Embedded Systems in Real Time Applications, Design & Architecture

[Student Paper]

A.L.SUSEELA (III-IT)
V.LALITH KUMAR (III-EEE)
S.T.I.E.T, GARIVIDI, VZM DT.

Motivation

Necessity is the mother of invention and embedded systems are inventions that were fuelled by the idea of making pre-programs to perform a dedicated narrow range of functions as part of large systems. Usually with minimal end user interactions, the 'giant leap technology' in future embedded systems is based on instruction-oriented design but not on design-oriented instructions. Embedded systems and real time operating systems (RTOS) are fast achieving ubiquity, blurring the lines between science fiction and hard reality. In general an RTOS has the following futures: 1. multitaskin; 2. process threads that can be prioritized; 3.A sufficient number of interrupt levels. An embedded system is any device controlled by instructions stored on a chip. These devices are usually controlled by a microprocessor that executes the instructions stored on a ROM chip. Embedded systems are used in navigation tools like global positioning system (GPS), automated teller machines (ATMs), networking equipment, digital video cameras, mobile phones, aerospace applications, telecom applications, etc. We concern ourselves with the development and implementation of model-based, real-time, embedded, hybrid control software. In particular, we target intelligent cruise control applications, including Adaptive Cruise Control (ACC), in which a forward-looking range sensor (radar or Lidar, usually) is used to follow a vehicle, and Cooperative ACC (CACC), a variation in which wireless communications are used to supplement the forward looking sensor. We discuss modeling on automated vehicles. Our approach emphasizes the maintenance of a single model throughout the development process, with particular emphasis on "tight-loop."

Introduction

Embedded systems and Real Time Operating systems (RTOS) are two among the several technologies that will play a major role in making these concepts possible. A large number of people are already depending on operating systems for real time applications, these 'eyes in the sky' are now going to make an impact on our every day lives in a more significant manner. What kind of help will these 'embedded systems' render unto humankind in the future? Even Nostradamus would have been hard

pressed to answer this question. Embedded systems are pre-designed without connections and operate as per the required task. But in operating systems instruction is design-oriented. These systems are basically platform-less systems. Embedded systems are the unsung heroes of much of the technology we use today -- the video game we play, or the CD player or the washing machines we use employ them. Without an embedded system we would not even be able to go online using modem.

Design Orientation

Embedded systems are usually low cost and are easily available off the shelf for most applications. They usually have low design risks, since it is easy to verify the design using tools fueling the growth of embedded systems.

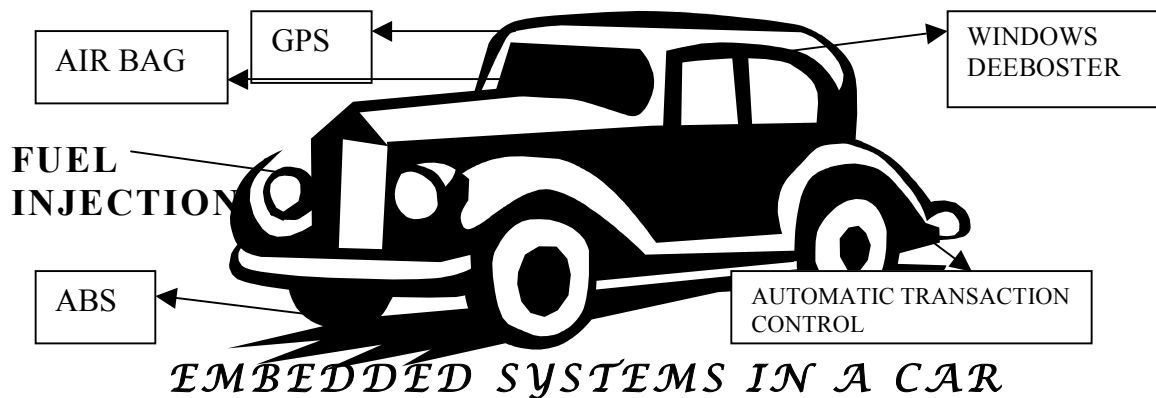Embedded systems have received a major shot in the arm as the result of three developments:

- The first was the development of standard run-time platforms like java, which enabled their use in myriad ways that were unimaginable in the past.
- The second was the coming together of embedded systems and the Internet, which made possible the networking of several embedded systems to operate as part of a large system across networks.
- The third was the emergence of several integrated software environments, which simplified the implementations of these applications.

