European Journal of Control (2005)11:1–31

European
Journal of
Control

Hybrid Control of Networked Embedded Systems

A. Balluchi, L. Benvenuti, S. Engell, T. Geyer, K.H. Johansson,
F. Lamnabhi-Lagarrigue*, J. Lygeros, M. Morari, G. Papafotiou,
A.L. Sangiovanni-Vincentelli, F. Santucci and O. Stursberg

Laboratoire des Signaux et Systemes, Centre National de la Recherche Scientifique 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France

Hybrid systems that involve the interaction of continuous and discrete dynamics have been an active area of research for a number of years. In this paper, we start by briefly surveying the main theoretical control problems that have been treated in the hybrid systems setting and classify them into stabilization, optimal control and language specification problems. We then provide an overview of recent developments in four of the most prominent areas where these hybrid control methods have found application: control of power systems, industrial process control, design of automotive electronics and communication networks.

Keywords: Hybrid Systems; Optimal Control; Stabilization

1. Introduction

The term hybrid systems is used in the literature to refer to systems that feature an interaction between diverse types of dynamics. Most heavily studied in recent years are hybrid systems that involve the interaction between continuous and discrete dynamics. The study of this class of systems has to a large extent been motivated by applications to embedded systems and control. Embedded systems by definition involve the interaction between digital devices and a predominantly analog environment. In addition, much of the design complexity of embedded systems comes from the fact that they have to meet specifications, such as hard real-time constraints and scheduling constraints, that involve a mixture of discrete and continuous requirements. Therefore, both the model and the specifications of embedded systems can naturally be expressed in the context of hybrid systems.

Control problems have been at the forefront of hybrid systems research from the very beginning. The reason is that many important applications with prominent hybrid dynamics come from the area of embedded control. For example, hybrid control has played an important role in applications to avionics, automated highways, communication networks, automotive control, air traffic management, industrial process control, manufacturing and robotics.

The Network of Excellence HyCon [66] aims to consolidate and promote research on hybrid systems throughout Europe. The network provides the means to coordinate the research of 23 European research institutions in the areas of modeling, analysis and control of hybrid systems with applications to power management, automotive control, industrial processes and communication networks. In addition to research integration, the activities of the HyCon network also include collaborative efforts in the teaching of hybrid systems, the development of a virtual library of hybrid systems literature, the establishment of a hybrid systems tool repository and the establishment of common benchmarks on which to test novel hybrid systems methods.

---

*E-mail: lamnabhi@lss.supelec.fr
Correspondence to: J. Lygeros.
E-mail: lygeros@ee.upatras.gr

Received 1 May 2005; Accepted 17 June 2005.

In this overview paper we review recent results emerging from research currently being conducted under the HyCon umbrella. We concentrate on results dealing with novel hybrid control methods and their applications. We start by surveying and classifying the control problems that have been investigated in the hybrid systems literature (Section 2). We then highlight recent developments in four key application areas of hybrid control: control of power electronics (Section 3), industrial process control (Section 4), automotive control (Section 5) and communication systems (Section 6). We conclude the paper with a discussion of the open problems, research challenges and vistas (Section 7).

2. An Overview of Hybrid Control Problems

2.1. Control Problem Classification

The control problems that have been studied in the literature differ, first of all, in the way in which they treat uncertainty. Generally, the problems can be grouped into three classes:

1. *Deterministic*. Here it is assumed that there is no uncertainty; control inputs are the only class of inputs considered.
2. *Non-deterministic*. In this case inputs are grouped into two classes, control and disturbance. The design of a controller for regulating the control inputs assumes that disturbance inputs are adverserial. Likewise, the requirements are stated as worst case: the controller should be such that the specifications are met for all possible actions of the disturbance. From a control perspective, problems in this class are typically framed in the context of robust control or game theory.
3. *Stochastic*. Again, both control and disturbance inputs are considered. The difference with the non-deterministic case is that a probability distribution is assumed for the disturbance inputs. This extra information can be exploited by the controller and also allows one to formulate finer requirements. For example, it may not be necessary to meet the specifications for all disturbances, as long as the probability of meeting them is high enough.

