Automotive Cyber–Physical Systems: A Tutorial Introduction

Samarjit Chakraborty
Technical University of Munich

Mohammad Abdullah Al Faruque
University of California, Irvine

Wanli Chang
TUM CREATE

Dip Goswami
Eindhoven University of Technology

Marilyn Wolf
Georgia Institute of Technology

Qi Zhu
University of California, Riverside

Editor's notes:
This tutorial gives an introduction to novices in CPS and particularly highlights the basics of control theory with respect to automotive applications. The authors furthermore describe the "semantic gap" between control models and their implementation and conclude that a new CPS-oriented design approach is required.
—Jörg Henkel, Karlsruhe Institute of Technology

■ **Today, most of** the innovation in the automotive domain is in electronics and software. All new features in modern cars—like advanced driver assistance systems—are based on electronics and software rather than on mechanical engineering innovations. A modern high-end car has over 100 million lines of code [1] and it is widely believed that this number will continue to grow in the near future. Such code implements different control applications spanning across various functionalities—from safety-critical functions, to driver-assistance and comfort-related ones. These applications run on a distributed electronics and electrical (E/E) architecture, consisting of often hundreds of programmable electronic control units (ECUs) that communicate via different types of communication buses like CAN [2], FlexRay [3], LIN [4], and more recently also automotive Ethernet [5].

Current hardware/software development workflows for the automotive domain are highly compartmentalized. Here, control algorithms are designed by development teams who often have no influence or even little or no view into the E/E architecture on which the software implementing these controllers will eventually run. As a result, the design of the control algorithms is based on a number of idealistic assumptions like the control inputs can be computed instantaneously, there is no delay between sensing and the use of the sensor data to compute the control input, or between the time the control input is computed and when it is available for actuation. Further, the availability of finite precision arithmetic and the side effects introduced when compiling controller models into code are also not systematically addressed. While these issues are anyway difficult to address because of the lack of well-established techniques for handling them in the control theory literature, their effects become very pronounced in the automotive domain because of the highly distributed and heterogeneous nature of automotive E/E architectures.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/MDAT.2016.2573598
Date of publication: 26 May 2016; date of current version: 17 June 2016.

2168-2356/16 © 2016 IEEE
Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

In other words, because many of the assumptions made at the model level do not hold when the designed controllers are implemented, there is a large performance gap between the controller models and their implementations. In extreme cases, even the stability of a controller—which was verified at the model level—might not hold after the code synthesized from the model is implemented. This semantic gap between control models and their implementation necessitates a significant amount of ex post facto testing, debugging and redesign, often using ad hoc techniques. This not only makes certification difficult, but also leads to resource over-dimensioning thereby increasing costs.

In addition to the importance of certification or guaranteeing correctness because of the safety-critical nature of many automotive functionalities, the automotive domain is also highly cost sensitive. Hence, resource-efficient implementation of automotive control software is an important issue. However, traditional techniques for controller design have only focused on ensuring stability and satisfying control performance metrics such as settling time or peak overshoot. While efficient implementation of algorithms is one of the cornerstones of computer science, when it comes to control algorithms, there are very few standard techniques for designing efficient implementations of these algorithms.

CPS-oriented design

The essence of the cyber–physical system (CPS) paradigm is the integrated design of control algorithms and the computational platforms on which these algorithms would run. While the distributed and heterogeneous nature of automotive E/E architectures makes them a perfect candidate for CPS-oriented design, it is only recently that some progress has been made in this direction [6], [7].

Techniques accounting for platform architecture characteristics at the control design stage, as well as resource-efficient design of control algorithms are starting to appear in the scientific literature [8]–[10]. There is now also an improved awareness of how design decisions for different subsystems and components in an automotive hardware/software architecture influence each other. This understanding has led to a number of codesign proposals. For example, traditionally HVAC (heating, ventilation, and air conditioning) systems in cars have only focused on passenger comfort and energy usage. However, for electric vehicles (EVs) the HVAC control can have a major influence on battery lifetime and capacity fading. Given the current costs of batteries and the issues with driving ranges of EVs, accounting for battery characteristics when designing automotive HVAC systems has been gaining traction recently. Similarly, as ECUs are becoming more powerful, they are increasingly being exposed to the side effects of semiconductor technology scaling like manufacturing variability, soft errors, and aging. This is especially critical in the automotive domain because such ECUs, unlike processors in consumer electronics, are exposed to extreme temperatures and electromagnetic interferences. By appropriately designing control algorithms (where the software implementing these algorithms would run on the ECUs) some of these reliability and aging effects can be mitigated.

Finally, security for the automotive domain [11], [12] is a challenging problem because of the resource-constrained and cost-sensitive nature of automotive E/E architectures. Here, investigating the impact of different light-weight security mechanisms on higher layer control algorithms and applications seems to be meaningful. Toward this, mechanisms for evaluating the tradeoffs between security and control performance are now starting to emerge.

Organization

This tutorial gives an overview of some of these problems, recent advancements in addressing them, and the challenges in moving forward. It is intended for beginning researchers and practitioners to enable further exploration and is not an exhaustive survey of this topic. For example, important issues in advanced driver assistance systems and autonomous driving—such as the interaction between video processing, AI, and control theory have not been discussed. The focus has instead been on highlighting the interplay between embedded systems and software design and control theory for the automotive domain.

While we have attempted to give the reader some insights into the mathematical details of the topics discussed here, wherever possible, we have

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

tried to keep the mathematics to a minimal and focused on the main intuitions.

The first section of this article describes basic control theory, which is required to appreciate what follows next. The next section, entitled Resource-aware automotive control software design, discusses how control/architecture codesign techniques may be used to systematically implement control algorithms on automotive E/E architectures, taking into account different resource constrains. The following section continues this discussion but focuses on platform reliability issues and battery usage in the case of electric vehicles. The section Automotive climate control continues with electric vehicles and outlines the benefits of coordinating the HVAC system in an electric car with its motor control, before discussing automotive CPS from the perspective of security.

Feedback control systems

In this section, we briefly describe the basics of feedback control applications that will be considered in the later sections. A control application regulates the behavior of a dynamic system [13]. Most of them are modeled by a set of linear differential equations

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

where $x(t) \in R^n$ is the system state, $y(t)$ is the system output, and $u(t)$ is the control input applied to the system. When implemented on embedded platforms, the system states are periodically measured by the sensors at discrete-time instances $t_k = kh,\ k = 0, 1, 2, 3, \ldots$. The interval $(t_{k+1} - t_k)$ is the sampling period h. The sampled system states are $x[k] = x(t_k)$. Similarly, the sampled system output is $y[k] = y(t_k)$. The control input is updated only at the discrete-time instances t_k and is held constant over the sampling interval h using a zero-order hold (ZOH) circuit. Thus

$$u(t) = u[k], t_k \leq t < t_{k+1}. \tag{2}$$

The above ZOH implementation can be modeled by solving (1), resulting in

$$\begin{aligned} x[k+1] &= A_d x[k] + B_d u[k] \\ y[k] &= Cx[k] \end{aligned} \tag{3}$$

where

$$A_d = e^{Ah}, B_d = \int_0^h (e^{A\tau} d\tau) B. \tag{4}$$

Clearly, A_d and B_d depend on the sampling period h.

Quality of control (QoC). Quality/performance of a control application is often quantified with respect to user requirements, for example, speed of response and comfort. Settling time is a widely used metric to quantify QoC. The time it takes for the system output $y[k]$ to reach and stay in a closed region around the reference value y_{ref} (e.g., $0.98y_{\text{ref}}$ to $1.02y_{\text{ref}}$) is the settling time and is denoted as t_s. Shorter settling time implies better QoC. In many safety-critical automotive control loops, there is a maximum settling time requirement t_s^0 that must be satisfied for functional correctness.

Controller. A controller aims to design $u[k]$ such that the QoC requirements are met. The general structure of a linear state-feedback controller is as follows:

$$u[k] = Kx[k] + Fr \tag{5}$$

where K is the feedback gain and F is the feedforward gain. A control algorithm computes the gains K and F.