During operation, the design structure may be changed as per our tasks. For example, consider two transistors; we can mould them using other passive elements as emitter coupled circuit, Darlington pair, etc., as per instruction.

Real Time Applications

Automobiles:

Almost every car that rolls off the production line these days makes use of embedded technology in one form or the other; most of the embedded systems in automobiles are rugged in nature, as most of these systems are made up of a single chip. No driver clashes or 'systems busy' conditions happen in these systems. Their compact profiles enable them to fit easily under the cramped hood of a car. These systems can be used to implement features ranging from adjustment of the suspension to suit road conditions and the octane content in the fuel to antilock braking systems (ABS) and security systems.

	GPS		WINDOWS DEEBOSTER
AIR BAG			

FUEL
INJECTION

ABS

AUTOMATIC TRANSACTION
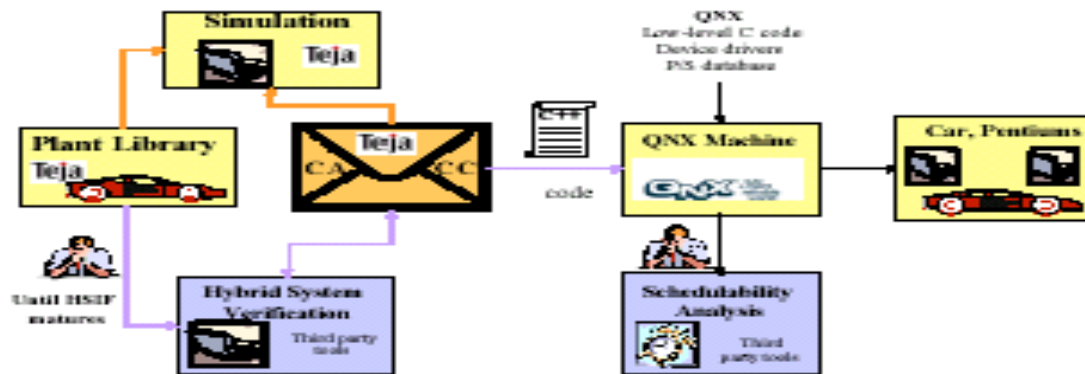CONTROL

EMBEDDED SYSTEMS IN A CAR

Embedded systems can also make drive-less vehicle control a reality. Major automobile manufacturers are already engaged in work on these concepts. One such technology is Adaptive Cruise control (ACC) from Ford. ACC allows cars to keep safe distances from other vehicles on busy highways. The driver can set the speed of his car and the distance between his car and others. He can over side the system anytime he wants by braking. Each car with ACC has a microwave radar unit or laser transceivers fixed in front of it to determine the distance and relative speed of any vehicle in its path. The ACC computer constantly controls the throttle and brakes of the car.

Another revolution is the way Internet services will be integrated into the car. So when you drive past your mechanic's, you will be reminded that that your engine oil needs a refill, and when you cross the city limits, the toll will automatically get deducted from your bank account. And while passing the shopping mail, your PDA, which is connected to the Net via the car, will inform you about a new scale. In fact, the automatic to;l deduction concept is already in effect in several countries around the globe.
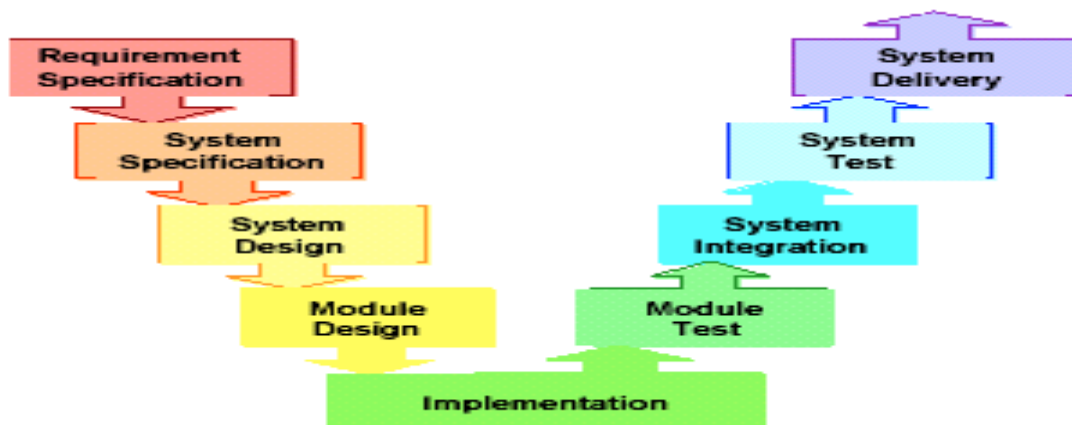

