Cyber-Physical Systems Design for Electric Vehicles

Martin Lukasiewycz[1], Sebastian Steinhorst[1], Florian Sagstetter[1]
Wanli Chang[1], Peter Waszecki[1], Matthias Kauer[1], Samarjit Chakraborty[2]

[1] TUM CREATE, Singapore, Email: martin.lukasiewycz@tum-create.edu.sg
[2] TU Munich, Germany, Email: samarjit@tum.de

***Abstract*—Electric vehicles are emerging as a solution to environmental changes and transportation challenges in growing mega-cities. Compared to combustion engine vehicles, electric vehicles bring along new challenges in the Cyber-physical Systems (CPSs) design. This paper gives an overview of several of these challenges and presents initial and potential solutions for the design of the electric powertrain and Electrical and Electronic (E/E) architectures for electric vehicles. The powertrain consists of multiple complex CPSs such as the battery, the electric motor, and a distributed energy management system. These components require a complex monitoring and control in order to guarantee safety and maintain a high efficiency. For this purpose, novel E/E architectures become necessary that facilitate a predictable distributed computation and communication, requiring a paradigm shift towards fully time-triggered systems. These E/E architectures will also enable novel CPSs such as innovative driver assistance systems, x-by-wire control to further increase the safety and energy-efficiency of electric vehicles, and a pervasive interaction of the vehicle and the grid. Instead of focusing on the specific applications, this paper describes the prerequisite architectural changes that are necessary to implement these novel functions.**

I. Introduction

A Cyber-physical System (CPS) is characterized by a tight interaction between the computational devices and the physical environment. Modern vehicles comprise a multitude of CPSs such as the Electronic Stability Program (ESP), Anti-lock Braking System (ABS), and Powertrain Control Module (PCM). These control systems were introduced to improve the safety and increase the efficiency of vehicles significantly. Electric vehicles bring along new challenges in CPS design by introducing new components like the battery, the electric motor, as well as the energy management for the entire electric powertrain. For safety and efficiency reasons, these spatially distributed components require a sophisticated monitoring and control. For this purpose, a paradigm change for Electrical and Electronic (E/E) architectures becomes necessary towards an entirely time-triggered scheme to enable a deterministic behavior of components and the system. At the same time, it is necessary to design the CPS taking the computational components and the physical environment concurrently into account. An example of two different designs of a control system is depicted in Figure 1. This example illustrates the effects of the software function design as well as the computation and communication devices on the behavior of the system: Delays or jitter in the communication and computation or inaccurate controller parameters may lead to a significantly worse performance of the system.

This publication is made possible by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

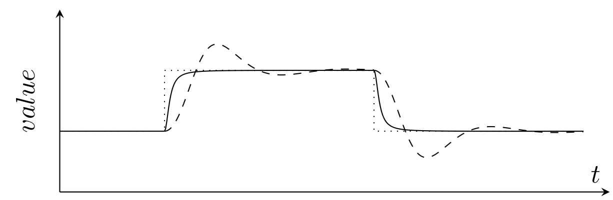

Fig. 1. Illustration of a control function of two different Cyber-physical System (CPS) implementations. The control quality depends on many factors, including the software function design as well as the computation and communication devices and their interaction. The optimal value is illustrated by the dotted line (⋯⋯). Here, the system illustrated by a solid line (——) has a better control quality than the system illustrated by the dashed line (– – –).

Contributions of the paper. In this paper, an overview of CPS design challenges for electric vehicles is presented and potential solutions are discussed. A concurrent design of E/E architectures and the electric powertrain tailored to electric vehicles becomes necessary. For this purpose, E/E architectures will require novel bus systems, computation devices, and operating systems. A paradigm shift towards a time-triggered scheme and towards integrated architectures based on parallel computation devices like multi-core systems that reduce the number of Electronic Control Units (ECUs) is required. These systems ensure a predictable behavior and are therefore particularly advantageous for safety and mission-critical CPSs. On the other hand, the optimal design of the architecture and scheduling of these complex systems is a challenging task which requires a simultaneous optimization of different aspects and objectives. For this purpose, a multi-layered scheme is proposed that copes with the complexity by separating the optimization in different steps.

Compared to combustion engine vehicles, novel functions will become necessary in electric vehicles like x-by-wire control, novel driver assistance functions, a pervasive interaction between the vehicle and the grid, and in particular the electric powertrain. In electric vehicles, the powertrain comprises components like the battery, the electric motor, converters, and inverters. The monitoring and optimal control of the electric powertrain and its components requires a sophisticated approach. Here, both safety and energy-efficiency have to be taken into account when designing these complex CPSs. Consequently, in-depth domain knowledge becomes necessary about the specific components and the E/E architecture. In this paper, an overview of the electric powertrain and its components as well as the interaction with the computational devices is given.

Organization of the paper. The remainder of the paper is organized as follows: Section II introduces components and design methods for E/E architectures that are tailored to electric vehicles and the present CPS design challenges in this domain. Section III presents an overview of components of the electric powertrain as well as control challenges for the entire system.

II. E/E Architecture

In the following, novel communication and computation devices and related design challenges for electric vehicles are presented. The discussed components are based on a time-triggered scheduling to simplify the design of CPSs by ensuring a predictable behavior. This is particularly important for electric vehicles where distributed control of the powertrain, x-by-wire control, etc. require a deterministic computation and communication to increase the safety and efficiency. For this purpose, a design methodology is introduced for time-triggered systems in the automotive domain where still mainly event-triggered systems are predominant.

A. Communication

Protocols and Bus Systems. To overcome the drawbacks of the predominant automotive bus system Controller Area Network (CAN) [1], a paradigm shift from event-triggered to time-triggered systems becomes necessary. A consortium of car manufacturers and suppliers developed the FlexRay protocol [2] which has become a widely accepted standard in the automotive industry. It offers a time-triggered communication in the static segment, a hybrid topology layout, and a high bandwidth of up to 10 Mbit/s per channel. With these characteristics, the FlexRay protocol is perfectly suited to safety-critical CPSs like x-by-wire applications [3] which are highly beneficial in electric vehicles to increase the energy-efficiency. FlexRay has been introduced in various top-of-the-range series vehicles in the last years, see [4].

