Let's put the Car in your Phone!

Martin Geier, Martin Becker, Daniel Yunge, Benedikt Dietrich,
Reinhard Schneider, Dip Goswami, Samarjit Chakraborty
Institute for Real-Time Computer Systems, TU Munich, Germany
lastname@rcs.ei.tum.de

ABSTRACT

Today high-end cars have extremely complex E/E architectures – with 50–100 electronic control units (ECUs), connected by communication buses like CAN, FlexRay and Ethernet. They are used to run several (control) applications with many million lines of code. We propose a radically new architecture where all these applications are instead run on a mobile phone being carried by the driver. The car now has a considerably simpler architecture with few or no ECUs, using RF links to connect sensors and actuators to the mobile phone with a powerful multicore processor. We discuss the advantages and challenges and describe a small prototype implementation with an adaptive cruise control application.

Categories and Subject Descriptors

C.3 [**Special-purpose and application-based systems**]: Real-time and embedded systems

General Terms

Design, Experimentation, Performance, Reliability

Keywords

Automotive, Mobile Phone, Consolidation

1. INTRODUCTION

Over the past one decade automotive electrical and electronic (E/E) architectures have become extremely complex and distributed, to the extent that this growth in complexity is no longer sustainable. High-end cars now contain 50–100 electronic control units (ECUs) or processors. They are connected by several different communication buses like CAN, FlexRay and Ethernet, with their wiring running into several kilometers in length. Such an architecture is then connected to various sensors and actuators and is used to run several control applications related to safety-critical, driver assistance and comfort functionalities. In total, all applications sum up to many million lines of code. However, the hardware infrastructure including the cabling weighs several tens of kilograms and has an impact on fuel or energy consumption, which is especially critical for electric vehicles. Such an architecture and the growth in the number of ECUs can be partially attributed to the need for more computational power; e.g., cars now have a rich set of driver assistance functions that process inputs from multiple cameras, radars and LIDAR sensors. However, this growth may also be attributed to the business models followed by the automotive industry, where a network of Tier 1 suppliers provide ECUs with specific functionalities that the OEM then integrates.

But increasingly it is being realized that this practice cannot continue for long – the large number of ECUs and complex networks are difficult to maintain, verify and debug, they lead to compatibility problems during architecture extensions and they increase the weight and the cost of the car. This has led to a lot of recent work to enable a move from such *federated* to *integrated* architectures [7], where a number of functionalities are integrated into a single, possibly multicore ECU, thereby reducing the number of ECUs. The use of software platforms like AUTOSAR [1] and virtualization [6] to provide isolation between applications is being explored to realize such integrated architectures. Another major drawback of current automotive architectures is that its hardware gets outdated within 1–2 years, given the fast pace at which technology and, thus, processors are currently developing, while a typical car has a lifetime of 10–15 years. This is problematic since now most innovation in modern cars is in their electronics and software.

Phone in the loop: As a possible solution to these problems, we propose a radically new automotive E/E architecture where all applications (including control) are instead run on a smartphone being carried by the driver. The car will now have a considerably simpler architecture with few or no ECUs, and all its sensors and actuators will instead communicate with the mobile phone via RF links. While currently available smartphones have quadcore processors, 100-core processors in such phones are not unforeseeable in the future, which can then replace the 100 ECUs currently in the car. Further, smartphones today already have multiple communication interfaces like Bluetooth and WiFi and this is likely to grow, too. As the driver enters the car, the phone will connect the available sensors and actuators in the car and run the computations that currently run on the ECUs. While a number of challenges need to be overcome in order to realize this, some of the advantages are obvious. The phone can be replaced easily, as we already know today. Software upgrades will also be simplified. The setup will be significantly more cost-effective, since the compute power in the ECUs currently is not used when the car is not being driven. With these ECUs being "buried" in the phone, they can be used for other purposes when the car is not in use. Further, reliability of ECUs is a major concern today, since they are exposed to harsh environments and extreme temperatures. The phone on the other hand is always in the passenger cabin. The phone may also carry driver preferences/profiles such as seat adjustments, display layouts and even emails, entertainment-related information and daily schedule. Many high-end cars today allow a smartphone to be connected to the car for accessing emails, music, etc. These functionalities along with personalization information can now be more seamlessly integrated with the car [9]. A video illustrating the concept and our prototype is available online [10].