Controller design. The discrete-time dynamics in (3) with control input (5) is called the closed-loop system dynamics

$$x[k+1] = (A_d + B_d K)x[k] + B_d Fr. \tag{6}$$

Stability of a plant and control system depends on the eigenvalues of $(A_d + B_d K)$, which are referred to as the system poles and are denoted by p_i for $1 \leq i \leq n$. A stable system requires all poles $|p_i| < 1$. Usually, poles closer to zero provide a faster response and require a higher value of the input signal $u[k]$. Once the feedback gain K is designed, the static feedforward gain F is obtained to achieve $y[k] \to y_{\text{ref}}$, and is given by

$$F = 1/(C_d(\mathbf{I} - A_d - B_d \times K)^{-1} B_d) \tag{7}$$

where $\mathbf{I}$ is an n-dimensional identity matrix. Clearly, the QoC of a control loop depends on the poles p_i

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

of $(A_d + B_d K)$. The poles can be placed at desired locations p_i for QoC optimization by pole-placement methodologies (by designing the feedback gain K [14]). The controller design question boils down to the choice of desired poles p_i such that QoC requirements are met and/or optimized.

Physical constraints. In almost every real-world system, due to the physical constraints of the actuator, there is some maximum available control input and the controller needs to be designed such that the maximum value of $|u[k]|$ does not exceed this limit $U_{\max}$.

Resource-aware automotive control software design

As described earlier, automotive E/E platforms are highly resource constrained as well as cost sensitive. In this section, through examples, we outline how computation-, memory-, and communication-aware control applications may be designed. Note that these are the three primary resource types that are targeted when designing efficient algorithms, but in the particular case of control algorithms, the techniques necessary deviate significantly compared to conventional control algorithm design. For the case of computation-aware design, some of the mathematical details have been outlined, especially to highlight the differences with conventional controller design as described in the earlier section Feedback control systems. For the remaining two cases (memory- and communication-aware design), only the high-level design strategy has been described.

Computation-aware control systems design

OSEK/VDX-compliant operating systems (OSs), with preemptive fixed-priority scheduling, are widely used in the automotive domain [15], [16]. With such an OS once each application gets released, it is allowed to access the processor periodically. There are various predefined periods of release times, and each application is assigned one. Different applications may have different periods. Every time an application is released, its program gets the chance to be executed, depending on its priority.

Here, a time table containing all the periodic release times within the alleged hyperperiod (i.e., the minimum common multiple of all periods) of the applications needs to be configured. An example with a set of three periods 2, 5, and 10 ms is illustrated in Table 1. The hyperperiod is equal to 10 ms, and the time table repeats itself every 10 ms by resetting the timer.

Generally for a feedback control application, a shorter sampling period allows the controller to respond to its plant more frequently, and is thus potentially able to achieve better QoC. The obvious downside is a higher processor load, since the control program is executed more frequently. Let us assume that the set of available periods restricted by an OSEK/VDX OS is ϕ. Denoting e_i to be the worst case execution time (WCET) of a control application C_i, if conventional controller design and a uniform sampling period of h is used, the processor load for C_i is

$$L_i = \frac{e_i}{h}. \tag{8}$$

The upper bound on the load of any processor is 1. Considering a single processor p

$$\sum_{\{i|C_i \text{ runs on } p\}} L_i \leq 1. \tag{9}$$

Clearly, increasing the sampling period of a control application decreases its processor load, and thus potentially enables more applications to be integrated on the ECU, thereby resulting in a more cost-effective system.

Since an OSEK/VDX OS only offers a limited set of predefined sampling periods to the control applications at hand and often the optimal sampling period for a given control application is not directly realizable, the conventional way is to use the largest sampling period offered by the OS that is

Table 1 An example OSEK/VDX OS time table of applications release.

Time	Release
0*ms*	Applications with periods of 2*ms*/5*ms*/10*ms*
2*ms*	Applications with the period of 2*ms*
4*ms*	Applications with the period of 2*ms*
5*ms*	Applications with the period of 5*ms*
6*ms*	Applications with the period of 2*ms*
8*ms*	Applications with the period of 2*ms*
10*ms*	**Repeat actions** at 0*ms*

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

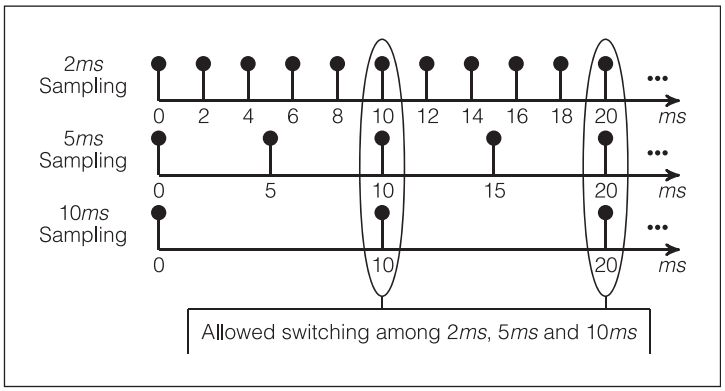

Figure 1. Allowed switching instants among multiple sampling periods.

smaller than the optimal one. This clearly wastes computation resources.

A computation-aware controller, on the other hand, can switch between multiple available sampling periods offered by the OSEK/VDX OS, thereby achieving the desired QoC and reducing processor load simultaneously [17]. However, the controller design in such cases has to take into account this switching between sampling periods and is different from the design outlined in the section Feedback control systems. Possible switchings with sampling periods of 2, 5, and 10 ms on OSEK/VDX OS are illustrated in Figure 1. For one application, switching between two sampling periods can only occur at the common multiple of them. For instance, switching between 2 and 5 ms is possible at the instants of 10, 20 ms, and so on. Therefore, possible sequences of sampling periods are $\{2\text{ ms}, 2\text{ ms}, 2\text{ ms}, 2\text{ ms}, 2\text{ ms},\ 5\text{ ms},\ 5\text{ ms},\ \textit{repeat}\}$, $\{5\text{ ms}, 5\text{ ms}, 10\text{ ms}, \textit{repeat}\}$, and so on.

The schedule $\{5\text{ ms}, 5\text{ ms}, 10\text{ ms}, \textit{repeat}\}$ achieves an average sampling period of 6.67 ms, which reduces the processor load compared to the schedule with a constant sampling period of 5 ms and thus saves computation resources. The challenge is to design a performance-oriented controller that exploits this nonuniform sampling but satisfies the requirements on QoC.

We assume that the cyclic sequence of sampling periods for a control application defines a schedule S

$$S = \{T_1, T_2, T_3, \ldots, T_N\} \tag{10}$$

where $\forall j \in \{1, 2, \ldots, N\}$, $T_j \in \phi$. Dictated by the schedule S, this boils down to N systems (combination of plant and controller) switching cyclically in a deterministic fashion. The dynamics of these N systems within one cycle of S is given by

$$\begin{aligned} x[k+1] &= A_d(T_1)x[k] + B_d(T_1)u[k-1] \\ x[k+2] &= A_d(T_2)x[k+1] + B_d(T_2)u[k] \\ &\vdots \\ x[k+N] &= A_d(T_N)x[k+N-1] + B_d(T_N)u[k+N_1]. \end{aligned} \tag{11}$$

In order to avoid varying sensor-to-actuator delays, the actuation occurs at the end of a sampling period and the sensor-to-actuator delay is equal to one sampling period. Therefore, $x[k+1]$ depends on its previous state $x[k]$ and the control input $u[k-1]$, which is computed according to $x[k-1]$ and applied at t_k. We now introduce a new state $z[k] = [\,x[k] \quad u[k-1]\,]^T$. Then, $\forall j \in \{1, 2, \ldots, N\}$

$$\begin{aligned} z[k+j] = &\begin{bmatrix} A_d(T_j) & B_d(T_j) \\ \mathbf{0} & 0 \end{bmatrix} z[k+j-1] \\ &+ [\mathbf{0}\ 1]^T u[k+j-1] \end{aligned} \tag{12}$$

where $\mathbf{0}$ is a zero vector. The system output is

$$y[k+j-1] = C_{\text{aug}} z[k+j-1] \tag{13}$$

where

$$C_{\text{aug}} = [C\ 0]. \tag{14}$$

The control input is designed as

$$u[k+j-1] = K_j z[k+j-1] + F_j r. \tag{15}$$

Denoting A_{aug} and B_{aug} as

$$\begin{aligned} A_{\text{aug}}(T_j) &= \begin{bmatrix} A_d(T_j) & B_d(T_j) \\ \mathbf{0} & 0 \end{bmatrix} \\ B_{\text{aug}}(T_j) &= [\mathbf{0}\ 1]^T. \end{aligned} \tag{16}$$