At the same time, Ethernet is emerging as a bus system for safety-critical systems in the automotive domain due to its twisted-unshielded pair implementation by BROADCOM [5] which significantly reduces the cost and weight of cabling. It allows a data-rate of 100 Mbit/s and higher while the implementation of the Precision Time Protocol (PTP) [6] enables a synchronization of ECUs. To facilitate real-time communication with different priority-levels, approaches like Audio Video Bridging (AVB) [7] might be applied. Several car manufacturers and suppliers including BMW, BOSCH, and FREESCALE have founded a consortium to define recommendations for implementing Ethernet for automotive applications [8], paving the way for Ethernet in cars.

Synchronous Scheduling. With a rising number of hard real-time applications in automobiles and particularly electric vehicles, predictable time-triggered data transmission gains significant importance in the design of CPSs. In contrast, the design of event-triggered systems or asynchronous scheduling requires real-time constraints to be validated with tools like SymTA/S [9] and tested thoroughly which is a time consuming task and also requires expensive oversampling for safety-critical systems. On the other hand, time-triggered systems promise a minimal delay and no jitter. For this purpose, all tasks and messages need to synchronize their time of execution in compliance with the bus schedule. An example of an entirely time-triggered system is given in Figure 2 where all tasks and messages are executed periodically by a predefined schedule such that the end-to-end delay is constant and no jitter occurs. For this purpose, a synchronization of all network participants becomes necessary such that all of them are aware of a common global time. Protocols like FlexRay or Ethernet PTP provide such a synchronization within the network.

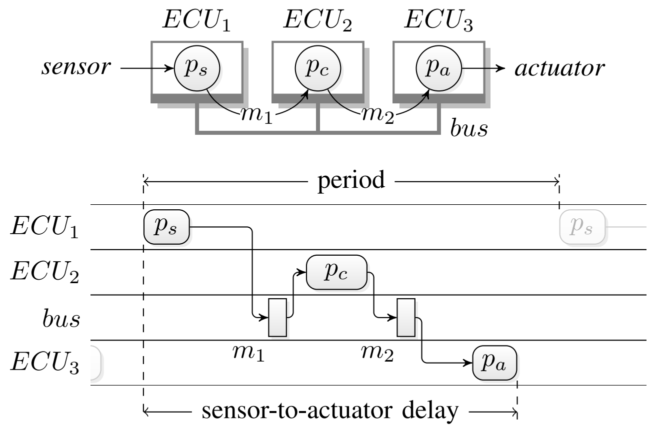

Fig. 2. Illustration of a bus topology, a distributed control function, and the corresponding time-triggered scheduling of tasks and messages.

The design of a global schedule may become a computationally expensive task. State-of-the-art approaches [10] are based on Integer Linear Programming (ILP) and are flexible to support different and heterogeneous bus systems. For the sake of scalability and in order to enable the optimization of different design objectives, both a modular approach and heuristics might be potential solutions to cope with the growing complexity.

FlexRay Switch. To further increase the reliability and effective data rate of FlexRay networks, a switch architecture was proposed [11] and implemented as prototype using a Field Programmable Gate Array (FPGA). The switch is a device that separates the FlexRay network into multiple branches transparently, i. e., complying with the protocol specification such that standard communication controllers can be used. In contrast to a gateway, the switch forwards the messages directly to the respective branches and therefore does not introduce any additional delays. The switch increases the effective data rate as illustrated in Figure 3 and increases the safety by isolating faulty branches. Thus, the switch is an ideal device for the implementation of distributed safety and mission-critical CPSs. In order to exploit the potential of the switch, a scheduling methodology is introduced in [12]. For this purpose, a heuristic three-step approach is used that determines the branches of the network, performs a local scheduling for each node, and finally assembles the local schedules into a global schedule.

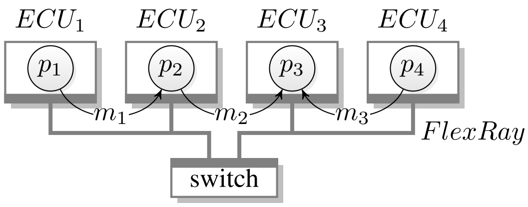

Fig. 3. Illustration of the FlexRay switch that enables the concurrent scheduling of message m_1 and m_3, respectively, by isolating the branches when necessary. In contrast, message m_2 is routed between the two branches.

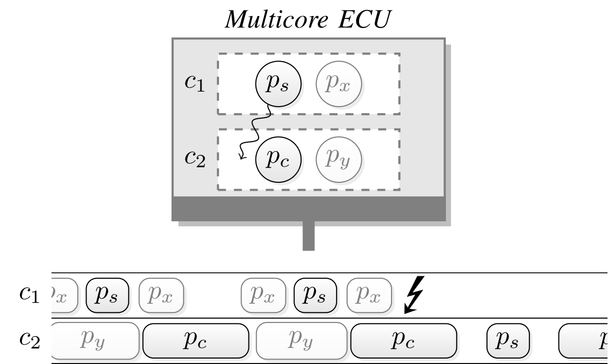

Fig. 4. Illustration of a multicore ECU with two cores. In case of a failure of one core, a degradation is applied. The safety-critical tasks (p_s, p_c) are executed on the working core while the low-priority tasks (p_x, p_y) are not executed anymore.

