

Embedded Systems and Software Challenges in Electric Vehicles

Samarjit Chakraborty¹, Martin Lukasiewicz², Christian Buckl³, Suhaib Fahmy⁴
Naehyuck Chang⁵, Sangyoung Park⁵, Younghyun Kim⁵, Patrick Leteinturier⁶, Hans Adlkofer⁶

¹ TU Munich, Germany, Email: samarjit@tum.de

² TUM CREATE, Singapore, Email: martin.lukasiewicz@tum-create.edu.sg

³ Fortiss GmbH, Munich, Email: buckl@fortiss.org

⁴ NTU, Singapore, Email: sfahmy@ntu.edu.sg

⁵ Seoul National University, Korea, Email: {naehyuck,sypark,yhkim}@elpl.snu.ac.kr

⁶ Infineon Technologies AG, Germany, Email: {patrick.leteinturier,hans.adlkofer}@infineon.com

Abstract—The design of electric vehicles require a complete paradigm shift in terms of embedded systems architectures and software design techniques that are followed within the conventional automotive systems domain. It is increasingly being realized that the evolutionary approach of replacing the engine of a car by an electric engine will not be able to address issues like acceptable vehicle range, battery lifetime performance, battery management techniques, costs and weight, which are the core issues for the success of electric vehicles. While battery technology has crucial importance in the domain of electric vehicles, how these batteries are used and managed pose new problems in the area of embedded systems architecture and software for electric vehicles. At the same time, the communication and computation design challenges in electric vehicles also have to be addressed appropriately. This paper discusses some of these research challenges.

I. INTRODUCTION

Nowadays most innovations within the automotive domain are driven by embedded systems and software solutions. Many of these innovations like anti-lock braking systems, electronic stability control, or emergency brake assistants significantly reduce vehicle accidents and increase safety. On the other hand, embedded systems increase the driving comfort with driver assistance functions like adaptive cruise control. Furthermore, infotainment systems and telematics increase the user acceptance and contribute to the value of modern cars. It can be observed that the costs for embedded solutions in vehicles are growing rapidly while mechanical engineering based solutions are stagnating in importance. In [1], it is stated that the prorated costs for embedded solutions in vehicles grew from 1 percent in 1980 over 7 percent in 1990 to 22 percent in 2007. It is projected that the importance and costs of embedded systems and software in electric vehicles will grow much further.

One major innovation within the automotive domain is the introduction of electric vehicles. It is widely understood that the approach of replacing the engine of a conventional car by an electric engine is only an intermediate solution on the way to a fully customized electric vehicle. Therefore, the embedded

systems and software challenges in electric vehicles go beyond the engine and energy control. A redesign of the communication and computation architecture in electric vehicles entails several opportunities, but also bears many challenges. Current electrical/electronic architectures (E/E architectures) consist of up to 100 Electronic Control Units (ECUs) for top-of-the-range vehicles. These complex E/E architectures are becoming a barrier to innovation and a redesign is becoming necessary to handle this complexity.

Currently, it is still the case that most suppliers provide functionality as a hardware device. This is also one of the main reasons why the number of ECUs has grown rapidly within the last years. However, this approach will not scale as more functionalities are implemented with electronics and in software, as will be the case in electric vehicles. Hence, a paradigm shift from distributed hardware to distributed software and services becomes necessary. For this purpose, flexible E/E architectures and standardized software component approaches for vehicle software like AUTOSAR (AUTomotive Open System ARchitecture) [2] will become necessary for electric vehicles.

To cope with the complexity of potential safety and comfort functions in electric vehicles, novel communication and computation approaches have to be investigated. It is projected that the communication will shift towards time-triggered systems based on FlexRay or Real-time Ethernet. A homogeneous core network or backbone will significantly reduce the networking complexity. For computation support, novel computational devices and programming paradigms have to be considered. While the automotive industry already started considering multi-core systems, alternative solutions might be graphic processors or reconfigurable hardware to cope with the computational demands of active safety functions.