The closed-loop dynamics is

$$\begin{aligned} z[k+j] &= A_{\text{aug}}(T_j)z[k+j-1] + B_{\text{aug}}(T_j)u[k] \\ &= (A_{\text{aug}}(T_j) + B_{\text{aug}}(T_j)K_j)z[k+j-1] \\ &\quad + B_{\text{aug}}(T_j)F_j r. \end{aligned} \tag{17}$$

We denote the closed-loop system matrix as

$$A_{cl,j} = A_{\text{aug}}(T_j) + B_{\text{aug}}(T_j)K_j. \tag{18}$$

The poles to be placed are the eigenvalues of $A_{cl,j}$ (recall the earlier discussion on pole-placement in the section Feedback control systems). Note

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

that (13), (15)–(18) are applied for every j in $\{1, 2, \ldots, N\}$. We now formulate an optimization problem for the pole placement as

$$\min_{D} t_s, \quad \text{subject to}$$
$$|u[k]| \leq U_{\max}, \quad t_s \leq t_s^0 \tag{19}$$

where the poles are decision variables and the settling time t_s is to be minimized as the objective. There are three constraints. First, the input saturation has to be respected. Second, the settling time requirement has to be satisfied. Third, $\mathbb{D}$ is a domain of poles that ensure the system stability. Such a constrained nonconvex optimization problem with significant nonlinearity can be solved by heuristics like particle swarm optimization (PSO).

We now show some evaluation results of such a computation-aware controller using an electromechanical braking (EMB) system. The settling time requirement t_s^0 is 150 ms. The set of available sampling periods offered by the OSEK/VDX OS in this case is

$$\phi = \{1 \text{ ms}, 2 \text{ ms}, 5 \text{ ms}, 10 \text{ ms}, 20 \text{ ms}, 50 \text{ ms}, 100 \text{ ms}, 200 \text{ ms}, 500 \text{ ms}\}. \tag{20}$$

As shown in Table 2 and Figure 2, the schedule $S1 = \{5 \text{ ms}\}$ cannot meet the settling time requirement. The largest sampling period smaller than 5 ms in ϕ is 2 ms. The schedule $S2 = \{2 \text{ ms}\}$ is able to fulfil all the requirements. Both of these cases rely on conventional controller design as described in the section Feedback control systems. According to (8), assuming that the WCET is 0.7 ms, the processor load of $S2$ is 35%. Clearly, this number can be unnecessarily large, preventing more applications from being packed into the same ECU. We then evaluate the schedule $S0 = \{2 \text{ ms}, 2 \text{ ms}, 2 \text{ ms}, 2 \text{ ms}, 2 \text{ ms}, 5 \text{ ms}, 5 \text{ ms}\}$ switching between 2 and 5 ms, with the resulting controller being designed as described in this section. $S0$ has a slightly longer settling time than $S2$, yet still fulfils the settling time requirement of 150 ms. Extending (8), the processor load is 24.5%, achieving a 30% reduction compared to $S2$, and now allowing more applications to be packed. While this is one among several possible techniques (e.g., see [18]) to achieve computationally efficient controllers, it illustrates the need to design controllers by taking into account the characteristics and constraints of the implementation platform, which also holds for other resource types like memory and communication.

Table 2 Settling time and processor load of three schedules. Bold numbers indicate that the requirement on settling time is satisfied.

Schedule	Settling Time	Load
$S1 = \{5ms\}$	256.40ms	14%
$S2 = \{2ms\}$	**113.27*ms***	35%
$S0 = \{2ms, 2ms, 2ms, 2ms, 2ms, 5ms, 5ms\}$	**132.14*ms***	24.5%

Memory-aware control systems design

Memory and especially on-chip memory on ECUs substantially increases the ECU cost. In many automotive setups, the code for different control applications is stored in a bigger inexpensive flash memory. Before a particular application is executed, its code is fetched from the flash to the on-chip memory located on the processor. The smaller the on-chip memory is, the more cost effective is the ECU. However, the additional latency involved in fetching the code from the flash memory deteriorates QoC. The question is, following a CPS approach, can the control algorithms be designed to mitigate such delays and exploit this memory hierarchy?

Given a collection of control applications (e.g., C_1, C_2, C_3), it is conventional to run the control loops of them in a round-robin fashion (C_1, C_2, C_3, C_1, C_2, C_3, ...). This frequently refreshes the ECU on-chip memory and the time it takes to fetch a code from the flash increases its WCET. In order to address this issue, again a nonuniform sampling scheme—but one that is different from what was used in the case of computation-aware

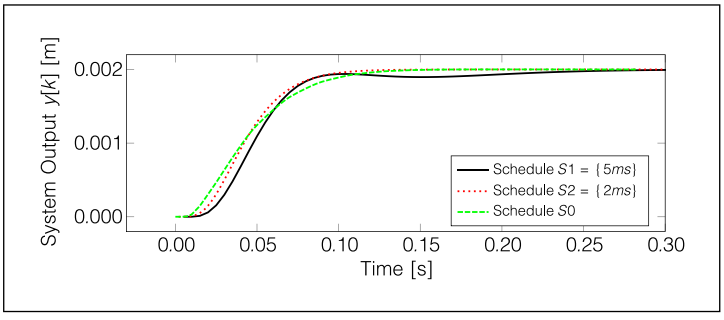

Figure 2. System output of three different schedules.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

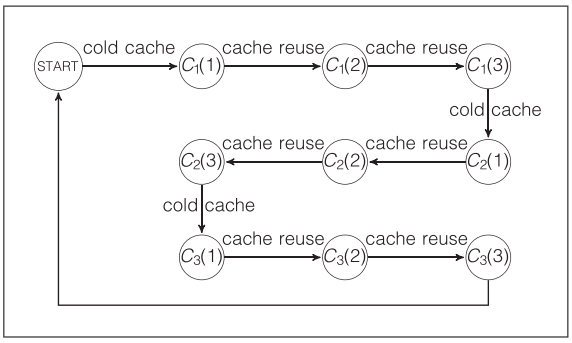

Figure 3. Each application is consecutively executed three times. After the first execution $C_i(1)$, some instructions in the cache can be reused and thus the WCETs of the following two executions are shortened, resulting in improved QoC.

controllers—is helpful. Here, the control loop of each application is consecutively run multiple times—in order to increase the cache or on-chip memory reuse, before moving on to the next application. For example, C_1, C_1, C_1, C_2, C_2, C_2, C_3, C_3, C_3, . . ., can be used.

As shown in Figure 3, $C_i(j)$ denotes the jth execution of the control application C_i. Before the first execution $C_i(1)$, the cache is either empty (i.e., cold cache) or filled with instructions from other applications, that are not used by C_i (equivalent to cold cache). The WCET of $C_i(1)$ can be computed by a number of existing standard techniques [19]–[21]. Before the second execution $C_i(2)$, the instructions in the cache are from the same application C_i and thus can be reused. This results in more cache hits and hence shorter WCET. Depending on which path the program takes, the amount of WCET reduction varies. This requires a technique to compute the guaranteed WCET reduction of $C_i(2)$ and $C_i(3)$, independent of the path taken. After WCET results are computed, the control timing parameters (e.g., sampling periods and sensor-to-actuator delays) can be derived. Now, a controller design technique similar to the one presented in the section on Computation-aware control systems design may be used to exploit the shortened (but nonuniform sampling) periods and achieve better QoC. Note that in contrast to conventional control systems design, where uniform or constant sampling periods are used, the nonuniform sampling requires nonstandard design techniques such as the one outlined in the previous section.