B. Computation

Automotive Multi-core Systems. Today, a major part of innovation introduced in modern cars happens at the level of E/E architectures and the implementation of complex CPSs using distributed computational devices. This is even more evident for electric vehicles, where x-by-wire applications replace many conventional mechanical and hydraulic control systems and batteries require reliable high-performance management systems with a long life-time due to the continued operation while the vehicle is not in use. However, due to power and heat constraints, the gain in performance of new E/E systems will not result from increasing clock frequencies, but from adding new cores to the ECUs [13].

First of all, the transition from single-core to multi-core brings benefits on the component level of the E/E architectures. By improving the performance per watt ratio and enabling higher levels of parallelism, multi-core ECUs simplify the compliance with safety requirements as defined by ISO 26262 [14]. An even more fundamental change due to the introduction of multi-core ECUs arises for the entire E/E architectures. Here, the current *federated systems* with one function per ECU will yield *integrated systems* with many applications consolidated on one multi-core ECU. The main advantages of the integrated systems are, besides the reduction of wiring costs and mounting efforts, the gain in flexibility and reliability through the decrease of the overall system complexity [15].

As a consequence of this paradigm shift, more and more semiconductor vendors offer multi-core ECUs for specific automotive applications [16]. For example, the *Qorivva* processor familiy from FREESCALE provides a selection of 32-bit Microcontroller Units (MCUs) for powertrain and gateway applications with two and three cores based on the *e200 PowerPC* architecture [17]. Correspondingly, the chassis and safety MCU from STMICROELECTRONICS uses two *e200z4* cores [18] and INFINEON even announced automotive controllers with up to five cores for 2014 [19].

Multicore Operating Systems. To obtain the maximum utilization of multiple execution cores, it is inevitable for applications to run upon a specialized multi-core Operating System (OS), which should ideally provide comprehensive software partitioning mechanisms and ensure efficient sharing of hardware resources. This implies, among other things, the ability to segregate trusted and non-trusted code, split event- and time-triggered execution tasks and in particular support intelligent cache utilization strategies.

Analyzing existing multi-core OS implementations, like *PharOS* [20] or *Corey* [21], can highlight general design trends and, in turn, help both finding appropriate hardware platforms and developing a flexible multi-core OS for E/E architectures of electric vehicles. For example, when dedicating the OS and the applications to different cores, using heterogeneous cache sizes for these cores can bring performance benefits as the working set size of the OS is typically larger than that of the applications [22]. Using explicit messages rather than shared caches for inter-core communication enables the OS to deploy various network optimizations, like pipelining or batching. Furthermore, scheduling can be improved by placing tasks with reference to communication patterns and network effects, as shown for the prototype multi-core OS *Barrelfish* [23].

Design Tool-chain. Regarding electric vehicles, domain specific CPS applications like x-by-wire or battery management systems demand for a multi-core OS with strict requirements on reliability, life-time, and performance. To provide both high-performance and fault-tolerance, parallelization, software isolation, and degradation mechanisms are needed. Furthermore, a fully deterministic real-time behavior can be guaranteed by time-triggered execution models. An example of a time-triggered multi-core scheduling that also supports degradation in case of a failure is illustrated in Figure 4. In this context, however, the hierarchical cache structure of multi-core platforms is an important factor. The system can consist of several core-dedicated private caches as well as higher level shared caches and thus requires sophisticated coherence algorithms to provide consistency and minimize cache-misses. Consequently, an uncertainty for the runtime memory behavior arises.

To cope with the multi-core cache problem, different approaches have been proposed. For example, in [24], the authors introduce a cache-aware *Pfair*-based scheduling scheme to reduce miss rates in shared caches. Whereas [25] describes how a multi-core OS implementation can use hardware Performance Monitoring Units (PMUs) to determine cache partition sizes or regions with bad cache usage.

In our opinion, however, it is not enough to regard this issue in isolation from the scheduling and the OS perspective, respectively. To guarantee the hard real-time behavior of multi-core ECUs, it is necessary to design a compiler which considers possible scheduling strategies provided by the OS. This scheduling-aware compiler ideally comprises a deep knowledge of the underlying hardware, including the caches, and supports a light-weight and time-triggered multi-core OS tailored to automotive control systems. Hence, the resulting fully predictable code execution not only enables Worst Case Execution Time (WCET) calculations but provides a reliable basis for the design of entire safety-critical E/E architectures.

C. Design Space Exploration

Design Flow. The concurrent optimization of the E/E architecture and scheduling for CPSs in electric vehicles is a challenging Design Space Exploration (DSE) task due to the high number of distributed applications and their complexity. While the architectural optimization comprises the determination of the resource allocation and configuration, task mapping, and message routing, the scheduling has to be performed such that all applications fulfill their end-to-end latency constraints. For time-triggered systems, the scheduling is defined by process task offsets and appropriate bus schedules which is a complex design task.

The DSE task is illustrated as Y-chart approach [26] in Figure 5 and is formulated as the following multi-objective optimization problem:

Definition 1 (Design Space Exploration)**.**
optimize $f(x)$
subject to:
x is a feasible *architecture*
it exists a feasible *schedule* y for *architecture* x

In real-world problems, the objective function f consists of multiple functions (sometimes also including non-linear calculations). In general, the principal design objectives depend on the architecture x while the scheduling y has to be performed such that all deadline constraints are fulfilled.

Multi-layer Optimization. A methodology for the architectural DSE is proposed in [27]. The problem of resource allocation and configuration, task mapping, and message routing is mapped to a set of linear constraints with binary variables such that a heuristic search as presented in [28] might be applied. This heuristic approach allows the efficient and effective optimization of multiple conflicting objectives for large problems. In [10], a scheduling framework for time-triggered systems is proposed. It is capable of handling different preemptive and non-preemptive operating systems and different bus systems including FlexRay. To combine both methods to the required DSE, a sophisticated concurrent optimization becomes necessary. Here, we propose to use an iterative approach that delivers candidate architectures using the method in [27]. For each architecture, a scheduling is performed following the method in [10]. In case the scheduling is feasible, the optimization is continued. If the architecture does not permit a feasible schedule, a conflict refinement is performed to determine and exclude the architecture decision that prevents a feasible schedule. First results show that this approach is capable of optimizing large problems efficiently.