Hybrid verification of the controller and analysis of timing properties are conducted through the use of third party tools.

The approach is applied to Adaptive Cruise Control (ACC) and Cooperative ACC systems. While regular cruise control systems track a desired vehicle speed, Adaptive Cruise Control (ACC) systems adapt their behavior if there is a vehicle ahead on the roadway, and follow the leader vehicle at a driver requested time gap using line-of-sight sensors such as radar and/or Lidar. When there is no "leader" vehicle present, ACC defaults to conventional cruise control and reverts to the driver-set speed. ACC systems are now available on several production cars, including the Nissan Q45 and FX45, the Mercedes S-class, the Lexus 330 and 430, the Audi A8, and select Jaguar and Cadillac models. These production ACC systems obtain their distance and closing rate information about the leading vehicle through the use of their forward-looking sensor. These sensors are typically subject to noise, interference, false alarms and dropouts, and their use requires heavy filtering. This in turn introduces delays into the system, and limits the ability of the ACC-equipped vehicles to follow the leader vehicle closely or respond quickly to change in its speed. A variant of this is Cooperative ACC (CACC), where the forward-looking sensor is complemented by a wireless communication link that provides hop-by-hop, leader-to-follower updates of critical information. Such a system can be designed to follow vehicles with higher accuracy and faster response than traditional ACC systems, and should allow for freeway throughput capacity increases. In addition, the CACC system can be designed to have proven string stability, so it could contribute to dampening shock waves in the freeway traffic stream.