Communication-aware control systems design

As an example of communication-aware automotive control systems design, we will use a setup where two ECUs communicate over a FlexRay bus. Here, a control application is partitioned and mapped onto these ECUs and control signals have to be sent over the FlexRay bus. FlexRay supports both time-triggered (TT) and event-triggered (ET) or priority-based communication schemes. When the characteristics of the communication bus are not considered during the controller design phase, the controller is designed with assumptions on timing parameters like sampling periods and sensor-to-actuator delays, which have to be satisfied by the implementation platform. In FlexRay, the TT communication has deterministic timing behavior and results in a constant message delay, whereas the delay suffered by messages mapped onto the ET segment varies. There is a tradeoff between the number of TT slots used and the QoC [22]. Hence, configuring the FlexRay parameters appropriately—to ensure certain message delay constraints—and mapping all control messages to the TT segment is a straightforward solution. However, TT slots are considered to be more expensive and the question is: Given a set of control applications and their corresponding control signals, can good QoC be achieved by using fewer TT slots compared to when all messages are mapped to TT slots?

In what follows, we describe a scheme that realizes this [23]–[25]. Here, control messages are switched between TT and ET slots. This protocol is illustrated in Figure 4. If a plant is in the steady state, i.e., the norm of the system state vector $\|x\|$ is less than or equal to a predefined threshold E_{th}, ET slots are used for the corresponding control messages. This is because variable control message delays, when the plant is anyway in a steady state, have little negative influence on QoC. However, when a disturbance forces the plant out of the steady state, it is checked whether a free TT slot is available. If a TT slot is available, then the control message is switched to such a slot. This TT slot is then used until the plant returns to the steady state.

Unlike in the previous two cases of computation- and memory-aware controller design, in this case, the switches between TT and ET slots are not predefined and can happen at arbitrary

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

times. This rules out the possibility of designing a controller for a known schedule, as was done in the two previous cases. Even when two controllers are designed—one for TT and the other for ET communication—and both are stable, the overall switched system might become unstable because of the switching. In this case, either common quadratic Lyapunov functions or switched Lyapunov functions need to be used to guarantee the overall stability of the system [26], which makes the controller design nontrivial. A comparison of QoC for pure TT, pure ET and such a hybrid communication scheme is shown in Figure 5. Note that the switched hybrid communication scheme results in a shorter settling time of the system and hence a better performance compared to the case where purely ET-based communication is used. At the same time, for a collection of control applications, fewer TT slots are used compared to the case where all control signals use TT communication.

In addition, the interaction between communication and control theory has attracted a lot of attention in general, and also in the context of in-vehicle communication protocols. One of the issues here is to quantify the tolerable message loss/delay in the case of distributed controller implementations, while still maintaining control quality [27]. Once such tolerable loss patterns are identified, the next issue is to design scheduling policies that respect such bounds or patterns. A related question is to verify whether a given scheduling policy satisfies specified control performance constraints [28].

Battery- and reliability-aware controllers

In the case of EVs, control loops are powered and actuated (using voltage and current signals) by a battery pack. Control algorithms compute the actuation signals which are further realized by a specific discharging current profile drawn from the battery pack. A discharging current profile can be represented by $\{(I_1, t_1), (I_2, t_2), \ldots, (I_n, t_n)\}$ where I_i is the current drawn from the battery at time t_i. The amount of electric charges that can be delivered after a battery is fully charged is called the full charge capacity (FCC). The FCC can be approximated by

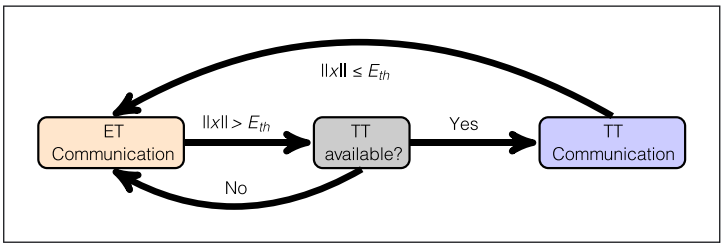

Figure 4. The hybrid communication protocol.

the following equation for a given discharging current profile:

$$Q_{FCC} = \sum_{k=1}^{n} I_k (t_{k+1} - t_k) \tag{21}$$

where the total duration of battery usage is $\sum_{k=1}^{n}(t_{k+1} - t_k)$ and it is divided into n intervals. The current I_k is assumed constant between any interval $(t_{k+1} - t_k)$. According to the battery rate capacity effect, the battery FCC and hence the total duration of battery usage depends on the discharging current profile [29], [30]. It is desired that this duration of battery usage is as high as possible in an EV. A lower I_k over all the intervals increases the battery usage. For any control loop, the resulting discharging current profile depends on the controller gains K and F (as described in the section Feedback control systems). Usually, the controller gains that achieve a higher QoC require a higher current I_k with a worse battery usage. A battery-aware design jointly optimizes the QoC and the battery usage. Toward this, we first model the relationship between them.

We still consider the settling time t_s of a system as a QoC metric. A shorter t_s implies a higher QoC and requires a higher I_k. The battery usage can be quantified by the number of times r the control system can reach a steady state after a disturbance

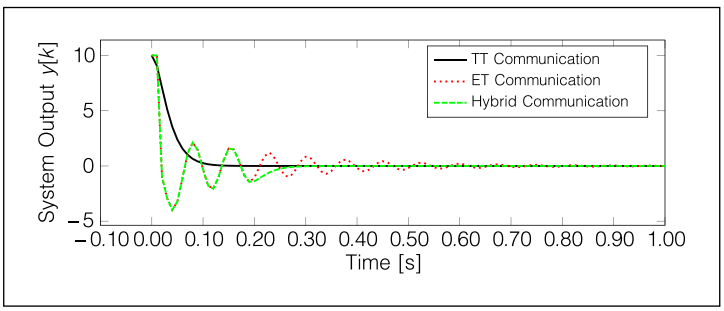

Figure 5. QoC comparison under different communication schemes.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

occurs with a fully charged battery pack. That is, r can be computed by the number of times we can accommodate t_s within the time duration of battery usage. When the battery only powers the control task, the following captures the above relation between battery usage and QoC [31], [32]:

$$r = \frac{a}{\left(\frac{\sum_{k=1}^{n_{sp}} I_k(t_{k+1}-t_k)}{ts}\right)^d \times t_s} \tag{22}$$

where the interval $t_{k+1} - t_k = h$ is the sampling period of the controller. n_{sp} is the total number of sampling periods within t_s, i.e., t_s/h. a and d are experimentally obtained constants. Ultimately, a battery-aware design obtains the control gains K and F that maximize r and minimize t_s with respect to (22).

Co-optimizing QoC and battery usage

As explained in the section Feedback control systems, determining the values of the gains K and F is based on the choice of the desired poles p_i, in order to 1) minimize the settling time t_s, and 2) maximize the battery usage r, based on (22). Usually, optimization techniques maximize or minimize both objectives and therefore we minimize $f_1 = t_s$ and $f_2 = -r$. There can be a number of constraints, e.g., physical constraints, to be satisfied. The design space is continuous with infinite design choices. Furthermore, there is no relationship between objectives and decision variables that can be explicitly formulated. Such optimization problems can be addressed using heuristics and we have explored sequential quadratic programming (SQP) and nondominated sorting genetic algorithms (NSGAs) [33]. We present our results using them later in this section.

Semiconductor aging effects

As mentioned earlier, automotive ECUs are also subject to the usual influences of technology scaling, like semiconductor aging. The question is whether control software can be designed to cope with these influences. As a processor ages, the switching time of its transistors increases, resulting in longer path delays [34]. A common way to deal with such effects is to reduce the processor's operating frequency to ensure that the signal transmission in any path can be completed within one clock cycle [35]. Obviously, the WCET of control tasks gets longer with a lower operating frequency. Consequently, the sampling period of a control loop gets longer, potentially degrading the QoC. A possible approach to aging-aware design is to reoptimize the control gains K and F taking the longer sampling period (due to aging and reduced operating frequency) into account and prevent any QoC degradation [36]. However, this would result in increased control effort being required, which for example in the case of EVs translates into worse battery usage. But through appropriate optimizations, this impact may be reduced.