The electric power train also has to be designed and controlled to achieve the highest possible efficiency. To achieve significant energy savings, a distributed embedded control approach becomes necessary to control the power management of the entire vehicle. This is a challenging task because multiple energy source like the batteries, solar panels, or regenerative braking have to be taken into account. As energy storage, hybrid batteries are a promising solution to cope with

the energy requirements of electric vehicles. For this purpose, control strategies for these batteries will become necessary that take into account different driving patterns.

In the rest of the paper we briefly discuss some of these challenges. In Section II we discuss system-level E/E architecture and software challenges in electric vehicles. In Section III we talk about computation-related challenges and which devices might be appropriate to cope with the demands of upcoming applications. In Section IV we outline in-vehicle communication challenges and discuss which protocols might be appropriate for electric vehicles. In Section V we outline design challenges for hybrid electrical energy storage systems and the first results from this domain.

II. E/E ARCHITECTURES AND SOFTWARE CHALLENGES

Today, embedded systems and software in vehicles is essential for the competitiveness of the automotive industry. Their most notable effects are to improve driving performance and comfort, and to enhance both passive and active safety. But these effects go further in electric vehicles. Here information and communication technology (ICT) becomes the foundation of the driving functions themselves. For these reasons, architectures and technologies for vehicle ICT cannot be viewed merely as a frame for gradual evolutionary innovations, as they were before. Instead, they must be revised so farsightedly that they can perform their indispensable future role in the evolution of the vehicle.

Over the past 30 years, embedded systems and software have made significant innovations in automotive industry: from the anti-lock braking system in 1978 to electronic stability control in 1995 and emergency brake assist in 2010. In total, embedded systems and software contributes some 30 to 40 percent of the total value in automotive systems. Nevertheless, the use of ICT lags well behind the technical possibilities. Safety is achieved mainly by passive safety measures; proactive safety functions (such as emergency brake assist) that make heavy demands on ICT are treated with great caution. In other areas such as infotainment or telematics, the ICT architecture in cars has not been keeping pace with developments in other sectors such as consumer electronics.

The major reasons for the above situation is the evolutionary grown complexity. Today, top-of-the-range vehicles have between 70 to 100 ECUs. The control devices are networked using a highly complex tree of cables, using multiple bus systems for example, for the engine chamber, chassis, passenger compartment and infotainment as well as using different communication protocols. The result has been that in today's vehicles, this complex ICT architecture is increasingly becoming a barrier to innovation. McKinsey estimates that car manufacturers could earn 15-20% more EBIT (Earnings before Interest) if there are no warranty costs from software faults [3].

A. Future ICT Architecture

In the evolutionary development of vehicle architectures, as Figure 1 shows, there is an evident trend towards architectures becoming much more complex than would actually be necessary for the realizing the gain in functionality. This results in the problem that new functions become more expensive to integrate, and the innovation trend therefore suffers. Only a

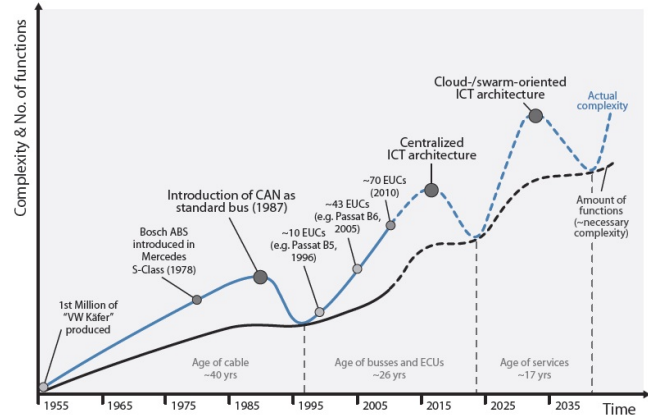


Fig. 1. Evolution of complexity in the automotive domain.

substantial revision of the hardware and software architecture and a technology leap can bring the actual complexity back down to only the necessary level.

This process has already been observed in the automotive industry in the past. To reduce emissions and improve comfort, in the 1980s it was necessary to use microcontrollers more widely. Complexity relatively quickly became a big problem, because it was almost impossible to cable all these electronics modules together. A solution came from communications buses like the Control Area Network (CAN) bus.

