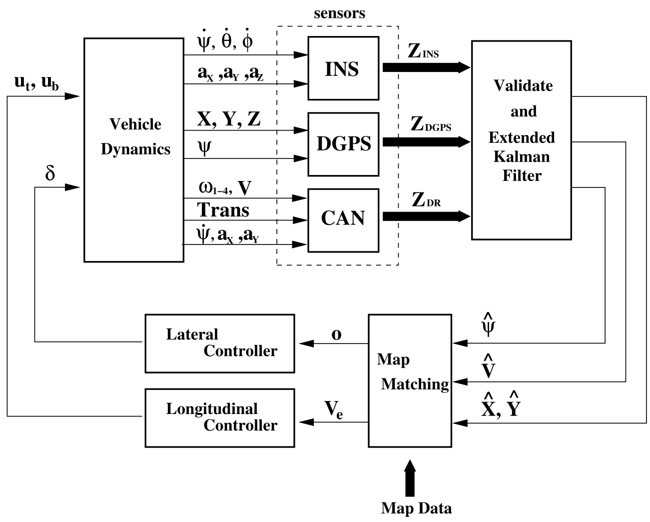

Fig. 12 Vehicle localization system.

Global Positioning System (GPS) receivers for redundancy, Inertial Measurement Units (IMU), and dead reckoning sensors (wheel speeds, transmission gear and speeds, throttle, brake, and steering wheel position) provided on the vehicle, a validation system to eliminate sensor errors, especially GPS-related step-change events caused by changes in differential correction status or the visible satellite constellation. To account for sensor errors, noise, and the different update rates of each sensor, an Extended Kalman filter is applied to generate the required state measurements. The overall layout of the localization system is shown in Fig. 12.

For the GPS, the observation equations appear as

$$Z_{gps}(k) = \epsilon_{gps}(k) + \nu + V_{gps}(k) \tag{9}$$

where the error $\epsilon_{gps}(k)$ represents a slowly time-varying bias in position and, to a lesser extent, heading angle and ν is the measurement noise. For the IMU, the observation equations appear as

$$Z_{imu}(k) = \epsilon_{B_{imu}}(k) + \epsilon_{P_{imu}}(k) + \nu + V_{imu}(k) \tag{10}$$

where the error $\epsilon_{B_{imu}}(k)$ is the almost fixed (given a constant operating temperature) sensor bias than can be estimated and canceled using the assumption that the error dynamics is given by $\dot{\epsilon_{B_{imu}}} = 0$, $\epsilon_{P_{imu}}(k)$ are the errors related to the vertical gyroscope's projection of the IMU data into an SAE (earth coordinate) navigation frame, which are primarily due to delay and damping effects, and ν is the measurement noise. Similarly, for the continuous state vehicle measurements, the observation equation is

$$Z_{dr}(k) = \epsilon_{dr}(k) + \nu + V_{dr}(k) \tag{11}$$

where the noise and bias from the vehicle sensors tends to be significantly larger than the higher grade GPS and IMU sensors.

8. Low Level Control

8.1 Command Interface

In a two-level control hierarchy as shown in Fig. 4, the low-level control receives operational instructions from the high-level control module. These instructions take the form of

1. A path to be followed, defined by a set of approximately evenly spaced control points

2. A desired speed
3. Commands to indicate starting and stopping
4. Special commands indicating motions that can be fully implemented at the lower level. Examples could be short movements along constant radius arcs, precision stops, etc.

The low level control will execute a given command set until either the command is completed and the vehicle is in a stationary state, or until the vehicle has driven off the end of the path provided, at which point the vehicle will be stopped, or until it receives a new command set.

Issues that distinguish an autonomous car from an indoor robot can be observed from the utilization of the above instructions. Cars would probably be using 1, 2, whereas indoor robots would mostly use 4.

8.2 Longitudinal Control

The interface and control of vehicle actuation is achieved by building or obtaining a drive-by-wire car. Our experience has been that a simple control algorithm, for example a set of PID controllers, is adequate to generate a virtual torque command to achieve the commanded speed, and a state machine is used to select between the use of throttle, active braking, or engine idle braking. Speed commands are modified to constrain the acceleration and jerk of the vehicle to preset comfortable limits. There may also be “emergency” deceleration modes that are less comfortable.

Urban driving, in contrast to highway or off-road driving, requires the vehicle to execute a precise stop. To accomplish this, the low level control determines the distance from the vehicle’s current position to a line drawn through the specified stopping point and perpendicular to the vehicle’s path of travel, taking into consideration the distance from the front bumper of the vehicle to its centroid. The speed of the vehicle is controlled to follow a specified nonlinear deceleration trajectory.