Vehicle Model And Controller Design

The vehicle model used for controller development is an eleven-state model, which includes vehicle state dynamics, throttle and brake system dynamics, a two-state model for the spark-ignition engine including external data maps which require interpolation, and models of the torque converter, transmission and wheel slip, as shown in the figure.
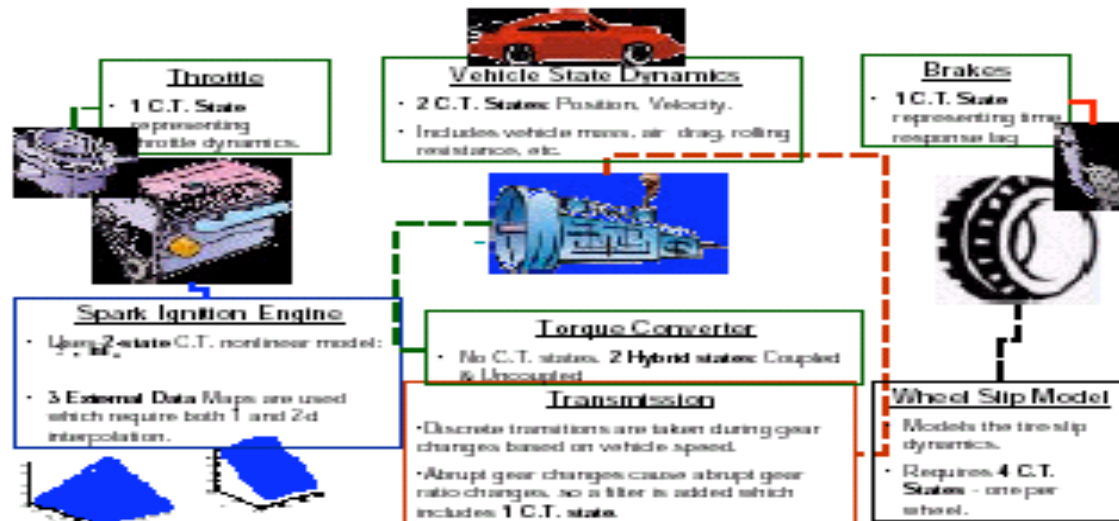
The vehicle state dynamics have two continuous states, vehicle position and velocity, and consider vehicle mass, air drag and rolling resistance. The throttle and brake dynamics are both first-order, with one continuous state for each representing actuator dynamics for the throttle and time response lag for the brakes. The controller design process stems from system requirements. Vehicles may be heterogeneous, that is of different types, makes and models. The controller was split hierarchically between an upper level controller that has several modes, namely cruise control (CC), adaptive cruise control (ACC) and coordinated adaptive cruise control (CACC). In ACC mode we use only information from the host vehicle's forward-looking sensors, and in CACC mode we supplement this information with data from the wireless communication system.
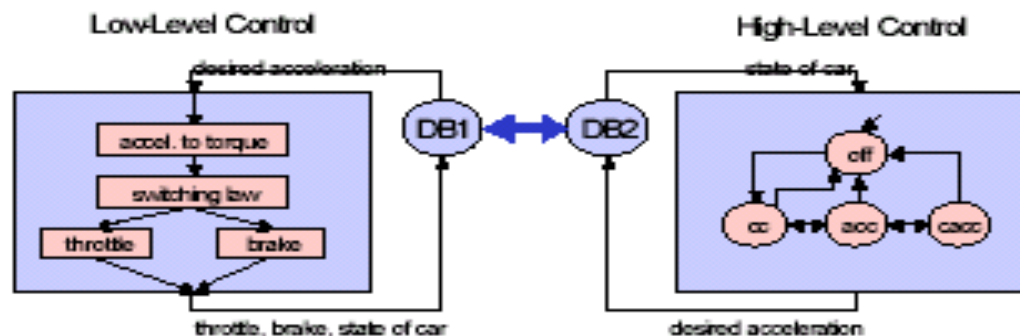
The upper control generates a desired host vehicle acceleration, which is sent to the lower-level controller. The lower-level controller converts this desired acceleration to a desired torque, then chooses whether to apply the brakes or throttle, and in what amount. Both controllers are run on separate control computers.

In the following equations, the following variables are used:
• F_a is the aerodynamic drag force

- Mrr is the rolling resistance moment
- Rg is the gear ratio (related to engine and Vehicle speeds)
- βades is the desired synthetic acceleration
- τct is the control torque
- h is the effective wheel radius

Throttle control is used if:

$$\beta_{ades} + R_g (M_{rr} + F_a + mg\sin\theta) - \tau_{ct} \geq 0$$

For throttle control, the desired torque is computed as:

$$\tau_{ades} = \beta_{ades} + R_g (M_{rr} + hF_a + mgh\sin\theta)$$

Brake control is used if:

$$\beta_{ades} + R_g (M_{rr} + F_a + mg\sin\theta) - \tau_{ct} < 0$$

For brake control, the desired torque is computed as:

$$\tau_{ades} = -\left| \frac{\beta_{ades} + \tau_{ct}}{R_g} \right| - M_{rr} - hF_a - mgh\sin\theta$$

1) Throttle control
From the desired torque, the desired throttle angle is computed using an engine map.

2) Brake control
From the desired torque, two different brake control strategies have been implemented. In the first strategy, the master cylinder pressure is controlled. A pressure regulator valve controls the pressure applied on the hydraulic actuator. Seal friction exists in the master cylinder and the actuator, and a small amount of hysteresis is present in the pressure regulation valve. The friction is modeled as hyperbolas from various points in the hysteresis loop and can be written as
$$Pmc = g (u)$$
Feed-forward plus proportional feedback control is used, as developed. The control law can be written as:

$$u_b = g^{-1}(P_{mc}) + k_b (P_{mc\_des} - P_{mc})$$

Where ub is the applied command input to the brake solenoid valve, Pmc_des the desired master cylinder pressure, Pmc the measured master cylinder pressure, and kb>0 a feedback gain. In the second brake control strategy, the wheel brake pressure is controlled, and the brake system is modeled. The control law uses dynamic surface control and can be written as:

$$P_{mc} = P_w - \left[\frac{\dot{P}_{wdes} - \lambda_b(P_w - P_{wdes})}{\frac{\partial P_w}{\partial V} C_q} \right]^2$$