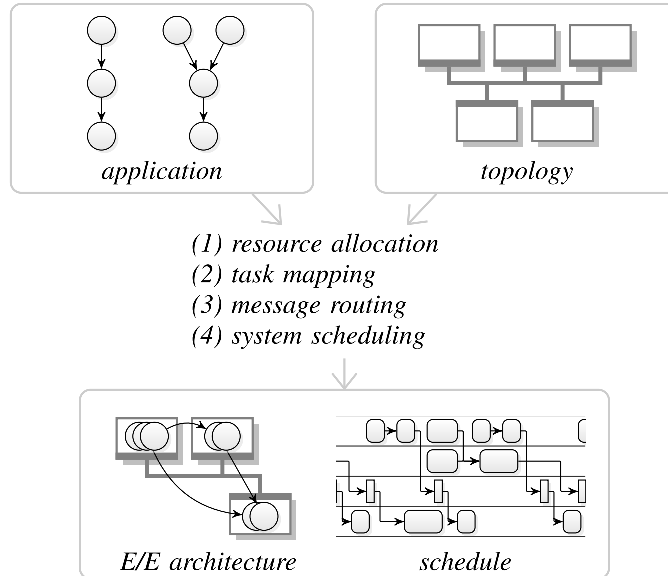

Fig. 5. Y-chart approach that concurrently optimizes the architecture and scheduling. The DSE comprises the resource allocation and configuration, task mapping, message routing, and scheduling.

III. Electric Powertrain

The electric powertrain is the main distinctive feature of an electric vehicle compared to a combustion engine vehicle. The powertrain comprises components like the battery, the electric motor, the converters, and inverters which need to be controlled and monitored by the E/E architecture. In the following, these components and their interactions are presented and discussed.

A. Battery Management

Battery Management System. With the continuous demand for increased energy and power density in batteries for electric vehicle applications, Battery Management Systems (BMSs) have significantly gained importance [29]. The architecture of a common BMS is illustrated in Figure 6. It has a hierarchical structure where the battery management device is communicating with the module management devices via a private bus. With the tight interaction between the cells and the computational devices as well as the spatial distribution of the components, the BMS is a complex CPS that requires a thoughtful design, taking safety issues and energy-efficiency into account.

Lithium-Ion batteries are widely accepted as dominating other battery technologies regarding energy and power density. This type of cell chemistry, however, is very sensitive to its operating parameters where exceeding a specified temperature range as well as deep discharging or overcharging severely damages the cells. Such damage would lead to a reduced lifetime or even safety-critical issues like explosions. Thus, to avoid out-of-specification conditions, the BMS monitors the operating conditions of the cells, determines the State of Charge (SoC), the State of Health (SoH), and controls the discharging and charging of the cells. The required information processed by sophisticated SoC and SoH models is based on each cell's temperature, voltage and, if available, the current flow.

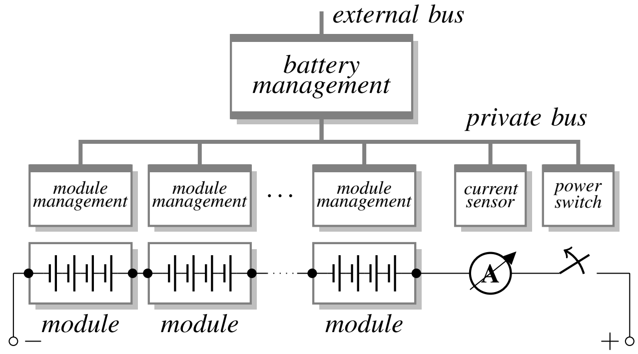

Fig. 6. Illustration of the Battery Management System (BMS) as CPS consisting of the computational and physical devices, see [29].

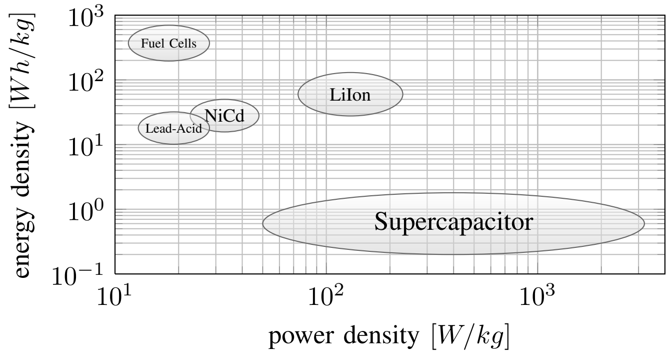

Fig. 7. Ragone plot that illustrates the energy and power densities of different energy storages. Here, Lithium-Ion (LiIon), Nickel-Cadmium (NiCd), and Lead-Acid batteries as well as fuel cells and supercapacitors are illustrated.

Due to small tolerances during manufacturing, each cell has small parameter differences that are exacerbated by aging. Therefore, the BMS performs cell balancing [30] to assure that all cells reach a common SoC. This can be done simply but inefficiently by either discharging individual cells that reached their full charge earlier than others (passive cell balancing) or in a more complex manner but efficiently by routing energy between cells to equalize the SoC of the cells (active cell balancing).

Distributed Battery Management. State-of-the-art BMS implementations currently used in the industry are based on a hierarchical master-slave architecture where all cells communicate with a central ECU. The BMS illustrated in Figure 6 introduces an additional hierarchy with the module-layer. Here, modules are formed within the series-connected cells that are controlled and monitored by the module management units that in turn communicate with a global master which performs the battery management.