In addition, the control problems studied in the literature differ in the specifications they try to meet:

1. *Stabilization*. Here the problem is to select the continuous inputs and/or the timing and destinations of discrete switches to make sure that the system remains close to an equilibrium point, limit cycle or other invariant set. Many variants of this problem have been studied in the literature. They differ in the type of control inputs considered (discrete, continuous or both) and the type of stability specification (stabilization, asymptotic or exponential stabilization, practical stabilization, etc.). Even more variants have been considered in the case of stochastic hybrid systems (stability in distribution, moment stability, almost sure asymptotic stability, etc.).
2. *Optimal control*. Here the problem is to steer the hybrid system using continuous and/or discrete controls in a way that minimizes a certain cost function. Again, different variants have been considered, depending on whether discrete and/or continuous inputs are available, whether cost is accumulated along continuous evolution and/or during discrete transitions, whether the time horizon over which the optimization is carried out is finite or infinite, etc.
3. *Language specifications*. Control problems of great interest can also be formulated by imposing the requirement that the trajectories of the closed-loop system are all contained in a set of desirable trajectories. Typical requirements of this type arise from reachability considerations, either of the safety type (along all trajectories the state of the system should remain in a "good" region of the state space), or of the liveness type (the state of the system should eventually reach a "good" region of the state space along all trajectories). Starting with these simple requirements, progressively more and more complex specifications can be formulated: the state should visit a given set of states infinitely often, given two sets of states, if the state visits one infinitely often it should also visit the other infinitely often, etc. These specifications are all related to the "language" generated by the closed-loop system and have been to a large extent motivated by analogous problems formulated for discrete systems based on temporal logic.

In this section we present an overview of the problems that have been addressed in the literature in these classes. We start by briefly introducing some modeling concepts necessary to highlight the differences between the different problems. We then discuss stabilization, optimal control and language specification problems in separate subsections.

2.2. A Simple Hybrid Control Model

Hybrid control problems have been formulated for both continuous- and discrete-time systems. In this section we introduce a model suitable for formulating continuous-time control problems for hybrid systems; a class of discrete time models is introduced in Section 3. We restrict our attention to hybrid systems that do not include any probabilistic phenomena; the formal definition of stochastic hybrid models requires considerable mathematical overhead, even in the simplest cases.

Since we are interested in hybrid dynamics, the dynamical systems we consider involve both a continuous state (denoted by $x \in X = \mathbb{R}^n$) and a discrete state (denoted by $q \in Q$). To allow us to capture the different types of uncertainties discussed above, we also assume that the evolution of the state is influenced by two different kinds of inputs: controls and disturbances. We assume that inputs of each kind can be either discrete or continuous, and we use $\upsilon \in \Upsilon$ to denote discrete controls, $u \in U \subseteq \mathbb{R}^m$ to denote continuous controls, $\delta \in \Delta$ to denote discrete disturbances and $d \in D \subseteq \mathbb{R}^p$ to denote continuous disturbances. The sets Q, Υ and Δ are assumed to be countable or finite.

The dynamics of the state are determined through four functions: a vector field $f : Q \times X \times U \times D \to X$ that determines the continuous evolution, a reset map $r : Q \times Q \times X \times U \times D \to X$ that determines the outcome of the discrete transitions, "guard" sets $G : Q \times Q \times \Upsilon \times \Delta \to 2^X$ that determine when discrete transitions can take place and "domain" sets $\text{Dom} : Q \times \Upsilon \times \Delta \to 2^X$ that determines when continuous evolution is possible.[1] To avoid pathological situations (lack of solutions, deadlock, chattering, etc.) one needs to introduce technical assumptions on the model components. Typically, these include continuity assumptions on f and r, compactness assumptions on U and D, and convexity assumptions on $\bigcup_{u \in U} f(q, x, u, d)$ and $\bigcup_{d \in D} f(q, x, u, d)$, etc. As for continuous systems, these assumptions aim to ensure that for all $q \in Q$, $x_0 \in X$ and $u(\cdot)$, $d(\cdot)$ measurable functions of time, the differential equation $\dot{x}(t) = f(q, x(t), u(t), d(t))$ has a unique solution $x(\cdot) : \mathbb{R}_+ \to X$ with $x(0) = x_0$. Additional assumptions are often imposed to prevent deadlock, a situation where it is not possible to proceed by continuous evolution or by discrete transition. Finally, in many publications assumptions are introduced to prevent what is called the Zeno phenomenon, a situation where the solution of the system takes an infinite number of discrete transitions in a finite amount of time. The Zeno phenomenon can prove particularly problematic for hybrid control problems, since it may be exploited either by the control or by the disturbance variables. For example, a controller may appear to meet a safety specification by forcing all trajectories of the system to be Zeno. This situation is undesirable in practice, since the specifications are met not because of successful controller design but because of modeling over-abstraction.