Where Pw_des is the desired wheel pressure, V is the volume of displaced brake fluid, Pw the pressure at the wheel, Cq a flow coefficient,

Cruise Control Law:

The purpose of cruise control is to maintain a desired velocity. A vehicle may be in cruise control mode if it is not equipped with ACC or CACC, has no vehicle immediately in front of it or has at least 100 meters of clearance to the preceding vehicle, or by decision of the human driver. The controller uses a feedback and feed-forward control law of the form:

$$a_d = \dot{v}_d - k(v - v_d)$$

Where: a_d is the desired acceleration of the vehicle, v is the speed of the vehicle, v_d is the desired speed of the vehicle and k is a gain set to 0.75.

ACC and CACC Law:

The control law for ACC and CACC is identical. The main difference between both modes is in the sensor fusion, and in the quality of the state information. Also, the operating logic is different in both modes. The purpose of an ACC or CACC law is to regulate the range between vehicles to a user-selected value, and to adjust the vehicle speed to the speed of downstream traffic. Velocity dependent (headway) control is used, with:

$$R_{des} = \lambda v + \mu$$

Typical values for headway time range form 1.8 seconds to 0.7 seconds. The control law was designed using sliding control, where a surface is usually defined as a function of the error, derivatives of the error and/or integrals of the error. The surface is defined such that the state will exponentially decay along the surface to the desired point. The input is chosen to guarantee that the state will converge and stay on the sliding surface. Error e is defined as:

e = R − Rd, where R=x1-x2.

The sliding surface control is derived in two different ways, which basically lead to the same control law.

Feedback linearization:

$$a_2 = a_1 - \ddot{R}_d + k_d \dot{e} + k_p e$$

Double first order method:

$$a_2 = a_1 - \ddot{R}_d + (\lambda + k)\dot{e} + k\lambda e$$

Once again, the first two terms in the control law are feed forward, and the last two are feedback. Both control laws are in essence equivalent.

Software Development Process

   A model-based approach is used throughout the control software development. Switching conditions from one mode to the next (for example ACC into CACC) were designed by hand. The chosen architecture can then be simulated, and C or C++ code can be generated for each task independently. This allows maintaining a single model containing all of control and software information. The code that is generated for the controller's interfaces with legacy code, such as device drivers, driver display units etc, through the use of a shared-memory database on the "publish-and-subscribe" model. On the experimental test vehicles, all of the software is run on Pentium computers running the QNX4.25 real-time operating system.

Experimental Platform:
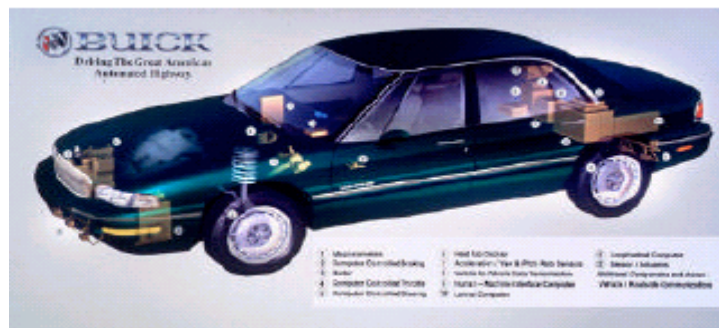
They are equipped with throttle, brake and steering actuating systems, as well as with numerous sensors, including accelerometers, wheel speed sensors, engine speed and manifold pressure sensors, as well as magnetometers that are used as part of the lateral control. In addition, both radars and the Lidar described above were mounted to the front bumper of the vehicles. There are two control computers located in the trunk. Both run the QNX 4.25 operating system and communicate over serial port connections. The computers run a host of tasks necessary for automated control of the vehicles, including reading sensor data and writing to actuators, control computations such as those described above for the ACC/CACC system and low-level controllers, and tasks pertaining to driver display information. There are about 30 different tasks running on the most heavily loaded of the control computers, and timing is fairly critical as human test drivers are in the cars during runs and their safety is paramount.