2. TECHNICAL CHALLENGES

A number of technical challenges need to be overcome to realize our proposed architecture, some of which even require new technological developments. First, current mobile phones cannot support the computational bandwidth required to replace all the ECUs. Thus, the path towards our proposal can also have a number of intermediate stops, i.e., few ECUs and the phone might be more realistic now. Furthermore, the phone could serve as a gateway to a cloud-based compute infrastructure, which we discuss in Section 4. Virtualization techniques that are already being investigated

DAC '13, May 29 - June 07 2013, Austin, TX, USA.

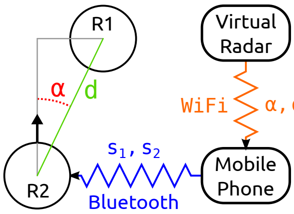

Figure 1: Prototype setup

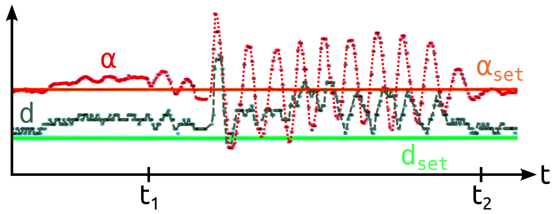

Figure 2: Oscillation due to RF interference

in the automotive context may be implemented inside the phone in order to isolate safety-critical applications from others. We believe that this can already be realized to a large extent and is a good research problem. The next obvious challenge is that of communication between the mobile phone and the sensors and actuators in the car. While Bluetooth is currently used for in-vehicle communication, it is restricted to the infotainment domain. Almost all current in-vehicle communication is performed over wired networks [8]. New reliable, energy-efficient, high-bandwidth, short range communication technologies will be needed to realize our proposal and a significant amount of research effort should go in this direction. While security issues are largely ignored in current automotive architectures and are only being studied recently, e.g., for firmware updates [5], they will become much more important in the proposed architecture. Finally, software and control systems development will require attention, too. Since sensor and control messages will now be transmitted over wireless networks, they will suffer variable delays and probably also loss. Therefore, controller design techniques will have to be more aware of computation and communication resource constraints [4] than they are today.

3. PROTOTYPE IMPLEMENTATION

As a starting point for investigating some of the challenges and potential solutions associated with the proposed architecture, we have implemented an *adaptive cruise control* (ACC) application on a robot. The setup (depicted in Figure 1) is that of a *Robotic Road Train* recently reported in [3], where a semiautonomous vehicle follows another vehicle in front, while maintaining a safe distance to it.

The first robot (R1) follows an arbitrary trajectory. The second robot (R2) receives two wheel speed values (s_1, s_2) from the mobile phone via a Bluetooth link. A virtual radar sensor (implemented using a Vision Tracking System) determines distance d and angle α between the robots. Using WiFi, this information is sent to the ACC application running on the mobile phone which finally closes the control loop and enables the second robot to follow the first.

This system is susceptible to delay, jitter and loss from various sources, such as RF interference and scheduling on the mobile phone. To show the effects of RF interference between sensor and controller, we set up a second WiFi link on the same channel as the sensor connection. By adding load we increase the collision probability of the two WiFi links until packet loss occurs which leads to control instability. Figure 2 shows set-points and actual values of d and α as recorded by the mobile phone during system operation. The second WiFi link was loaded for a short interval centered around t_1. It can be seen that for $t_1 < t < t_2$ both d and α oscillate around their respective set points which can be observed as snaking movement as shown in our video [10].