Today's ICT architecture again faces similar problems, but in this case because of the large number of ECUs. For electric vehicles the following main design principles for future E/E architectures and software design are being proposed: (i) *Flexible E/E architecture*: Future E/E architectures will be based on a centralized or hierarchical topology based on scalable ECUs to execute all hardware-independent software functions. (ii) *Highly-integrated mechatronic components*: Sensors and actuators will become smart components that communicate over standardized data interfaces with the E/E architecture. (iii) *Standardized communication backbone*: A time-triggered homogeneous in-vehicle core network has to be provided that can guarantee a required quality of service. (iv) *Operating system / middleware*: New technology have to be developed and deployed to take extra-functional properties into account.

In contrast to standards like AUTOSAR [2] that focus predominantly on the functional integration, the future middleware technology must support guarantees with respect to extra-functional properties such as timing, fault-tolerance, and security. Furthermore, plug & play capabilities must be supported to integrate new functions easily in the car.

B. Studies and Prototypes

During the last couple of years, several studies have been made for electric vehicles. The study [4] is based on 240 interviews world-wide with experts from all relevant fields including amongst others OEM (original equipment manufacturers), OES (original equipment suppliers), political and consumer organizations. The goal was to investigate the potential

AUTOSAR 4.0 recently introduced mechanisms to describe the timing and also fault-tolerance requirements. However these techniques must be further improved.

changes in ICT architectures induced by the shift to electric vehicles.

Based on the results of the study, an experimental platform was implemented following the suggested ICT architecture. The most challenging hardware setup has been selected to show the capabilities of future ICT architectures: a car with four independently controllable driving / steering units interconnected only by a network. The hardware setup of the evaluation platform is based on the eCorner concept from Siemens VDO [5]. It consists of four eCorner modules, which can be controlled independently of each other. Each of the eCorner modules is composed of a drive and a steering motor. No mechanical axes are used for the synchronization of the eCorner modules. Instead, the whole control system is based on a communication system (X-by-Wire). No mechanical fallback solutions were integrated in this car, so that the whole reliability and safety must be ensured by software.

III. COMPUTATION CHALLENGES

As mentioned before, lately, the number of ECUs in top-of-the-range vehicles has grown to almost 100 since new applications were generally introduced as hardware devices. It is projected that the number of novel driver assistance or active safety functions will grow further and it is expected that electric vehicles will require much more advanced computation to manage the automotive systems including the electrical power train. This will require more complex ECUs and a paradigm shift from the development of distributed hardware to a centralized and hierarchical ECU architecture that runs hardware-independent software. Compared to current ECUs with low clock speeds, novel computational devices that run a multitude of applications in parallel will have a significantly higher computing power. This will require novel operating systems and middleware concepts.

Available candidates for computational devices in electric vehicles are multi-core processors systems, reconfigurable hardware, or graphic processor systems. The suitability of these novel hardware solutions for electric vehicles is an open question, including the interaction with application software and operation systems. On the other hand, applications in vehicles have very different requirements regarding the performance, reliability, and costs. For instance, infotainment applications have to provide a high performance. On the other hand, for passive and active safety functions the reliability aspects have priority.

A. Multi-core Systems

Car manufacturers and suppliers currently favor multi-core ECUs as the computing devices for next-generation vehicles, see [6]. One advantage of multi-core systems is that existing software of multiple conventional single-core ECUs can be executed in parallel without any software changes using an appropriate operating system. This ECU consolidation can help to reduce the number of ECUs efficiently within the near future and lead to a centralized E/E architecture in electric vehicles.

At the same time, multi-core systems are at the lower end of the price scale. This is a very important factor since monetary

costs of a system in the automotive domain is very important because high quantities are produced and delivered.

B. Reconfigurable Hardware

Field Programmable Gate Arrays (FPGAs) are programmable hardware devices that offer the high performance of custom hardware, with some of the flexibility of software. They consist of logic and routing resources that can be configured to implement any arbitrary circuit. Hence, a data path tailored to a specific application can result in orders of magnitude speedup over general-purpose processors for computationally-intensive tasks.