8.3 Lateral Control

The path that the vehicle is to follow is specified as a set of control points. This would be true in all three application domains we are considering. The lateral controller identifies both the current location of the vehicle and the look-ahead point a pre-specified distance ahead of the vehicle along its lateral axis and extracts a subset of control points closest to each location. Constant radius circles are fitted to the points in each subset and these circles are used to compute the vehicle offset distances from the path and to estimate desired yaw rates. Each subset of points also defines a desired yaw angle for the vehicle. The offset distances, yaw angle error measurements, and desired yaw rates can be used to generate a feedback signal for the steering controller. There are a number of algorithms that can be used in this control loop, and a simple PID controller with fixed gains is not enough to cover all possible driving and path-shape scenarios. The variations here are speed dependent and turn-radius dependent. See [18] and [19] for some examples.

9. Conclusions

In this paper we have attempted to provide a unified framework for considering autonomous ground vehicles. We have specifically considered AHS, off-road driving and urban driving and pointed out the similarities of:

1. Hierarchies and overall architecture
2. The finite-state machine based hybrid control system approach
3. Sensor suites
4. Sensor fusion approaches
5. Core control needs

Although we have only used vehicles that we developed through the last twelve years, we believe that these were quite representative in the approaches taken, if not the specific technologies or algorithms utilized.

We have not dwelt on vehicle-to-vehicle communication issues. Although both the PATH platoons in Demo’97 [8]–[10] and Japanese developers in 2000 [20],[21] used such communication in specific scenarios, a full autonomous vehicle demonstration/challenge that would illustrate coordination and collaboration is yet to come.

Acknowledgments

The authors would like to thank The Ohio State University College of Engineering, National Instruments, ST Microelectronics, and the Transportation Research Center (East Liberty, OH) for supporting this effort. Projects of this scope reflect the work and contributions of many individuals. We would like to specifically mention a few of these: Scott Biddlestone, Alex Hseih, Arda Kurt, John I. Martin, Roberto Mati, Ashish Shah, and Ahmet Yazici.

References

[1] K. Redmill and Ü. Özgüner: The Ohio State University Automated Highway System Demonstration Vehicle, *(SAE Transactions 1997): Journal of Passenger Cars*, SP-1332, Society of Automotive Engrs., 1999.

[2] Q. Chen, Ü. Özgüner, and K. Redmill: Ohio State University at the 2004 DARPA Grand Challenge: Developing a Completely Autonomous Vehicle, *IEEE Intelligent Systems*, Vol.19, No.5, pp. 397-412, Sept.-Oct. 2004.

[3] Q. Chen and Ü. Özgüner: Intelligent Off-road Navigation Algorithms and Strategies of Team Desert Buckeyes in the DARPA Grand Challenge ’05, *Journal of Field Robotics*, Vol.23, No.9, pp. 729-743, Sept. 2006.

[4] K. Redmill, J. I. Martin, Ü. Özgüner: Sensing and Sensor Fusion for the 2005 Desert Buckeyes DARPA Grand Challenge Offroad Autonomous Vehicle, *2006 IEEE Intelligent Vehicles Symposium*, pp. 528-533, June 2006.

[5] D. Farkas, J. Young, B. Baertlein, and Ü. Özgüner: Forward Looking Radar Navigation System for 1997 AHS Demonstration, *IEEE Conference on Intelligent Transportation Systems*, pp. 672-676, November 2007.

[6] K. Redmill, T. Kitajima, and Ü. Özgüner: DGSP/INS Integrated Positioning for Control of Automated Vehicles, *2001 IEEE Intelligent Transportation Systems Conference*, pp. 172-178, August 2001.

[7] K. Redmill: A Simple Vision System for Lane Keeping, *1997 IEEE Intelligent Transportation Systems Conference*, pp. 212-217, November 1997.

[8] W.-B. Zhang: National Automated Highway System Demonstration: A Platoon System, *1997 IEEE Intelligent Transportation Systems Conference*, November 1997.

[9] H.-S. Tan, R. Rajamani, and W.-B. Zhang: Demonstration of an Automated Highway Platoon System, *1998 Proceedings of the American Control Conference*, pp. 1823-1827, June 1998.

[10] S. Shladover: PATH at 20 - History and Major Milestones, *IEEE Transactions on Intelligent Transportation Systems*, Vol.8, No.4, pp. 584-592, Dec. 2007.

[11] M. Dogruel and Ü. Özgüner: Stability of Hybrid Systems, *Proceedings of the 1994 IEEE International Symposium on Intelligent Control*, pp. 129-134, Aug. 1994.

[12] K. Passino and Ü. Özgüner: Modelling and Analysis of Hybrid Systems: Examples, *Proceedings of the 1994 IEEE International Symposium on Intelligent Control*, pp. 251-256, Aug. 1994.