Illustrative results: Electric motor control

Here we show typical tradeoffs between QoC and battery usage, and also how battery usage slightly deteriorates when a controller has been updated as the underlying processor ages. The governing dynamics of an electric motor can be represented in the form of (1)

$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}-\frac{b}{J} & \frac{K_t}{J}\\ -\frac{K_e}{L} & -RL\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{V}{L}\end{bmatrix}c$$
$$y = [1 \quad 0]\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} \tag{23}$$

where R and L are resistance and inductance in the armature circuit, K_t is the motor torque constant, J is the moment of inertia of the motor, and K_e is the back electromagnetic force constant. A viscous friction model is assumed and the friction torque is proportional to the shaft angular velocity $\dot{\theta}$ by a factor of b. V is the direct current (dc) voltage (motor input) provided by the battery pack. The motor input is adjusted by duty cycle c of the pulse-width-modulation signals. The electric motor speed ranges from 4800 to 7200 RPM. The output target $\dot{\theta}$ is 7200 RPM. The sampling period h of the system is 0.1 s. Parameters of the entire motor control system are the same as shown in [31].

Figure 6 shows the results. A battery-aware design provides a Pareto front between the battery usage and QoC. Each point represents a design with gains K and F. A design point with higher QoC demands more current I_k, resulting in a poor battery usage. Figure 6 also shows that the effect of aging can be mitigated by controller (i.e., K and F) reoptimization. However, the battery usage slightly deteriorates due to the larger control effort that is now required.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

Automotive climate control

In this section, we will continue discussing issues in the context of EVs and illustrate how control strategies in different EV components influence each other and determine the overall system performance. In particular, we discuss how system-level control of multiple EV subsystems may further improve the reliability and energy efficiency of the EV battery.

Since the energy stored in an EV battery is restricted due to the volume and weight limitations, the driving range of EVs is today an important concern. Further, the maximum energy that can be stored in a battery (battery capacity) decreases over time due to charging–discharging cycles. Typically, when the battery capacity reaches 80% of its nominal value from the manufacturing time, the battery has to be replaced. Controllers for different EV devices are responsible for monitoring the device status and controlling the operating variables in order to improve performance and reliability. For instance, the temperature and utilization of the battery cells are monitored and controlled by a battery management system (BMS). Maintaining the battery temperature and adjusting the battery cell utilization improves the battery lifetime and energy efficiency and thereby the EVs driving range [37]–[39].

In addition, there are other influences on battery lifetime and capacity. For instance, a driving route of an EV affects the power consumption of the electric motor [40]–[43], and the HVAC also influences the total power consumption of the EV [44], [45]. These power requests influence the battery utilization and temperature and thereby the battery lifetime and driving range. Therefore, instead of all the subsystems functioning independently, having a system-level controller will help in monitoring and coordinating the influences of these subsystems, thereby improving the overall performance and reliability of the EV.

In order to illustrate such a system-level controller, we will focus on automotive climate control. The HVAC system is monitored and controlled by the automotive climate control in order to provide thermal comfort for the passengers. Multiple variables, e.g., the cabin temperature, ambient temperature, and solar radiation may be monitored and the HVAC system is controlled accordingly to cool/heat the cabin. Automotive climate control utilizes variable-speed fans, valve dampers, and blend doors to control the airflow. The heater and cooler in the HVAC system (evaporator, condenser, compressor, etc.) control the air temperature by exchanging heat. The percentage of total energy consumed by the HVAC system in an EV is much higher than in an internal combustion engine (ICE) vehicle. As shown in Figure 7, the power consumption of major devices in an EV and an ICE vehicle are demonstrated in percentage for different temperatures. Note that the HVAC device can be a major power consumer in the EVs—up to 21% in cold weather. This large amount of HVAC power consumption may decrease the driving range and degrade the battery lifetime further. On the other hand, an HVAC device can be categorized as a flexible load in EVs, since the power consumption can be adjusted without affecting any critical functionality.

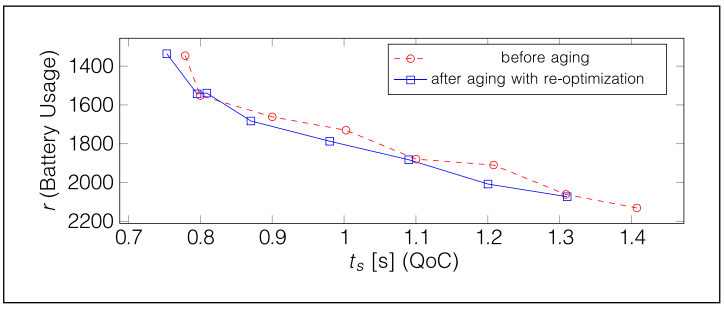

Figure 6. Battery-aware electric motor control and reoptimization taking into account the effect of aging.

In order to implement a battery-lifetime-aware automotive climate control, battery modeling is required as it describes the changes in the battery parameters considering different power requests from the EV, including those from the HVAC. Behavioral modeling of a battery cell can be used to describe its chemical and physical characteristics. The battery lifetime or state of health (SoH) shows how much capacity is left in a battery cell compared to its nominal value when it was manufactured. SoH degrades over time and the degradation rate depends on the stress put on the battery. For instance, if we consider the battery state of charge (SoC) which shows the available charge in the battery, the battery stress can be modeled by the deviation and average of the SoC values over a period of a charge–discharge cycle. The battery SoC changes as the EV is being driven. The SoC change is due to both, the energy

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

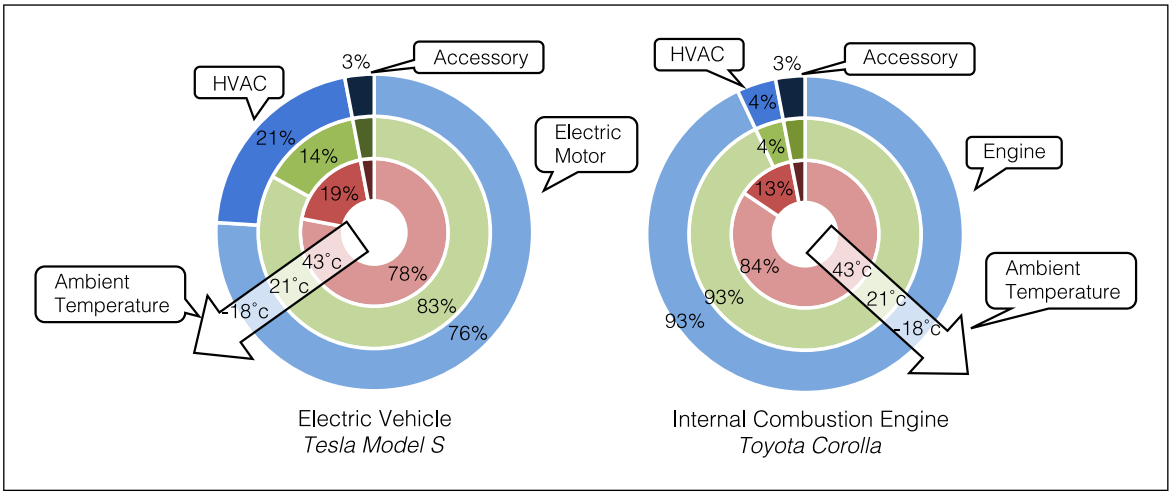

Figure 7. Power consumption of devices in EV and ICE vehicle for different ambient temperatures.

consumption and generation by the EV. Hence, by controlling the EV power request we are able to optimize the average and the deviation of the SoC while driving, thereby controlling how the battery degrades over time.

For system-level automotive climate control, battery characteristics may be leveraged for optimizing both HVAC operation and the battery lifetime. The HVAC operation is adjusted such that the power request from the HVAC device compensates the electric motor's power request. This reduces the stress put on the battery by decreasing the deviation of the battery SoC. Hence, both battery lifetime and driving range get improved. It needs to be noted that adjusting the power request from the HVAC will affect the cabin temperature and may decrease the passenger comfort. However, there should be a limit on the flexibility of the cabin temperature such that it does not compromise passenger comfort significantly.