As FPGAs are dynamically programmable, they can be reconfigured to different applications in-situ. For example, a driver-assistance system might use different algorithms at day and night, or when driving through tunnels. Rather than have all the algorithms implemented in custom hardware at the same time, an FPGA-based system can select the most suitable algorithm, reconfigure, and continue processing. Currently, driver assistance and entertainment are two main areas where high-intensity computation is required. Processing videos at real-time rates and the extraction of useful information from sensors requires computational power that can be provided efficiently by FPGAs. A key issue FPGAs can address within the context of electric vehicles is their comparatively low power consumption in case of complex applications. FPGAs allow these compute-intensive applications to be implemented at smaller size, lighter weight, and much lower power, all factors that make them attractive for incorporation in electric vehicles.

Upcoming FPGA devices promise even more exciting possibilities. Xilinx is soon to release the Zync series of FPGAs, that include an independently-functioning dual-core ARM Cortex A9 processor. Alongside it, is a large area of reconfigurable fabric for implementing custom hardware, with very high interconnectivity bandwidth between the two. This promises to make FPGAs even more attractive in systems where software capability is also a requirement.

With the newer FPGAs housing dedicated processors, it is worth exploring how applications currently implemented solely on ECUs can be virtualized across the two resources. For example, at certain times, a particular (e.g., control) function may be in a simple mode that requires little processing, and hence can be run on the processor. External conditions may result in this function changing to a complex mode that requires hardware acceleration. Hence, it should be migrated to a hardware version running in the FPGA, with this happening seamlessly. This requires a virtual abstraction layer that can monitor system performance, respond to external events, and manage migration of tasks.

IV. IN-VEHICLE COMMUNICATION CHALLENGES

An in-vehicle communication network for electric vehicles has to satisfy very high availability and reliability requirements. At the same time, protocols have to guarantee real-time properties for safety-critical and advanced driver assistance functions. Possible candidates for next-generation networks in electric vehicles are FlexRay [7] and (real-time) Ethernet [8] as well as the combination of these two protocols. Novel concepts

and scheduling approaches have to be investigated to provide the required safety and flexibility for upcoming systems in electric vehicles.

In the following, bus protocols for in-vehicle networks of electric vehicles, the device integration challenges, and the development of time-triggered automotive systems are discussed.

A. Bus Protocols Candidates

Future automotive communication networks will play a major role in the integration of automotive electronic components. Currently, the Controller Area Network (CAN) [9] bus is still the predominant bus system in the automotive domain, mostly due to low costs and ease of configuration. It is projected that the CAN bus will be replaced in electric vehicles since it has only a limited bandwidth and cannot cope with the requirements of upcoming advanced driver assistance and safety functions. Existing heterogeneous in-vehicle communication systems with multiple protocols and gateways might be replaced by a homogeneous core network. The two major candidates for the core network protocol of electric vehicles are FlexRay and Ethernet.

Both FlexRay and Ethernet have a significantly higher bandwidth than the CAN bus. The FlexRay protocol allows a time-triggered communication which is necessary for several control functions with strict real-time requirements. In a dual channel mode, the FlexRay protocol even allows to double the bandwidth. Other extensions like switched FlexRay [10] further increase the effective bandwidth and reliability of FlexRay networks. Due to the standardization of FlexRay for the automotive domain it is a strong candidate for upcoming in-vehicle networks that require a significantly higher bandwidth than existing systems.

Ethernet [8] might be an alternative to cope with the high data volume of upcoming electric vehicles. To ensure real-time properties, special implementations of the Ethernet protocol become necessary [11]. The electromagnetic compatibility and exposure to very high temperatures for Ethernet in the automotive domain is not as well researched and understood in contrast to FlexRay [12]. Currently, Ethernet is only considered for non-critical multimedia applications as a compensation for the cost-intensive Media Oriented Systems Transport (MOST) [13].