This hierarchical architecture has several drawbacks that may be addressed by a novel concept that considers an entirely distributed BMS. In a distributed BMS, each module or cell is an autonomous system that interacts with other entities either via powerline or wireless communication, eliminating wiring and weight. Such an approach might bring along several features, including an increased safety and efficiency. In addition, a more frequent and therefore more precise SoC estimation on the cell level becomes possible for distributed BMS implementations. Approaches like the *Kalman* filter described in [31], [32], [33] that measure the Open Circuit Voltage (OCV) to estimate the SoC could be performed locally without communication restraints and may be combined with *Coulomb*-counting the cell current.

However, a distributed approach also bears significant challenges at system-level. Algorithms have to be designed for self-organization, neighbor discovery, and cell balancing without a global knowledge. First results towards this for portable devices are presented in [34]. The balancing hardware to be used could be along the lines of [35]. Overall, we gain more flexibility concerning the spatial distribution of the battery cells, can calibrate the initial SoC more precisely during charging, and most importantly we have no single point of failure anymore – smart cells might even switch off faulty neighbors and replace them in a hot-spare fashion.

Distributed Control Verification. The verification and validation of battery packs and their management systems is of very high importance due to the hazards posed by the constantly growing energy density. This motivates the investigation of control verification techniques for CPSs.

The classic control system approach assumes a flawless communication in the background. This is reasonable for a single system without any communication delays. It may become too optimistic once sensor, controller and actuator are spatially distributed in the BMS in Figure 6 and delays occur due to the communication overhead.

Many attempts, e.g., [36], have been made to prove control requirements such as stability for distributed systems. Most of them are addressing a specific setup that is facing delay and/or packet loss described in a worst-case fashion. Distributed control verification provides a richer interface for the interaction of control setup and communication setup. Based on [37], [38], a control performance requirement is translated into an ω-regular language that describes a transmission behavior requirement. On the other hand, modeling communication and processing hardware in a suitable fashion will produce an ω-regular language that describes the possible transmission behaviors. With such a modeling, a standard model checker might then be used to verify that a specific communication setup meets the control requirements. Once such a verification framework is in place, an optimization and exploration procedure becomes possible which might be applied to common hierarchical or distributed BMS.

B. Energy Storage Systems

Hybrid Systems. The energy storage technology has been fast evolving in the past decade, including different kinds of batteries and double-layer capacitors (supercapacitors). A plot that illustrates the energy and power density of some common energy storage systems is given in Figure 7. The performance metrics of energy storage systems are manifold including specific energy, specific power, cycle life, cycle efficiency, charging capability, self-discharge rate, etc. [39]. Even for one type of a storage system, different types with different properties exist. While Lithium-Ion batteries are widely accepted as dominating other battery technologies for full electric vehicles, other types of energy storage elements are advantageous in different aspects and thus the advantages of each type of element may be utilized with a proper Hybrid Electrical Energy Storage (HEES) system design while the disadvantages may be masked.

The most common example of HEESs for electric vehicles is a combination of a battery and a range extender which is a combustion engine that generates energy in case the main battery is empty. A gasoline or fuel cell range extender has a significantly higher energy density than any battery but a higher energy cost and lower efficiency. Therefore, the control has to ensure that the range extender is only activated in case the primary energy storage is depleted. In case of large electric vehicles like city buses, a combination of other energy storage systems becomes feasible to handle the high power requirements for instance with double-layer capacitors.

Major challenges of the design of HEESs are the determination of an optimal topology and control strategy. One challenge lies in power electronics offering appropriate connection among different energy storage elements. Relevant topologies are proposed in [40] but if the costs of investing in power electronics components is too high, the advantages of the HEES could be compromised. Another challenge is the control which more specifically includes topics of charge allocation, charge migration and charge replacement. Optimization of charge migration in terms of efficiency is investigated in [41], but generalized methodologies are mandatory for future HEES design.

Heterogeneous Batteries. It is not only possible to form an HEES system with different types of batteries but also with different types of cells in the same kind of battery forming heterogeneous batteries. For instance, battery technologies have varying specific energy and power densities as illustrated in Figure 7 and described in [42]. Therefore, one possibility might be to build up an energy storage system with both use-oriented specific power and energy densities with different types of cells of Lithium-Ion or other cell chemistries.

One of the major challenges in heterogeneous batteries is the design of the BMS. Current BMSs are designed for a single type of cell. Multiple different cells would require a thoughtful design of the control and monitoring. Therefore, for heterogeneous batteries, a highly flexible BMS is required that can cope with the requirements of thermal management and SoC estimation of such heterogeneous topologies.

C. Electric Motor Control

IGBT Inverter and Electric Motor. The electric motor is one of the main components of electric vehicles and the design of the controller has to take reliability, safety, and efficiency into account. For example Permanent Magnet Synchronous Motors (PMSMs) are increasingly applied in different domains including hybrid and electric vehicles. PMSMs are similar to brushless motors and have permanent magnets on the rotor and windings on the stator. These synchronous electric motors have several advantages over non-synchronous motors such as the power efficiency and a high power density. For a smooth motor operation, PMSMs are driven by sinusoidal waveforms that are controlled by Insulated Gate Bipolar Transistors (IGBTs) [43]. IGBTs are three-terminal power semiconductors primarily used as electronic switches using Pulse Width Modulation (PWM). In electric vehicles and hybrid electric vehicles, a package of six IGBTs is usually used to control the synchronous electric motor with the three-phase sinusoidal output as illustrated in Figure 8.