Such a system-level battery-lifetime-aware climate control requires the modeling and estimation of the electric motor's power request, the HVAC's state and power consumption, and the battery lifetime (see Figure 8). Drive profiles are used to predict route information such as average acceleration (α), average speed (v), and route slope (a). This information is used to estimate the driving forces on the EV, e.g., aerodynamic drag, gravitational force, and rolling resistance. The electric motor generates the propelling force required to cancel out the driving forces. Hence, the power request by the electric motor can be estimated at each time instance. On the other hand, the HVAC power consumption is estimated based on the current state of the cabin (T_z), and the states of the cooling, and heating devices. Then, the HVAC and electric motor power requests are applied to the battery device model in order to estimate the total energy consumption (P), battery SoC, and the battery lifetime or SoH degradation ($\bigtriangledown SoH$). The value of these state variables such as cabin temperature, total power, and SoH degradation are estimated by the automotive climate control over a specific control window $(t \rightarrow t + n \bigtriangleup t)$ in the future. Then, a model-predictive control algorithm optimizes these variables in order to both extend the battery lifetime, and improve the driving range. Finally, the optimized values are applied to the HVAC device at the next time step $(t \rightarrow t + \bigtriangleup t)$.

The benefit of this methodology over currently existing climate controls is that it offers the capability of controlling devices in a system while considering the behavior of other devices that are being influenced. It may reduce the HVAC consumption when the electric motor is consuming a higher power and it would pre-cool the cabin (in case the outside is warmer) before the electric motor consumption gets higher. In this way, the electric motor's consumption would be complemented by the HVAC consumption. Existing climate controls only monitor the HVAC device status and cabin conditions, without evaluating the state

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

of the electric motor or the influence of power consumption of the battery SoH.

The performance of the outlined climate control methodology typically depends on the drive profile of the EV. Figure 9 shows some results for different driving profiles and two different types of HVAC operation—on/off and fuzzy based (where the changes induced by the HAVC system are more gradual). It can be seen that the battery–lifetime-aware climate control performs better in a drive profile with a higher fluctuation. This is due to the fact that the model-predictive control algorithm will compensate the electrical motor power request by adjusting the HVAC power in order to reduce battery stress. Moreover, the performance may vary for different ambient temperatures. Since the HVAC energy consumption changes for different ambient temperatures, the compensation for the electric motor's power request may be different.

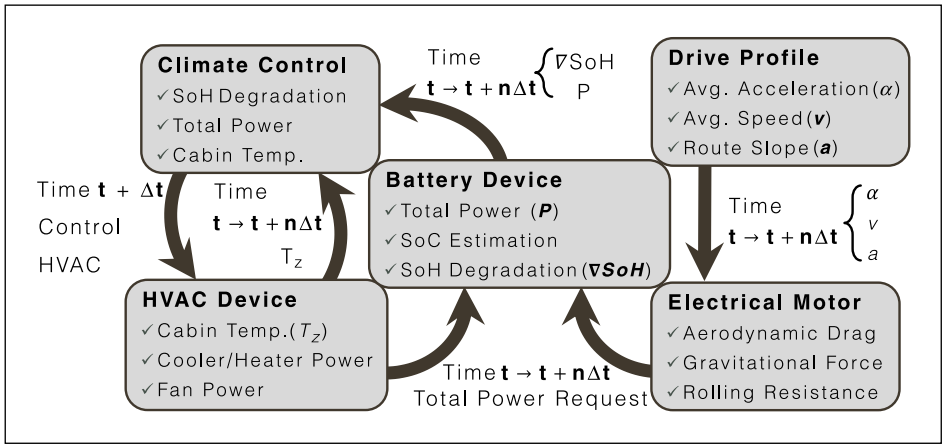

Figure 8. Battery-lifetime-aware automotive climate control methodology.

Cyber–physical automotive security

With increasing vehicle intelligence and connectivity, security and privacy have become pressing concerns for automotive systems. In this section, we will discuss automotive security challenges and the importance of using cyber–physical approaches to address them.

Researchers have shown that modern vehicles can be attacked from a variety of interfaces including physical access such as OBD-II and USB, short-range wireless such as Bluetooth, remote keyless entry, tire pressure sensors and RFID car keys, and long-range wireless channels such as broadcast channels and addressable channels [46], [47]. Furthermore, by compromising a single ECU through some interface, a capable attacker may gain access to other ECUs via internal communication buses such as controller area network (CAN), and attack safety critical subsystems [48], [49]. In [47], Checkoway et al. successfully compromised a production vehicle by hacking into its engine control system, brake control system, and other electronic components. The attacks are conducted through CAN buses using packet sniffing, targeted probing, fuzzing, and reverse engineering.

To address these security attacks, not only the various vehicle interfaces need to be protected, the internal embedded architecture needs to be hardened with security mechanisms as well, such as message authentication mechanisms for CAN [50]–[52] and for time-triggered systems

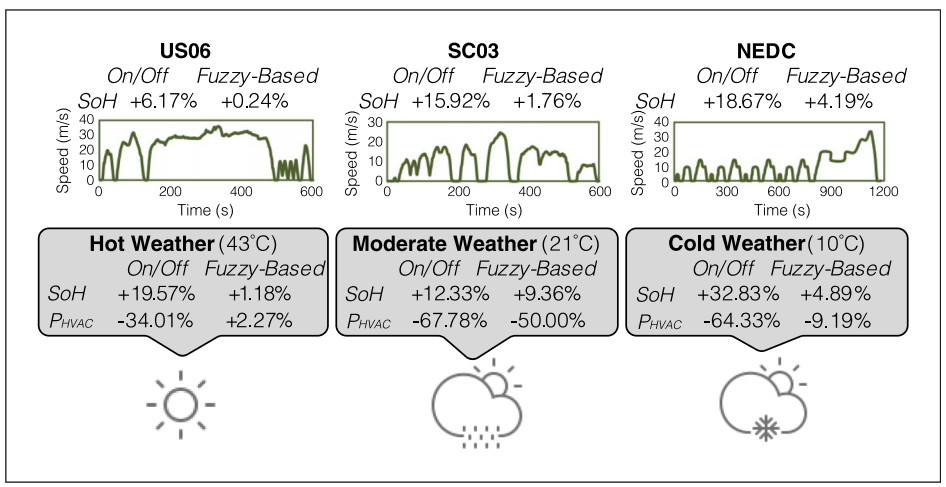

Figure 9. Battery lifetime and HVAC power analysis for different drive profiles and climates.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

[53]–[56]. However, the presence of tight resource constraints, including limited communication bandwidth and computational resources, and strict timing requirements for system safety and performance, makes it difficult or even impossible to add those security mechanisms after the initial design stages, without violating the system constraints or impeding the system performance [57], [58]. In [59], a general security-aware design methodology is proposed to address automotive security from early design stages, with the consideration of stringent timing and resource constraints. The authors propose security-aware mapping algorithms that can map a software task graph onto CAN-based or time-division multiple-access (TDMA)-based platforms, while applying authentication of bus messages through message authentication codes (MACs).

The results in [59] demonstrate the importance of considering security during the design process rather than trying to add security measurement as an afterthought. However, it only considers traditional cybersecurity techniques and will not be able to defend against physical attacks (e.g., those that alter sensor inputs by manipulating the physical environment). The task graph abstraction also loses important functional information that directly affects system security, control performance, and other metrics. To effectively address the automotive security issue, the solution has to start from the initial model capturing system functionality and has to address both cyber and physical aspects.

In [60], a cross-layer design framework is proposed to combine control-theoretic methods at the functional layer and cybersecurity techniques at the embedded platform layer, and addresses security together with other design metrics such as control performance under resource and real-time constraints. The work proposes a general framework for cyber–physical systems and then refines it for automotive systems. It considers multiple control loops sharing an embedded platform, with messages transmitted from sensors to controllers and from controllers to actuators. Each controller (implemented as a control task) collects the sensed information, processes it on an ECU (or ECUs) that might be shared with others, and sends commands to various actuators.

The control-theoretic methods can detect cyber or physical attacks based on the analysis of the system state and dynamics, while the cybersecurity techniques can protect against cyber attacks such as message eavesdropping. However, applying techniques such as message encryption will introduce computation and communication overhead, through the elongation of message transmission time, the additions of decryption tasks, and consequently the elongation of control task execution time due to resource contention. This will in turn have a significant impact on system schedulability and control performance (or QoC).