The solutions of this class of hybrid systems can be defined using the notion of a hybrid time set [54]. A hybrid time set $\tau = \{I_i\}_{i=0}^{N}$ is a finite or infinite sequence of intervals of the real line, such that for all $i < N$, $I_i = [\tau_i, \tau_i']$ with $\tau_i \leq \tau_i' = \tau_{i+1}$ and, if $N < \infty$, then either $I_N = [\tau_N, \tau_N']$ or $I_N = [\tau_N, \tau_N')$, possibly with $\tau_N' = \infty$. Since the dynamical systems considered here are time invariant, without loss of generality we can assume that $\tau_0 = 0$.

Roughly speaking, solutions of the hybrid systems considered here (often called "runs" or "executions") are defined together with their hybrid time sets and involve a sequence of intervals of continuous evolution followed by discrete transitions. Starting at some initial state (q_0, x_0), the continuous state moves along the solution of the differential equation $\dot{x} = f(q_0, x, u, d)$ as long as it does not leave the set $\text{Dom}(q_0, \upsilon, \delta)$. The discrete state remains constant throughout this time. If at some point x reaches a set $G(q_0, q', \upsilon, \delta)$ for some $q' \in Q$, a discrete transition can take place. The first interval of τ ends and the second one begins with a new state (q', x') where x' is determined by the reset map r. The process is then repeated. Notice that considerable freedom is allowed when defining the solution in this "declarative" way: in addition to the effect of the input variables, there may also be a choice between evolving continuously or taking a discrete transition (if the continuous state is in both the domain set and a guard set) or between multiple discrete transitions (if the continuous state is in many guard sets at the same time).

A bit more formally, a run of the hybrid system can be defined as a collection $(\tau, q, x, \upsilon, u, \delta, d)$ consisting of a hybrid time set $\tau = \{I_i\}_{i=0}^{N}$ and sequences of functions $q = \{q_i(\cdot) : I_i \to Q\}_{i=0}^{N}$, $x = \{x_i(\cdot) : I_i \to X\}_{i=0}^{N}$, etc. that satisfy the following conditions:

- Discrete evolution: for $i < N$,
  1. $x_i(\tau_i') \in G(q_i(\tau_i'), q_{i+1}(\tau_{i+1}), \upsilon_i(\tau_i'), \delta_i(\tau_i'))$.
  2. $x_{i+1}(\tau_{i+1}) = r(q_i(\tau_i'), q_{i+1}(\tau_{i+1}), x_i(\tau_i'), u_i(\tau_i'), d_i(\tau_i'))$.
- Continuous evolution: for all i with $\tau_i < \tau_i'$
  1. $u_i(\cdot)$ and $d_i(\cdot)$ are measurable functions.
  2. $q_i(t) = q_i(\tau_i)$ for all $t \in I_i$.

---

[1] As usual, 2^X stands for the set of all subsets of X.

(3) $x_i(\cdot)$ is a solution to $\dot{x}_i(t) = f(q_i(t), x_i(t), u_i(t), d_i(t))$ over the interval I_i starting at $x_i(\tau_i)$.
(4) $x_i(t) \in \mathrm{Dom}(q_i(t), \upsilon_i(t), \delta_i(t))$ for all $t \in [\tau_i, \tau_i')$.

This model allows control and disturbance inputs to influence the evolution of the system in a number of ways. In particular, control and disturbance can

(1) Steer the continuous evolution through the effect of u and d on the vector field f.
(2) Force discrete transitions to take place through the effect of υ and δ on the domain Dom.
(3) Affect the discrete state reached after a discrete transition through the effect of υ and δ on the guards G.
(4) Affect the continuous state reached after a discrete transition through the effect of u and d on the reset function r.

An issue that arises is the type of controllers one allows for selecting the control inputs u and υ. The most common control strategies considered in the hybrid systems literature are, of course, static feedback strategies. In this case the controller can be thought of as a map (in general set valued) of the form $g : Q \times X \to 2^{\Upsilon \times U}$. For controllers of this type, the runs of the closed-loop system can easily be defined as runs, $(\tau, q, x, \upsilon, u, \delta, d)$, of the uncontrolled system such that for all $I_i \in \tau$ and all $t \in I_i$, $(\upsilon_i(t), u_i(t)) \in g(q_i(t), x_i(t))$.