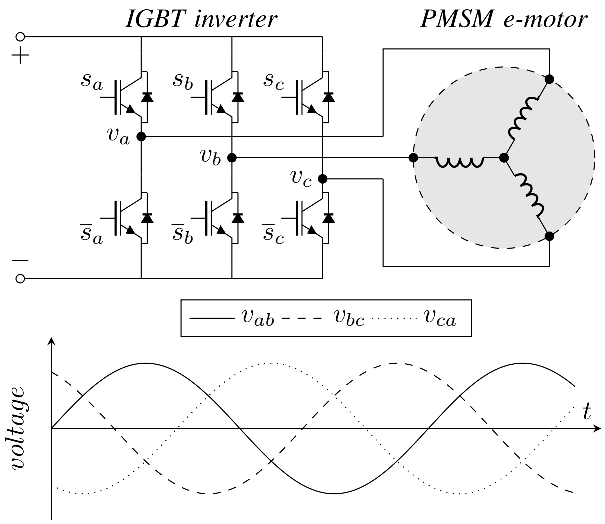

Fig. 8. Illustration of the control of a Permanent Magnet Synchronous Motor (PMSM) using an inverter consisting of six Insulated Gate Bipolar Transistors (IGBTs) $s_a, \overline{s}_a, s_b, \overline{s}_b, s_c, \overline{s}_c$. The IGBTs are controlled such that the three voltages are sinusoidal waveforms that are phase-shifted by $\frac{2}{3}\pi$.

Control signals produced by micro-controllers or relevant circuits are input of the gate terminals of all six IGBTs. The technique implemented is Space Vector Pulse Width Modulation (SVPWM) [44]. For safety reasons, isolation between the IGBT package and the driving board with the microcontrollers becomes necessary [45].

Depending on the motor and its operation and requirements, different alternative setups exist. For instance, a three-level package of 12 IGBTs can replace the current two-level three-phase package of six IGBTs with the potential advantages of higher rail voltage, higher conversion efficiency, and better sinusoidal output with low distortion at the cost of design complexity. In terms of the transistors, Silicon Carbide (SiC) Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) are proposed as alternative to IGBTs with lower turn-on loss and even higher switching speed and lower capacitance. Nevertheless, the general setup remains the same with an inverter that generates the necessary voltages and currents to control the electric motor.

Fault-tolerant Control Strategies. The failure of an IGBT is usually handled by switching off the whole motor which might have severe consequences for instance in the scenario of overtaking. Therefore, it is a challenging problem to handle IGBT faults appropriately. Over-voltage or over-current due to temperature influences could cause short-circuit IGBT faults and then open-circuit failures. The faulty IGBT could also be shut down by the driver board if it detects some signals indicating unsafe behavior leading to open-circuit IGBT faults. With appropriate current and voltage sensors, it is not difficult to detect the open-circuit IGBT but rather challenging to properly handle it. In case of an IGBT failure, the three-phase output will be distorted and thus unable to control the electric motor appropriately [46]. This has a direct influence on the safety and therefore needs to be addressed properly.

One simple way to handle IGBT faults is hardware redundancy with additional IGBTs. However, an obvious drawback is that the cost of the motor control system is increased significantly, which is definitely undesirable especially in the domain of electric vehicles and hybrid electric vehicles where cost reduction is very important. An alternative would be a control strategy adaption in the case of a failure. The regular SVPWM control signals are generated to output smooth sinusoidal waveforms and in case one IGBT fails, a different set of control signals might be generated to control the remaining five IGBTs in such a way that the motor performance is preserved as good as possible. The latter approach deserves more attention because of its low cost nature and the acceptable performance potential. A systematic way of the control signal generation needs to be explored.

D. Energy Flow Control

Powertrain Topology. The topology of the components of the electric powertrain has a direct influence on its efficiency and the required control. Figure 9 illustrates a basic electric powertrain of a full electric vehicle. The main energy source is the battery which supplies the electric motor via the inverter and the 12 V grid via a converter. However, the battery serves also as energy sink while charging or recuperation when the vehicle is braking and the electric motor becomes an energy source. More complex topologies comprise multiple motors, multiple energy storages, and additional external consumers like air conditioning and heaters, as well as additional producers like solar panels. This is in particular the case for larger electric vehicles like electric city buses. In this case, a determination of the topology and control of this complex CPS becomes a challenging task. For the optimization of the topology, also the material and diameter of the cables has to be considered since it has a direct influence on the weight and energy-efficiency. For this purpose, methods from Electronic Design Automation (EDA) like [47] might be applied to determine optimal topologies and control strategies for the electric powertrain.

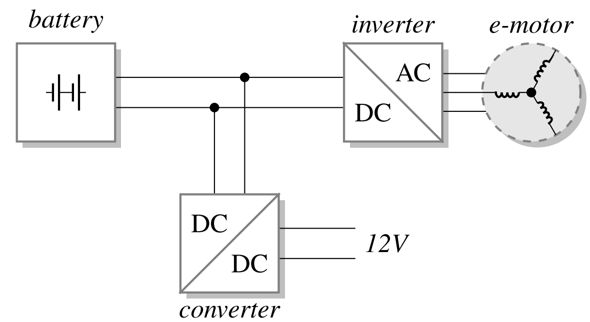

Fig. 9. Illustration of a basic electric powertrain of a full electric vehicle.

Real-time Control. With several energy sources that have to be considered in different electric vehicle powertrain implementations and energy storage topologies [48], a sophisticated real-time control approach for energy flow management is mandatory in the system design [49]. These energy sources can comprise a battery, a combustion engine as range extender, a supercapacitor, and solar panels. Some of the energy consumers can operate as source and vice versa, depending on the energy flow management. While air conditioning or heating are solely energy consumers, the electric motor can recuperate energy during braking that has to be routed to the battery or the supercapacitor. For a powertrain hardware topology with an optimized layout of DC-DC converters for maximum energy efficiency, a real-time control algorithm has to manage the routing of the energy in the system.

The capability of the energy flow control to sense and quickly react to all possible driving conditions is a complex task that moreover has to be strictly verified for error-free operation. An error in the routing where, instead of recuperation of breaking-energy from the motor, the system routes energy to the motor due to a wrong driving state detection or wrong timing assumptions might have fatal consequences. Therefore, a formal control verification has to be conducted that proves that all possible driving conditions are covered by the control approach. Furthermore, a fault-tolerant control has to cope with situations where malfunctions exist without sacrificing safety.