The proposed framework addresses security, control performance, and schedulability in a holistic fashion. As shown in Figure 10, control performance and system security level are measured at the functional layer, while schedulability is analyzed at the embedded platform layer. To bridge these metrics, a set of interface variables are introduced, specifically the sampling period of every control task and the selection of messages to be encrypted. Intuitively, when the sampling period of a control task increases, its control performance decreases while platform schedulability increases with less frequent activation of the control task. On the other hand, when the number of messages being encrypted increases, the system security level increases while platform schedulability decreases because of the increased overhead. Furthermore, the sampling periods may have to increase for schedulability concern thereby worsening the control performance. These

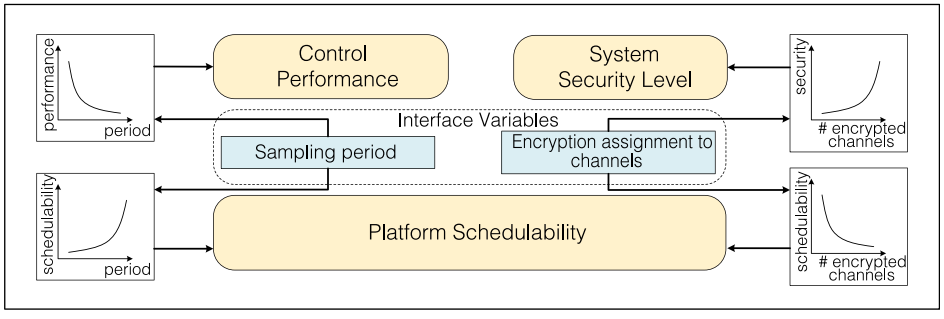

Figure 10. Control and platform codesign for secure cyber–physical systems.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

relations are quantitatively modeled in the codesign formulation in [60].

A simulated annealing algorithm is proposed to explore the selection of messages for encryption and the assignment of task periods based on the codesign formulation, while quantitatively analyzing the tradeoffs between control performance and security level. Figure 11 shows the Pareto front between the two normalized metrics for an automotive case study. We can clearly see the tradeoffs between control performance and security level. During design, constraints on these two metrics may be set according to system requirements. The generated Pareto front will provide a feasible region that is important for making decision choices. For instance, an example feasible region is shown in the figure, under the requirements that the system control performance should be no less than 0.3 and the system security level should be no less than 0.3. Without the codesign approach, it is impossible to identify the feasible designs under such requirements. Instead, the designers might get a solution that violates security requirement if they only optimize for control performance (Figure 11a), or a solution that violates performance requirement if they simply choose to encrypt all messages (Figure 11b). This shows the importance of codesigning security with other design metrics across functional and embedded system layers.

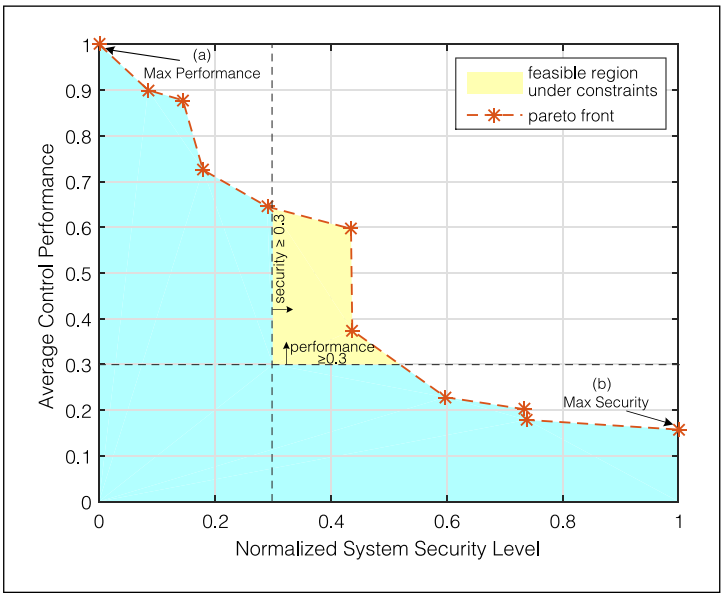

Figure 11. Pareto front between normalized control performance and security level for the industrial example. An example feasible region denotes all feasible solutions under requirement that control performance $\geq$ 0.3 and security level $\geq$ 0.3.

WE HAVE DISCUSSED a number of aspects of automotive cyber–physical systems—starting from reliably and efficiently implementing control algorithms on distributed automotive E/E architectures, to accounting for battery health and semiconductor reliability when designing automotive control algorithms, then system-level control of an HVAC system and the electric motor in an EV, and finally how to implement automotive security while accounting for high-level application performance concerns. In all of these cases, the underlying message was that an integrated design of control algorithms and strategies and the embedded hardware/software architecture leads to improved design, lower costs, and better system reliability. Such approaches not only need innovations in the domain of embedded systems, but also new techniques in control theory. We believe that both these communities will see a surge of new results in the coming years and will continue to work together under the umbrella of cyber–physical systems. The automotive domain, in our opinion—because of the highly distributed nature of automotive E/E architectures, the resource constraints, and the cost pressures—is one of the most interesting application domains for innovating and applying new CPS-oriented design methods. ■

References

[1] R. Charette, "This car runs on code," *IEEE Spectrum*, vol. 46 no. 3, p. 3, 2009.

[2] *Bosch, CAN Specification*, 1991. Version 2.0.

[3] Flexray Consortium, *The FlexRay Communications System Specifications*, 2005. Version 2.1.

[4] *LIN Consortium, LIN Specification Package*, 2010. Revision 2.2A.

[5] H. Kopetz, "The rationale for time-triggered Ethernet," in *RTSS*, 2008.

[6] M. Lukasiewycz et al., "Cyber-physical systems design for electric vehicles," in *DSD*, 2012.

[7] D. Goswami et al., "Challenges in automotive cyber-physical systems design," in *SAMOS*, 2012.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

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

[9] R. Schneider, D. Goswami, S. Zafar, M. Lukasiewycz, and S. Chakraborty, "Constraint-driven synthesis and tool-support for FlexRay-based automotive control systems," in *CODES+ISSS*, 2011.

[10] F. Zhang, K. Szwaykowska, W. Wolf, and V. J. Mooney III, "Task scheduling for control oriented requirements for cyber-physical systems," in *RTSS*, 2008.

[11] P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. Fahmy, and S. Chakraborty, "Lightweight authentication for secure automotive networks," in *DATE*, 2015.

[12] P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Security analysis of automotive architectures using probabilistic model checking," in *DAC*, 2015.

[13] K. Astrom and R. Murray, "*Feedback Systems: An Introduction for Scientists and Engineers*," Princeton, NJ, USA: Princeton Univ. Press, 2009.

[14] J. Ackermann and V. Utkin, "Sliding mode control design based on Ackermann's formula," *IEEE Trans. Autom. Control*, vol. 43, no. 2, pp. 234–237, 1998.

[15] OSEK/VDX Consortium, *OSEK/VDX Operating System Specification*, 2005. Version 2.2.3.

[16] P. Feiler, "Real-Time application development with OSEK: A review of the OSEK standards," Tech. Rep. Carnegie Mellon Univ., 2003.

[17] D. Goswami, A. Masrur, R. Schneider, C. J. Xue, and S. Chakraborty, "Multirate controller design for resource- and schedule-constrained automotive ECUs," in *DATE*, 2013.

[18] R. Schneider, D. Goswami, A. Masrur, M. Becker, and S. Chakraborty, "Multi-layered scheduling of mixed-criticality cyber-physical systems," *J. Syst. Architect., Embedded Syst. Design*, vol. 59, no. 10-D, pp. 1215–1230, 2013.

[19] R. Wilhelm et al., "The worst-case execution-time problem—Overview of methods and survey of tools," *ACM Trans. Embedded Comput. Syst.*, vol. 7, no. 3, pp. 36:1–53, 2008.

[20] R. Wilhelm et al., "Memory hierarchies, pipelines, and buses for future architectures in time-critical embedded systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 7, pp. 966–978, 2009.

[21] S. Andalam, A. Girault, R. Sinha, P. Roop, and J. Reineke, "Precise timing analysis for direct-mapped caches," in *DAC*, 2013.