B. Integration Challenges

Modern vehicles comprise dozens of spatially distributed embedded systems. These ECUs are often developed by different suppliers with most diverse requirements on safety, reliability, costs, and computational power. The amount of communication in in-vehicle networks is constantly increasing due to the growing number of safety, comfort, and entertainment functions. As a result, the integration of distributed components into a modern vehicle is a challenging task. With the introduction of software component concepts like AUTOSAR, the complexity in electric vehicles will even increase with the growing degree of freedom. At the same time, multi-core ECUs will increase the amount of data that is sent and received by a single device.

For the swift integration of the ECUs in a in-vehicle network, a unified definition of interfaces and real-time requirements becomes necessary. At the same time, tools for the verification of real-time requirements have to be used to ensure a correct functionality. Also diagnosis and flashing issues of these ECUs also have to be considered at design time.

While the interaction of different ECUs is a well-known integration problem, the interaction with external communication partners will become an additional issue. To increase the safety, reduce the energy consumption, and improve the route planing, a car-to-x and car-to-car communication will be introduced in electric vehicles. Besides the multitude of advantages of such a communication with the environment, security issues have to be addressed and solved appropriately at the level of ECUs and in-vehicle networks.

C. Time-triggered Systems

With the introduction of the FlexRay bus in production vehicles, the design paradigm in the automotive domain might shift towards time-triggered systems in electric vehicles. In case the ECUs are not synchronized with the bus, undesired delays and jitter in the communication are possible. Therefore, a synchronous time-triggered system is desirable [14] where all network participants comply with a global schedule. Functions that are executed periodically benefit from such a time-triggered architecture, leading to an improved control quality due to the deterministic behavior and minimal jitter. At the same time, simulation, integration, and testing efforts are reduced significantly due to the predictability of the system. These properties make such time-triggered systems ideal candidates for next-generation automotive architectures. In particular, such architectures have a growing need in electric vehicles due to the increased software and electronics content.

For the success of a new concept for in-vehicle communication, the tool support and flexibility are highly important. Currently, the design of time-triggered systems is more complicated than event-triggered distributed systems based on the CAN bus. It is therefore necessary to provide design methodologies and tools for the next-generation in-vehicle networks to increase the acceptance among car manufacturers.

V. BATTERY AND SUPERCAPACITOR HYBRID STORAGE

As the cruising range is one of the most important metrics of electric vehicles, the efficiency of the energy storage is crucial. It is well known that batteries are subject to the *rate-capacity effect* such that a larger charging/discharging current results in less amount of usable battery energy.

The maximum charging current of most secondary batteries is much smaller than their maximum discharging current, and so the rate-capacity effect is said to be more serious for charging operations. Therefore, power capacity of batteries is often short during the charging process while driving, that is, regenerative braking for electric vehicles.

Supercapacitors, however, have a very high power capacity both for the charging and discharging operations and are not subject to the rate-capacity effect. Such characteristics of supercapacitors have inspired battery and supercapacitor *hybrid storages*, shown in Figure 2 [15], [16], [17], for electric vehicles. There are two goals to achieve in the hybrid storage:

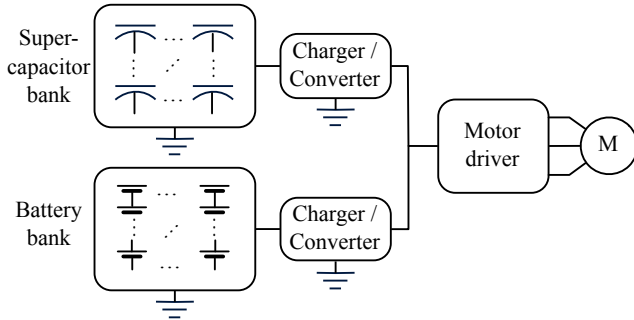


Fig. 2. The topology of target electric vehicle energy storage.

maintaining a stable state of charge (SoC) of the supercapacitors and achieving high energy efficiency of the hybrid storage. As supercapacitors have relatively much smaller energy capacity in comparison with batteries in a typical hybrid storage, previous work mostly focused on supercapacitor SoC management [15], [16], [17].