Vehicle-to-Grid. Modern power grid infrastructures face a serious challenge to deliver the appropriate amount of energy into the grid for a highly volatile demand. With the advent of electric vehicles, a new approach to handle demand response and peak leveling is enabled by Vehicle-to-Grid (V2G) approaches [50] constituting a complex CPS. An electric vehicle that is connected to the grid and controllable by a V2G management can perform peak load leveling and serve as a spinning reserve. Hence, either the energy flow to the vehicle is managed to consider times of low demand, or in peak load situations of the grid, the energy storage system of the vehicle can feed power back into the grid.

A large number of electric vehicles connected to a V2G system can significantly contribute to overcome load-related issues in modern power grids and contribute to a more efficient power generation by load leveling. From an economical perspective, a pricing model has to consider the benefits of the energy providers and the inconvenience for the car owner such as battery degradation due to more cycles. It therefore becomes necessary to implement a cooperative framework that enables an efficient V2G management.

IV. Concluding Remarks

In this paper, we highlighted several design challenges and trends for the CPS design for electric vehicles and discussed initial and potential solutions. Compared to combustion engine vehicles, the main component that changes for electric vehicles is the powertrain comprising components like the battery, electric motor, converters, and inverters. These components require sophisticated monitoring and control since they form a complex distributed CPS. For this purpose, novel E/E architectures become necessary, which focus on predictability and verifiability of the designed system. It is mandatory that the design methodologies for CPSs in electric vehicles should be highly flexible since leaps in technology can be expected that lead to alternating design problems. Hence, sophisticated EDA methods from different domains need to be applied to cope with the complexity of the design problems.

References

[1] CAN, "Controller Area Network," http://www.can.bosch.com/.

[2] FlexRay Consortium, "FlexRay Communications Systems - Protocol Specification," http://www.flexray.com.

[3] TTTech Computertechnik AG, "Advanced Electro-Mechanical Braking System: Delphis Brake-by-Wire Concept (Brochure)," 2006.

[4] J. Berwanger, M. Peteratzinger, and A. Schedl, "FlexRay startet durch - FlexRay-Bordnetz fuer Fahrdynamik und Fahrerassistenzsysteme (in German)." *Elektronik automotive: Sonderausgabe 7er BMW*, 2008.

[5] Broadcom, "BroadR-Reach Ethernet Hardware," 2011. [Online]. Available: http://www.broadcom.com

[6] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," 2008.

[7] H. Lim, L. Volker, and D. Herrscher, "Challenges in a Future IP/Ethernet-based In-car Network for Real-time Applications," in *Proc. of DAC 2011*, 2011, pp. 7–12.

[8] "OPEN Alliance Special Interest Group," 2011. [Online]. Available: http://www.opensig.org/

[9] Symtavision, "SymTA/S." [Online]. Available: http://www.symtavision.com/symtas.html

[10] M. Lukasiewycz, R. Schneider, D. Goswami, and S. Chakraborty, "Modular Scheduling of Distributed Heterogeneous Time-triggered Automotive Systems," in *Proc. of ASP-DAC 2012*, 2012, pp. 665–670.

[11] P. Milbredt, B. Vermeulen, G. Tabanoglu, and M. Lukasiewycz, "Switched FlexRay - Increasing the Effective Bandwidth and Safety of FlexRay Networks," in *Proc. of EFTA '10*, Bilbao, Spain, 2010.

[12] M. Lukasiewycz, S. Chakraborty, and P. Milbredt, "FlexRay Switch Scheduling A Networking Concept for Electric Vehicles," in *Proc. of DATE 2011*, 2011, pp. 76–81.

[13] J. Wolf, M. Gerdes, F. Kluge, S. Uhrig, J. Mische, S. Metzlaff, C. Rochange, H. Casse, P. Sainrat, and T. Ungerer, "RTOS Support for Parallel Execution of Hard Real-Time Applications on the MERASA Multi-core Processor," in *Proc. of ISORC 2010*, May 2010, pp. 193–201.

[14] N. Navet, A. Monot, B. Bavoux, and F. Simonot-Lion, "Multi-source and Multicore Automotive ECUs-OS Protection Mechanisms and Scheduling," in *Proc. of ISIE 2010*, 2010, pp. 3734–3741.

[15] P. Peti, R. Obermaisser, F. Tagliabo, a. Marino, and S. Cerchio, "An Integrated Architecture for Future Car Generations," in *Proc. of ISORC 2005*, 2005, pp. 2–13.

[16] S. Schliecker, J. Rox, M. Negrean, K. Richter, M. Jersak, and R. Ernst, "System level performance analysis for real-time automotive multicore and network architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 7, pp. 979–992, 2009.

[17] F. Semiconductor, "Automotive Solutions - Setting the Space for Innovation," 2010, http://www.freescale.com/.

[18] STMicroelectronics, "32-bit Power Architecture Microcontroller for Automotive," 2011, http://www.st.com/.

[19] H. Eul, "IFX Day 2010," 2010, http://www.infineon.com/.

[20] C. Aussaguès, D. Chabrol, V. David, D. Roux, N. Willey, A. Tournadre, and M. Graniou, "PharOS, a Multicore OS Ready for Safety-related Automotive Systems: Results and Future Prospects," *Proc. ERTS 2010*, 2010.

[21] S. Boyd-Wickizer, H. Chen, R. Chen, Y. Mao, F. Kaashoek, R. Morris, A. Pesterev, L. Stein, M. Wu, Y. Dai *et al.*, "Corey: An Operating System for Many Cores," in *Proc. of OSDI 2008*, 2008, pp. 43–57.

[22] D. Wentzlaff and A. Agarwal, "Factored Operating Systems (FOS): The Case for a Scalable Operating System for Multicores," *Artificial Intelligence*, vol. 43, no. 2, pp. 76–85, 2009.