[22] D. Roy, L. Zhang, W. Chang, D. Goswami, and S. Chakraborty, "Multi-objective co-optimization of FlexRay-based distributed control systems," in *RTAS*, 2016.

[23] D. Goswami, R. Schneider, and S. Chakraborty, "Re-engineering cyber-physical control applications for hybrid communication protocols," in *DATE*, 2011.

[24] A. Masrur et al., "Timing analysis of cyber-physical applications for hybrid communication protocols," in *DATE*, 2012.

[25] H. Voit, A. Annaswamy, R. Schneider, D. Goswami, and S. Chakraborty, "Adaptive switching controllers for systems with hybrid communication protocols," in *ACC*, 2012.

[26] H. Lin and P. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 308–322, 2009.

[27] D. Goswami, R. Schneider, and S. Chakraborty, "Relaxing signal delay constraints in distributed embedded controllers," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2337–2345, 2014.

[28] P. Kumar et al., "A hybrid approach to cyber-physical systems verification," in *DAC*, 2012.

[29] D. Doerffel and S. Sharkh, "A critical review of using the peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries," *J. Power Sources*, vol. 165, no. 2, pp. 395–400, 2006.

[30] T. Kim and W. Qiao, "A hybrid battery model capable of capturing dynamic circuit characteristics and nonlinear capacity effects," *IEEE Trans. Energy Conv.*, vol. 26, no. 4, pp. 1172–1180, 2011.

[31] W. Chang, A. Proebstl, D. Goswami, M. Zamani, and S. Chakraborty, "Battery- and aging-aware embedded control systems for electric vehicles," in *RTSS*, 2014.

[32] D. Rakhmatov and S. Vrudhula, "An analytical high-level battery model for use in energy management of portable electronic systems," in *ICCAD*, 2001.

[33] A. Kumar, D. Sharma, and K. Deb, "A hybrid multi-objective optimization procedure using PCX based NSGA-II and sequential quadratic programming," in *CEC*, 2007.

[34] D. Lorenz, M. Barke, and U. Schlichtmann, "Aging analysis at gate and macro cell level," in *ICCAD*, 2010.

[35] A. Masrur et al., "Schedulability analysis for processors with aging-aware autonomic frequency scaling," in *RTCSA*, 2012.

[36] W. Chang, A. Proebstl, D. Goswami, M. Zamani, and S. Chakraborty, "Reliable CPS design for mitigating

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

semiconductor and battery aging in electric vehicles," in *CPSNA*, 2015.
[37] K. K. Vatanparvar and M. Al Faruque, "OTEM: Optimized thermal and energy management for hybrid electrical energy storage in electric vehicles," in *DATE*, 2016.
[38] D. Shin, M. Poncino, and E. Macii, "Thermal management of batteries using a hybrid supercapacitor architecture," in *DATE*, 2014.
[39] D. Shin et al., "Battery-supercapacitor hybrid system for high-rate pulsed load applications," in *DATE*, 2011.
[40] K. Vatanparvar and M. Al Faruque, "Eco-friendly automotive climate control and navigation system for electric vehicles," in *ICCPS*, 2016.
[41] S. Park, Y. Kim, and N. Chang, "Hybrid energy storage systems and battery management for electric vehicles," in *DAC*, 2013.
[42] K. Vatanparvar, J. Wan, and M. Al Faruque, "Battery-aware energy-optimal electric vehicle driving management," in *ISLPED*, 2015.
[43] W. Chang, M. Lukasiewycz, S. Steinhorst, and S. Chakraborty, "Dimensioning and configuration of EES systems for electric vehicles with boundary-conditioned adaptive scalarization," in *CODES+ISSS*, 2013.
[44] K. Vatanparvar and M. Al Faruque, "Battery lifetime-aware automotive climate control for electric vehicles," in *DAC*, 2015.
[45] M. Al Faruque and K. Vatanparvar, "HVAC system and automotive climate control influence on electric vehicle and battery," in *ASPDAC*, 2016.
[46] K. Koscher et al., "Experimental security analysis of a modern automobile," in *EEE Symp. Security Privacy*, 2010.
[47] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX Security Symp.*, 2011.
[48] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop Embedded Security Cars*, 2004.
[49] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks | practical examples and selected short-term countermeasures," in *SAFECOMP*, 2008.
[50] D. Nilsson, U. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," in *VTC*, 2008.
[51] A. Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth|A simple, backward compatible broadcast authentication protocol for CAN bus," in *Workshop Embedded Security Cars*, 2011.
[52] B. Groza, P. Murvay, A. Herrewege, and I. Verbauwhede, "LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks," in *CANS*, 2012.
[53] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications," in *DSN*, 2009.
[54] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks," in *WESS*, 2010.
[55] A. Wasicek, C. El-Salloum, and H. Kopetz, "Authentication in time-triggered systems using time-delayed release of keys," in *ISORC*, 2011.
[56] C. Szilagyi, "*Low cost multicast network authentication for embedded control systems*," PhD dissertation, Carnegie Mellon Univ., 2012.
[57] C. Lin, Q. Zhu, C. Phung, and A. Sangiovanni-Vincentelli, "Security-aware mapping for CAN-based real-time distributed automotive systems," in *ICCAD*, 2013.
[58] C. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware mapping for TDMA-based real-time distributed systems," in *ICCAD*, 2014.
[59] C. Lin, B. Zheng, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware design methodology and optimization for automotive systems," *ACM Trans. Design Autom. Electron. Syst.*, vol. 21, no. 1, pp. 18:1–18:26, 2015.
[60] B. Zheng, P. Deng, R. Anguluri, Q. Zhu, and F. Pasqualetti, "Cross-layer codesign for secure cyber-physical systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2016.

Samarjit Chakraborty is a Professor of Electrical Engineering at the Technical University of Munich (TUM), Munich, Germany, where he holds the Chair for Real-Time Computer Systems. His research interests cover all aspects of embedded and cyber–physical systems and software design. Chakraborty has a PhD in electrical and computer engineering from ETH Zurich, Zurich, Switzerland (2003).

Mohammad Abdullah Al Faruque is currently with the University of California Irvine (UCI), Irvine, CA, USA, where he is a tenure track Assistant Professor and directing the Cyber-Physical Systems Lab. His current research is focused on system-level design of embedded systems and

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.

cyber–physical systems (CPSs) with special interest on model-based design and CPS security. Al Faruque has a PhD in computer engineering from Karlsruhe Institute of Technology, Karlsruhe, Germany (2009).

Wanli Chang is currently a Research Associate at TUM CREATE, Singapore, focusing on resource-aware automotive control systems. Chang has an MS in communications engineering from Technical University of Munich, Munich, Germany (with distinction, 2010), where he is currently working toward a PhD.

Dip Goswami is an Assistant Professor at the Electrical Engineering Department, Eindhoven University of Technology, Eindhoven, The Netherlands. His research interests include embedded control systems, robotics, and cyber–physical systems. Goswami has a PhD in electrical and computer engineering from the National University of Singapore (NUS), Singapore (2009).

Marilyn Wolf is the Rhesa "Ray" S. Farmer, Jr. Distinguished Chair in Embedded Computing Systems and Georgia Research Alliance Eminent Scholar at the Georgia Institute of Technology, Atlanta, GA, USA. Her research interests include cyber–physical systems, embedded computing, embedded video and computer vision, and VLSI systems. Wolf has a PhD in electrical engineering from Stanford University, Stanford, CA, USA (1984). She is a Fellow of the IEEE and the Association for Computing Machinery (ACM) and an IEEE Computer Society Golden Core member.

Qi Zhu is an Assistant Professor at the Electrical and Computer Engineering Department, University of California, Riverside, Riverside, CA, USA. His research interests include model-based design and software synthesis for cyber–physical systems, CPS security, energy-efficient buildings, and system-on-chip design. Zhu has a PhD in electrical engineering and computer science from the University of California Berkeley, Berkeley, CA, USA.

■ Direct questions and comments about this article to Samarjit Chakraborty, Technical University of Munich, D-80290 Munich, Germany; samarjit@tum.de.

Authorized licensed use limited to: IEEE Xplore. Downloaded on September 01,2024 at 06:25:53 UTC from IEEE Xplore. Restrictions apply.