Recent hybrid storages perform active SoC control of the supercapacitors using a charger. The charger efficiency should be carefully considered because it is largely variable by the supercapacitor SoC and the magnitude of the charging current [18]. These make it even more difficult to determine when and how to charge/discharge the supercapacitor and battery banks. This section introduces a method that deals with these two important considerations, which have not been thoroughly studied yet.

A. Charge Management of Hybrid Storages in Electric Vehicles

Since supercapacitors have cycle efficiency advantage over batteries, charging the supercapacitor bank as much as possible and later charging the batteries may result in a higher energy efficiency [17]. Speed-sensitive charge management of supercapacitors may enhance the efficiency of the hybrid storage system [16]. When the vehicle speed is low, there are higher chances of acceleration which requires high-power discharging, and thus a high supercapacitor SoC helps. When the vehicle speed is low, there are higher chances of regenerative braking, which generates high power from the traction motor, and thus low supercapacitor SoC helps. It is beneficial to limit the current flow to and from the batteries and to handle excess current with the supercapacitors, since it mitigates the rate-capacity effect of the batteries [15].

However, most of the previous works does not seriously consider one important factor: the power converter efficiency and the maximum power transfer from the traction motor during regenerative braking. Thus, we jointly consider the efficiency of both the batteries and supercapacitors, and the converter efficiency so that the maximum power can be transferred from the traction motor to the hybrid storage.

B. Maximum Power Transfer Tracking for Regenerative Braking with a Hybrid Storage

We aim at electric vehicle power optimization from the perspective of HEES (hybrid electrical energy storage) system optimization [19]. The cycle efficiency of an HEES system

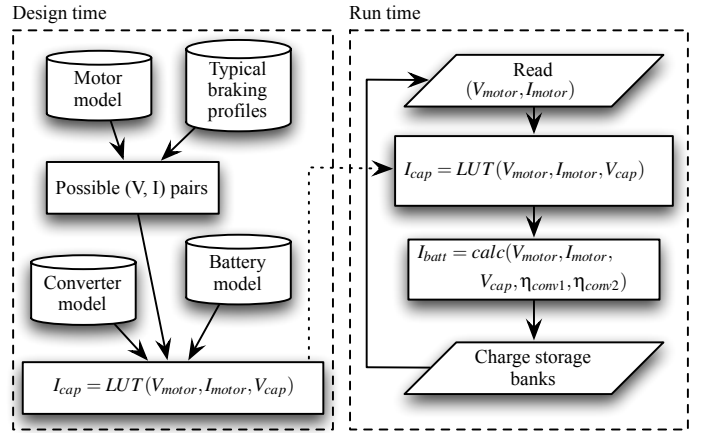


Fig. 3. The proposed MPTT of regenerative braking [20].

is determined by the cycle efficiency of the storage elements and the power conversion efficiency of the power conversion circuits. In addition, it is important to maximize the actual power delivered from the power source to the hybrid [18].

We introduce a maximum power transfer tracking (MPTT) method for regenerative braking with a battery-supercapacitor hybrid storage. The power generation of the traction motors changes according to the braking conditions. The output current of the traction motor is a direct function of its torque, and the output voltage is affected by the rotational velocity (RPM) which varies significantly during the braking period. The MPTT mandates to keep up with the optimal charging setup of the supercapacitor and battery banks during the entire braking process.

Figure 3 shows the design framework of the proposed MPTT for regenerative braking. We use a torque and RPM profile of a regenerative braking, and the initial SoC of the battery and supercapacitor banks, as the input of the design. The output is the charging current profile of battery and supercapacitor banks over time during the braking period. We devise a lookup table-based algorithm to perform MPTT. Figure 3 shows the framework. We compute the power conversion efficiency at design time and build a lookup table with the input of V_{motor} , I_{motor} and V_{cap} . We do not include SoC of batteries due to the size of the lookup table without loss of generality, because battery energy capacity is much bigger than that of supercapacitors. We determine the optimal I_{cap} by the use of the lookup table. Besides, we calculate I_{batt} fast enough at run time because we have analytical model of the converter efficiency.