[23] A. Baumann, P. Barham, P. Dagand, T. Harris, R. Isaacs, S. Peter, T. Roscoe, A. Schüpbach, and A. Singhania, "The Multikernel: a New OS Architecture for Scalable Multicore Systems," in *Proc. of SOSP 2009*, 2009, pp. 29–44.

[24] J. Anderson, J. Calandrino, and U. Devi, "Real-time Scheduling on Multicore Platforms," in *Proc. of RTAS 2006*, 2006, pp. 179–190.

[25] R. Azimi, D. Tam, L. Soares, and M. Stumm, "Enhancing Operating System Support for Multicore Processors by Using Hardware Performance Monitoring," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 2, pp. 56–65, 2009.

[26] B. Kienhuis, E. Deprettere, P. Van Der Wolf, and K. Vissers, "A Methodology to Design Programmable Embedded Systems," in *Embedded Processor Design Challenges*. Springer, 2002, pp. 321–324.

[27] M. Lukasiewycz, M. Streubühr, M. Glaß, C. Haubelt, and J. Teich, "Combined System Synthesis and Communication Architecture Exploration for MPSoCs," in *Proc. of DATE 2009*, 2009, pp. 472–477.

[28] M. Lukasiewycz, M. Glaß, C. Haubelt, and J. Teich, "SAT-Decoding in Evolutionary Algorithms for Discrete Constrained Optimization Problems," in *Proc. of CEC 2007)*, 2007, pp. 935–942.

[29] M. Brandl, H. Gall, M. Wenger, V. Lorentz, M. Giegerich, F. Baronti, G. Fantechi, L. Fanucci, R. Roncella, R. Saletti, S. Saponara, A. Thaler, M. Cifrain, and W. Prochazka, "Batteries and Battery Management Systems for Electric Vehicles," in *Proc. of DATE 2012*, 2012, pp. 971 –976.

[30] J. Cao, N. Schofield, and A. Emadi, "Battery Balancing Methods: A Comprehensive Review," in *Proc. of VPPC 2008*, 2008, pp. 1 –6.

[31] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 1. Background," *Journal of Power Sources*, vol. 161, no. 2, pp. 252–261, 2004.

[32] ——, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2. Modeling and identification," *Journal of Power Sources*, vol. 161, no. 2, pp. 262–276, 2004.

[33] ——, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 3. State and parameter estimation," *Journal of Power Sources*, vol. 161, no. 2, pp. 277–292, 2004.

[34] S. Mandal, P. Bhjowani, S. Mohanty, and R. Mohapatra, "IntellBatt: Towards Smarter Battery Design," in *Proc. of DAC 2008*, 2008, pp. 872–877.

[35] T. Kim, W. Wiao, and L. Qu, "Series-connected Self-reconfigurable Multicell Battery," in *Proc. of APEC 2011*, 2011, pp. 1382–1387.

[36] G. Walsh, Y. Hong, and L. Bushnell, "Stability Analysis of Networked Control Systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.

[37] G. Weiss and R. Alur, "Automata based interfaces for control and scheduling," *Hybrid Systems: Computation and Control*, vol. 4416, pp. 601–613, 2007.

[38] R. Weiss, G.and Alur, "Regular Specifications of Resource Requirements for Embedded Control Software," in *Proc. of RTAS 2008*, 2008, pp. 159–168.

[39] N. Chang, Y. Kim, and Y. Wang, "Hybrid Electrical Energy Storage Systems," in *Proc. of ISLPED 2010*, 2010, pp. 363–368.

[40] A. Khaligh and Z. Li, "Battery, Ultracapacitor, Fuel Cell, and Hybrid Energy Storage Systems for Electric, Hybrid Electric, Fuel Cell, and Plug-In Hybrid Electric Vehicles: State of the Art," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 2806–2814, 2010.

[41] Y. Wang, Y. Kim, Q. Xie, N. Chang, and M. Pedram, "Charge Migration Efficiency Optimization in Hybrid Electrical Energy Storage (HEES) Systems," in *Proc. of ISPLED 2011*, 2011, pp. 103–108.

[42] D. Linden and T. Reddy, *Handbook of Batteries*. McGraw-Hill, 2002.

[43] V. Khanna, *The insulated Gate Bipolar Transistor: IGBT Theory and Design*. Wiley-IEEE Press, 2003.

[44] C. Govindaraju and K. Baskaran, "Efficient Sequential Switching Hybrid-Modulation Techniques for Cascaded Multilevel Inverters," *IEEE Transactions on Power Electronics*, vol. 26, pp. 1639–1648, 2011.

[45] A. Maamoun and M. Ahmed, "Microprocessor Control System for PWM IGBT Inverter Feeding Three-Phase Induction Motor," in *Proc. of IECON 1995*, 1995, pp. 1354–1359.

[46] K. Rothenhagen and F. Fuchs, "Performance of Diagnosis Methods for IGBT Open Circuit Faults in Three Phase Voltage Source Inverters for AC Variable Speed Drives," in *Proc. of ECPEA 2005*, 2005, pp. 1354–1359.

[47] M. Lukasiewycz, M. Glaß, C. Haubelt, J. Teich, R. Regler, and B. Lang, "Concurrent Topology and Routing Optimization in Automotive Network Integration," in *Proc. of DAC 2008*, 2008, pp. 626–629.

[48] T. Hofman and R. van Druten, "Energy Analysis of Hybrid Vehicle Powertrains," in *Proc. of VPP 2004*, 2004.

[49] H. Yoo, S.-K. Sul, Y. Park, and J. Jeong, "System Integration and Power-Flow Management for a Series Hybrid Electric Vehicle Using Supercapacitors and Batteries," *IEEE Transactions on Industry Applications*, vol. 44, no. 1, pp. 108 –114, 2008.

[50] E. Sortomme and M. El-Sharkawi, "Optimal Scheduling of Vehicle-to-Grid Energy and Ancillary Services," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 351 –359, 2012.