It turns out that for certain kinds of control problems one can restrict attention to feedback controllers without loss of generality. For other problems, however, one may be forced to consider more general classes of controllers: dynamic feedback controllers that incorporate observers for output feedback problems, controllers that involve non-anticipative strategies for gaming problems, piecewise constant controllers to prevent chattering, etc. Even for these types of controllers, it is usually intuitively clear what one means by the runs of the closed-loop system. However, unlike feedback controllers, a formal definition would require one to formulate the problem in a compositional hybrid systems framework and formally define the closed-loop system as the composition of a plant and a controller automaton.

2.3. Stabilization of Hybrid Systems

For stabilization, the aim is to design controllers such that the runs of the closed-loop system remain close and possibly converge to a given invariant set. An invariant set is a set of states with the property that runs starting in the set remain in the set forever. More formally, $W \subseteq Q \times X$ is an invariant set if for all $(\hat{q}, \hat{x}) \in W$ and all runs $(\tau, q, x, \upsilon, u, \delta, d)$ starting at $(\hat{q}, \hat{x})$,

$$(q_i(t), x_i(t)) \in W, \quad \forall I_i \in \tau, \ \forall t \in I_i.$$

The most common invariant sets are those associated with equilibria, points $\hat{x} \in X$ that are preserved under both discrete and continuous evolution.

The definitions of stability can naturally be extended to hybrid systems by defining a metric on the hybrid state space. An easy way to do this is to consider the Euclidean metric on the continuous space and the discrete metric on the discrete space $[d_D(q, q') = 0$ if $q = q'$ and $d_D(q, q') = 1$ if $q \neq q']$ and define the hybrid metric by

$$d_H((q, x), (q', x')) = d_D(q, q') + \|x - x'\|.$$

The metric notation can be extended to sets in the usual way. Equipped with this metric, the standard stability definitions (Lyapunov stability, asymptotic stability, exponential stability, practical stability, etc.) naturally extend from the continuous to the hybrid domain. For example, an invariant set, W, is called stable if for all $\epsilon > 0$ there exists $\epsilon' > 0$ such that for all $(q, x) \in Q \times X$ with $d_H((q, x), W) < \epsilon'$ and all runs $(\tau, q, x, \upsilon, u, \delta, d)$ starting at (q, x),

$$d_H((q_i(t), x_i(t)), W) < \epsilon, \quad \forall I_i \in \tau, \ \forall t \in I_i.$$

Stability of hybrid systems has been extensively studied in recent years (see the overview papers [30,51]). By comparison, the work on stabilization problems is relatively sparse. A family of stabilization schemes assumes that the continuous dynamics are given, for example, stabilizing controllers have been designed for each $f(q, \cdot, \cdot, \cdot)$. Procedures are then defined for determining the switching times (or at least constraints on the switching times) to ensure that the closed-loop system is stable, asymptotically stable, or practically stable [46,72,85,88]. Stronger results are possible for special classes of systems, such as planar systems [87]. For non-deterministic systems, in Ref. [34] an approach to the practical exponential stabilization of a class of hybrid systems with disturbances is presented. Issues related to the stability and stabilization of systems controlled over communication networks are highlighted in Section 6.

2.4. Optimal Control of Hybrid Systems

In optimal control problems it is typically assumed that a cost is assigned to the different runs of the hybrid system. The objective of the controller is then to minimize this cost by selecting the values of the control variables appropriately. Typically, the cost function assigns a cost to both continuous evolution and discrete transitions. For example, for the cost assigned to a run $(\tau, q, x, \upsilon, u, \delta, d)$ with $\tau = \{I_i\}_{i=0}^{N}$, the cost function may have the form

$$\sum_{i=0}^{N} \Bigg[\int_{\tau_i}^{\tau_i'} l(q_i(t), x_i(t), u_i(t), d_i(t))\,\mathrm{d}t + g(q_i(\tau_i'), x_i(\tau_i'), q_{i+1}(\tau_{i+1}), x_{i+1}(\tau_{i+1}), u_i(\tau_i), d_i(\tau_i), \upsilon_i(\tau_i'), \delta_i(\tau_i')) \Bigg],$$

where $l : Q \times X \times U \times D \to \mathbb{R}$ is a function assigning a cost to the pieces of continuous evolution and $g : Q \times X \times Q \times X \times U \times D \times \Upsilon \times \Delta \to \mathbb{R}$ is a function assigning a cost to discrete transitions. Different variants of optimal control problems can be formulated, depending on, e.g., the type of cost function, the horizon over which the optimization takes place (finite or infinite) or whether the initial and/or final states are specified. Examples of problems of this type that arise in the control of power networks and industrial processes are discussed in Sections 3 and 4, respectively.