C. Experimental Results

We show that the proposed MPTT method outperforms the conventional techniques that maintain a constant battery current [15] or the supercapacitor first policy [17]. We use regenerative brake profile from [20] shown in Figure 4, which is the case for an electric vehicle slowing down from 70 km/h to 0 km/h.

We have implemented the experimental framework in MATLAB environment. The battery bank terminal voltage is 375 V and it contains 53 kWh energy when it is fully charged.

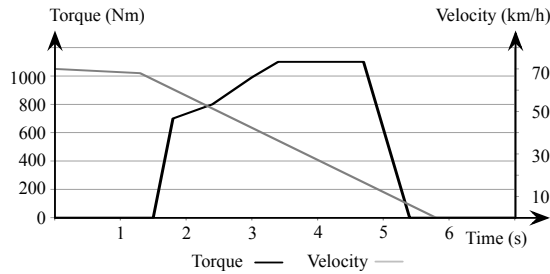


Fig. 4. Braking profile of an electric vehicle from 70 km/h to 0 km/h [20].

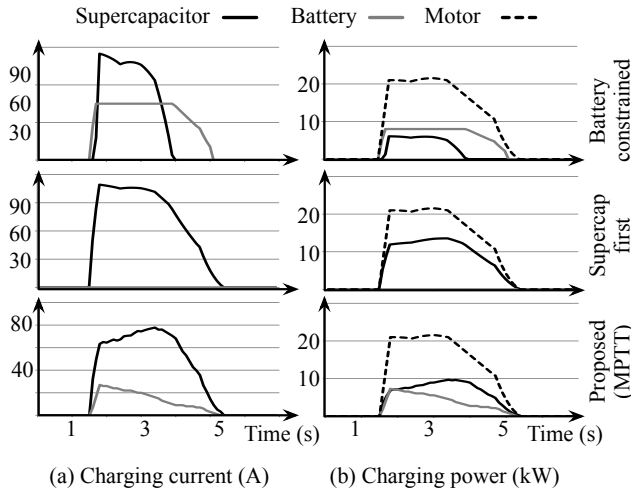


Fig. 5. Charging current and power for each components.

The values closely resemble that of an electric vehicle from Tesla. For the supercapacitor bank, we use Maxwell 7 F supercapacitor bank which is the same as [21] and stack them to store comparable terminal voltage to the battery bank. We borrow the traction motor model parameters such as back electromotive force (EMF) coefficients and torque coefficients from [22].

The resulting profiles are shown in Figure 5. The top two graphs show the charging profiles for the first baseline, where battery charging current is constant and rest of the generated power is handled by the supercapacitor bank. The sum of battery and supercapacitor charging power is far below the motor power generation. The main reason for the efficiency degradation is the rate-capacity effect loss from the battery bank. The next two graphs show results for supercapacitor first policy. They show much more efficient curve. However, due to excessive output current of the supercapacitor bank charger and increased terminal voltage, there is slight degradation in the power conversion efficiency. The last two graphs show result for the proposed MPTT regenerative braking. The proposed technique mitigates conversion loss from the supercapacitor charger by sharing the burden with the battery bank. The overall energy efficiency values for three cases are 59.6%, 62.6% and 68.2%, respectively, which is equivalent to 14.4% and 8.9% more energy generation compared with the

constant-battery-current policy and supercapacitor-first policy.

VI. CONCLUDING REMARKS

In this paper we have discussed a number of systems and software related challenges in the context of electric vehicle design. We have seen that not only new computational structures need to be designed, but they also have to be supported by appropriate communication networks and protocols. Finally, we have outlined a number of issues related to hybrid storage systems and presented some preliminary results.