Other Applications

Wired Wearables

A mobile phone in the form of a ring or earring? What about cool sunglasses, with streaming video displays built into them? All these can soon be a reality. Embedded systems have a small footprint and consume very little power, which makes them ideal for wearable computing applications. The minimal system requirements of the devices ensure that the hardware is almost microscopic.

IBM is already working on the prototype of a mobile phone that can be worn as jewelry. The components of the phone will be distributed among different pieces of jewelry -- earring, necklace, ring and bracelet.

The phone is likely to have blue tooth capability built into it. The earring will have embedded speakers and will act as the receiver. The necklace will have embedded microphones that will act as mouthpiece users can talk into. IBM calls the ring part of the phone the 'decoder ring'. Light emitting diodes (LED's) will flash to indicate an incoming call. The ring will also have features that will enable it to be programmed to flash different colours for a particular user or to indicate the importance of a call. A video graphics array (VGA) will be built into the bracelet, which will display the name and plans to incorporate voice recognition technology for dialing a number. The phone may also have features to indicate new E-mail.

Pacemakers
Imagine a time when body transplants like cardiac pacemakers will be able to monitor & manage themselves remotely. These systems will be so compact that the patient wouldn't even be aware that they are embedded in his body, and developments are pointing towards the use of pacemakers that can be transplanted in or near the heart itself. The pacemaker will be able to monitor parameter like blood pleasure blood flow, pressure rate temperature, etc., using microsensors placed in various parts of the body. The capability will enable the pacemaker to automatically vary its operation to suit the changing body conditions. It will also transmit data using micro sensors planted in various parts of the body. This capability will enable the pacemaker to automatically vary its operation to suit the changing body conditions. It will also transmit data using wireless transmission.

Embedded technology advances less transmission is likely to be done by a transmitter implanted near the surface of the skin. In case in an abnormality is detected. The doctor will be able to take remedial action even from remote locations.

A variety of operating systems are available for use with embedded computers. Many of them are not true real-time operating systems (RTOS), as they do not support the precise scheduling of tasks and predictable reactions times of real-time events. A true RTOS must suspect prioritized, pre-emptive scheduling.

Embedded computers perform their jobs by executing software instructions. Unlike desktop computers, the user has little or no

information on what is happening -- code is executed automatically in response to 'real time' events. For example, when an intruder opens a door connected to a security system, the microprocessor turns on an alarms, dials the number of the security company, and transmits an alarm signal. Smarter systems analyze data about the intrusion and turn on only if the intruder has human attributes. Other than setting the alarm and checking for messages, the user of the alarm has no control over the software being executed.

Conclusion

This paper presents the use of a model-base approach to the development of real-time, embedded, hybrid control software. The concepts are illustrated with a scenario involving speed profile tracking and vehicle following applications for using the cruise controller. Robotic technologies such as range, velocity and acceleration measurements, and their processing and fusion were used as part of the system. In addition, vehicles can present very nonlinear behavior, especially at low speeds, and their control presents a formidable challenge. The problem domain of intelligent cruise control applications has been described in detail, along with control and software development methodologies. All these application areas are just tiny drops in the big ocean of embedded systems technology. These proverbial Davids are all set to conquer a world that is forbidden territory for the popular desktop OS Goliaths -- so hold your breath and wait for the fireworks to come. They are sure to blow our mind.

REFERENCES:

[1]   Personal communication, Ken Henry, GM Research.
[2]   http://www.wikipedia.org/wiki/Embedded_system, from Wikipedia, the free Encyclopedia
[4]   D. Cho and J.K. Hedrick, "Automotive Engine Modeling for Control", ASME Journal of Dynamic Systems, Measurement and Control, December 1989, Vol. 111, pp. 568-576.
[5]   Girard, A.R., Spry, S.C., Kretz, P.R. Dickey, S.R., Empey, D.M., Misener, J.A., Variaya, P.P. and Hedrick, J.K., "Vehicle-to-Vehicle Open Experimental Platform Reference Manual."
[6] www.teja.com