The problem of network failures shown in the above small-scale example will become even more prominent in our proposed architecture. To overcome this and the aforementioned problems, advanced design techniques will be required in multiple domains. Increased reliability and fault tolerance of the wireless communication network may be achieved using *link diversity schemes* and *integrated control strategies*.

The former uses multiple diverse wireless communication paths to avoid network failures. Modern mobile phones support several RF interfaces, each of them having different transmission characteristics (link rate, latency, resilience) resulting from parameters such as frequency, bandwidth and transmission power. Based on availability of individual communication paths and on current control requirements, a *communication management unit* may dynamically modify the information routing during runtime.

The latter takes the underlying communication network into account to better deal with its delay, jitter and loss. These factors lead to multiple *switching* control subsystems and create a major challenge in designing *platform-aware* control algorithms to stabilize the resulting switched system.

4. OUTLOOK

Moving a bulk of the computation away from automotive-specific ECUs and onto a smartphone carried by the driver is a paradigm shift in automotive architectures. But it has a number of advantages and certainly can be realized given the current rate of progress in processor architectures and wireless communication technologies. Eventually, the phone can serve as a gateway to a cloud that will support computations from many cars. This will also result in new business models where along with the car, one needs to additionally buy a service, i.e., computation time on the cloud. Thus, the latest hardware/software technologies will be available to a car and largely free it from maintenance requirements. This will be especially useful for electric cars where most of the functionalities will anyway be implemented in software and the need to cut down weight and energy consumption is severe. However, such a scheme will also need several developments in cloud computing, such as support for real-time computing and low-delay communication. The design of control algorithms will also be influenced, with cloud-based implementations of compute-intensive control algorithms (e.g., visual servoing) already being studied [11]. Such a setup will also open up the possibility of new advanced driver assistance functionalities and autonomous cars [2] that require too much computation (and hence energy and cooling infrastructure) to be feasibly implemented *inside* a car.

5. REFERENCES

[1] Automotive open system architecture, "AUTOSAR Specification", release 4.0, 2012.

[2] D. Bernstein, N. Vidovic, and S. Modi. A cloud PAAS for high scale, function, and velocity mobile applications - with reference application as the fully connected car. In *Int. Conf. on Systems and Networks Comm. (ICSNC)*, 2010.

[3] E. Coelingh and S. Solyom. All aboard the robotic road train. In *IEEE Spectrum*, November 2012.

[4] D. Goswami, R. Schneider, and S. Chakraborty. Co-design of cyber-physical systems via controllers with flexible delay constraints. In *ASP-DAC*, 2011.

[5] M. S. Idrees, H. Schweppe, Y. Roudier, M. Wolf, D. Scheuermann, and O. Henniger. Secure automotive on-board protocols: A case of over-the-air firmware updates. In *Nets4Cars/Nets4Trains*, 2011.

[6] A. Masrur, S. Drössler, T. Pfeuffer, and S. Chakraborty. VM-based real-time services for automotive control applications. In *RTCSA*, 2010.

[7] M. D. Natale and A. L. Sangiovanni-Vincentelli. Moving from federated to integrated architectures in automotive: The role of standards, methods and tools. *Proceedings of the IEEE*, 98(4):603 – 620, 2010.

[8] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert. Trends in automotive communication systems. *Proceedings of the IEEE*, 93(6):1204 – 1223, 2005.

[9] D. Siewiorek. Generation smartphone. In *IEEE Spectrum*, September 2012.

[10] DAC2013 WACI Video: Let's put the Car in your Phone!, December 2012. `http://www.rcs.ei.tum.de/en/research/byop/`.

[11] H. Wu, L. Lou, C.-C. Chen, S. Hirche, and K. Kühnlenz. Cloud-based networked visual servo control. *IEEE Tran. Industrial Electronics*, 60(2):554 – 566, 2013.