REFERENCES

- [1] J. Scott-Thomas, "Under the Hood: Silicon in Autos Driving Patent Plans," *EE Times*, 2007. [Online]. Available: <http://www.eetimes.com/design/automotive-design/4004755/Under-the-Hood-Silicon-in-autos-driving-patent-plans>
- [2] AUTOSAR, "AUTomotive Open System ARchitecture," <http://www.autosar.org>.
- [3] D. Hoch, W. Huhn, U. Nher, and A. Zielke, "The race to master automotive embedded systems development," *McKinsey Brochures*, 2006.
- [4] C. Buckl, "The Software Car: Information and Communications Technology (ICT) as an Engine for the Electromobility of the Future," *eCar ICT System Architecture for Electromobility*, 2011. [Online]. Available: <http://www.fortiss.org/research/projects/electric-mobility-ict-architecture-2030.html>
- [5] "Siemens VDO Making a Case for In-Wheel Systems: the eCorner Project," *Green Car Congress*, Sep 2006. [Online]. Available: http://www.greencarcongress.com/2006/09/siemens_vdo_mak.html
- [6] N. Navet, A. Monot, B. Bavoux, and F. Simonot-Lion, "Multi-source and Multicore Automotive ECUs-OS Protection Mechanisms and Scheduling," in *2010 IEEE International Symposium on Industrial Electronics (ISIE)*, 2010, pp. 3734–3741.
- [7] FlexRay Consortium, "FlexRay Communications Systems - Protocol Specification," <http://www.flexray.com>.
- [8] R. Daoud, H. Amer, H. Elsayed, and Y. Sallez, "Ethernet-based Car Control Network," in *Proc. of CCECE '06*, 2006, pp. 1031–1034.
- [9] CAN, "Controller Area Network," <http://www.can.bosch.com/>.
- [10] P. Milbredt, B. Vermeulen, G. Tabanoglu, and M. Lukasiwycz, "Switched FlexRay - Increasing the Effective Bandwidth and Safety of FlexRay Networks," in *Proc. of EFTA '10*, Bilbao, Spain, 2010.
- [11] D. Jansen and H. Buttner, "Real-time Ethernet: the EtherCAT solution," *Computing and Control Engineering*, vol. 15, p. 16, 2004.
- [12] *FlexRay Communications System, Electrical Physical Layer Specification, Version 2.1*, FlexRay Consortium, May 2005. [Online]. Available: <http://www.flexray.com>
- [13] MOST, "Media Oriented Systems Transport," <http://www.mostcooperation.com/>.
- [14] H. Zeng, M. Di Natale, A. Ghosal, and A. Sangiovanni-Vincentelli, "Schedule Optimization of Time-Triggered Systems Communicating Over the FlexRay Static Segment," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 1–17, 2011.
- [15] S. Lukic, S. Wirasingha, F. Rodriguez, J. Cao, and A. Emadi, "Power management of an ultracapacitor/battery hybrid energy storage system in an hev," in *IEEE VPPC*, 2006.
- [16] M. Ortuzar, J. Moreno, and J. Dixon, "Ultracapacitor-based auxiliary energy system for an electric vehicle: Implementation and evaluation," *IEEE T. on Industrial Electronics*, 2007.
- [17] R. Carter and A. Cruden, "Strategies for control of a battery/supercapacitor system in an electric vehicle," in *SPEEDAM*, 2008.
- [18] Y. Kim, N. Chang, Y. Wang, and M. Pedram, "Maximum power transfer tracking for a photovoltaic-supercapacitor energy system," in *IEEE/ACM ISLPED*, 2010.
- [19] M. Pedram, N. Chang, Y. Kim, and Y. Wang, "Hybrid electrical energy storage systems," in *IEEE/ACM ISLPED*, 2010.
- [20] M. Park, S. Kim, L. Yang, and K. Kim, "Development of the control logic of electronically controlled hydraulic brake system for hybrid vehicle," in *SAE*, 2009.
- [21] J. Dixon, M. Ortuzar, and E. Wiechmann, "Regenerative braking for an electric vehicle using ultracapacitors and a buck-boost converter," in *EVS*, 2000.
- [22] J. S. Hsu, C. W. Ayers, C. L. Coomer, and R. H. Wiles, "Report on Toyota/Prius motor torque capability, torque property, no-load back EMF, and mechanical losses," Oak Ridge National Laboratory, Tech. Rep., 2004.