[13] A. Kurt and Ü. Özgüner: Hybrid State System Development for Autonomous Vehicle Control in Urban Scenarios, *2008 Proceedings of the IFAC World Congress*, to appear, July 2008.

[14] E. D. Dickmanns and B. D. Mysliwertz: Recursive 3-D Road and Relative Ego-State Recognition, *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol.14, No.2, pp. 199-210, Feb. 1992.

[15] E. E. Catmull and R. J. Rom: A Class of Localing Interpolating Splines, in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld (ed.), Academic Press, Orlanda, pp. 317-326, FL 1974.

[16] P. Barry and R. Goldsman: A Recursive Evaluation Algorithm for a Class of Catmull-Rom Splines, *Computer Graphics SIGGRAPH 88*, Vol.22, No.4, pp. 199-204, 1998.

[17] Y. Wang, D. Shen, and E. K. Teoh: Lane Detection using Catmull-Rom Spline, *Proceedings of the 1998 IEEE Intelligent Vehicle Symposium*, pp. 51-57.

[18] K. A. Ünyelioglu, C. Hatipoglu, and U. Özgüner: Design and Stability Analysis Of A Lane Following Controller, *IEEE Transactions on Control Systems Technology*, Vol.5, No.1, pp. 127-134, January 1997.

[19] C. Hatipoglu, Ü. Özgüner, and K. Redmill: Automated Lane Change Controller Design for Intelligent Transportation Systems, *IEEE Transactions on Intelligent Transportation Systems*, Vol.4, No.1, pp. 13-22, March 2003.

[20] S. Tsugawa, S. Kato, K. Tokuda, T. Matsui, and H. Fujii: A Cooperative Driving System with Automated Vehicles and Inter-Vehicle Communications in Demo 2000, *2001 IEEE Intelligent Transportation Systems Conference*, pp. 918-923, Aug. 2001.

[21] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii: A Cooperative Driving System with Automated Vehicles and Intervehicle Communications in Demo 2000, *IEEE Transactions on Intelligent Transportation Systems*, Vol.3, No.3, pp. 155-161, Sept. 2002.

Ümit Özgüner

Received the Ph.D. from the University of Illinois, Chicago in 1975. He has held research and teaching positions at IBM T. J. Watson Research Center, the University of Toronto, and Istanbul Technical University. Since 1981, he has been with The Ohio State University, where he is presently Professor of Electrical and Computer Engineering and holds the TRC Inc. Chair on Intelligent Transportation Systems. The team he coordinated participated successfully in the (US DOT supported) 1997 Automated Highway Systems Technical Demonstration in San Diego, where they demonstrated three fully automated cards doing lane-keeping, convoying, and passing using radar and vision-based guidance. Recently, he has been leading a group developing autonomous off-road and on-road vehicles and participated in the 2004 and 2005 DARPA Grand Challenge and the 2007 DARPA Urban Challenge. He is the author of over 300 publications which have appeared in journals, as book chapters, and in conference proceedings. He has been an invited plenary speaker on ITS topics for many international meetings in the U.S., Japan, China, India, Turkey, and Italy. His areas of research interest are in ITS, decentralization, and autonomy issues in large systems. Dr. Özgüner was the firt President of the IEEE ITS Council in 1999 and re-elected in 2000. He has also served the IEEE Control Society in many positions, and was an elected member of its Board of Governors (1999-2001). He participated in the organization of many conferences (including the IFAC World Congress) and has been the general chair of the 2002 CDC. He was the program chair for the 1997 IEEE ITS Conference, which he helped initiate, and served as the General Chair for the 2003 Intelligent Vehicles Symposium.

Keith Redmill

Received the B.S.E.E and B.A. Mathematics degrees from Duke University, Durham, NC in 1989 and the M.S. and Ph.D. degrees from The Ohio State University, Columbus in 1991 and 1998, respectively. He has been a Research Scientist in the Department of Electrical and Computer Engineering, The Ohio State University, since 1999. He is actively involved in research involving autonomous ground and aerial vehicles, and traffic and transportation systems. Recently, he led the system integration of autonomous off-road and on-road vehicles for the 2004 and 2005 DARPA Grand Challenges and the 2007 DARPA Urban Challenge. His areas of research interest include control and system theory, intelligent transportation systems, autonomous vehicles and robotic systems, real-time embedded systems, GPS and inertial positioning and navigation, transit and traffic monitoring, image processing, wireless digital communications for vehicles, sensor technologies, decentralized multiagent hierarchical and hybrid systems, and numerical analysis and scientific computer. Dr. Redmill is a member of the IEEE and SIAM.