As with continuous systems, two different approaches have been developed for addressing such optimal control problems. One is based on the maximum principle and the other on dynamic programming. Extensions of the maximum principle to hybrid systems have been proposed by numerous authors; see Refs [40,73,80]. The solution of the optimal control problem with the dynamic programming approach typically requires the computation of a value function, which is characterized as a viscosity solution to a set of variational or quasi-variational inequalities [19,23]. Computational methods for solving the resulting variational and quasi-variational inequalities are presented in Ref. [58]. For simple classes of systems [e.g., timed automata (TA)] and simple cost functions (e.g., minimum time problems) it is often possible to exactly compute the optimal cost and optimal control strategy, without resorting to numerical approximations (see Ref. [20] and the references therein).

A somewhat different optimal control problem arises when one tries to control hybrid systems using model predictive or receding horizon techniques. This approach is discussed in greater detail in Section 3, in the context of power system control.

2.5. Language Specification Problems

Another type of control problem that has attracted considerable attention in the hybrid systems literature revolves around language specifications. One example of language specifications is the *safety specifications*. In this case a "good" set of states $W \subseteq Q \times X$ is given and the designer is asked to produce a controller that ensures that the state always stays in this set; in other words, for all runs $(\tau, q, x, \upsilon, u, \delta, d)$ of the closed-loop system

$$\forall I_i \in \tau \;\; \forall t \in I_i, \quad (q_i(t), x_i(t)) \in W.$$

The name "safety specifications" (which has a formal meaning in computer science) intuitively refers to the fact that such specifications can be used to encode safety requirements in a system, to ensure that nothing bad happens, e.g., ensure that vehicles in an automated highway system (see the discussion in Section 6) do not collide with one another. An example of a control problem of this type that arises in the area of industrial processes is discussed in Section 4.

Safety specifications are usually easy to meet, e.g., if no vehicles are allowed on the highway collisions are impossible. To make sure that in addition to being safe the system actually does something useful, liveness specifications are usually also imposed. The simplest type of *liveness specification* deals with reachability: given a set of states $W \subseteq Q \times X$, design a controller such that for all runs $(\tau, q, x, \upsilon, u, \delta, d)$ of the closed-loop system

$$\exists I_i \in \tau \;\; \exists t \in I_i, \quad (q_i(t), x_i(t)) \in W.$$

In the automated highways context a minimal liveness type requirement is to make sure that the vehicles eventually arrive at their destination. Mixing different types of specifications similar to the ones given above one can construct arbitrarily complex properties, e.g., ensure that the state visits a set infinitely often, ensure that it reaches a set and stays there forever after, etc. Such complex *language specifications* are usually encoded formally using temporal logic notation.

Controller design problems under language specifications have been studied very extensively for discrete systems in the computer science literature. The approach was then extended to classes of hybrid systems, such as TA (systems with continuous dynamics of the form $\dot{x} = 1$, [5]) and rectangular automata (systems with continuous dynamics of the form $\dot{x} \in [l, u]$ for fixed parameters l, u, [86]). For systems of this type, exact and automatic computation of the controllers may be possible using model checking tools [18,28,43]. In all these cases the controller affects only the discrete aspects of the system evolution, i.e., the destination and timing of discrete transitions. More general language problems (e.g., where the dynamics are linear, the controller affects the continuous motion of the system) can often be solved automatically for discrete time systems using methods from mathematical programming (see Section 3 for a discussion).

Extensions to general classes of hybrid systems in continuous time have been concerned primarily with computable numerical approximations of reachable sets using polyhedral approximations [4,25,70], ellipsoidal approximations [22] or more general classes of sets. A useful link in this direction has been the relationship between reachability problems and optimal control problems with an l_∞ penalty function [82]. This link has allowed the development of numerical tools that use partial differential equation solvers to approximate the value function of the optimal control problems and hence indirectly characterize reachable sets [58].

3. Model Predictive Control in Power Electronics

3.1. Control Problems in Power Electronics

Power electronics systems represent a well-established technology that has seen significant performance improvements over the last two decades. In general, these systems are used to transform electrical power from one – usually unregulated – form to another regulated one. This transformation is achieved by the use of semiconductor devices that operate as power switches, turning on and off with a high switching frequency. From the control point of view, power electronic circuits and systems constitute excellent examples of hybrid systems, since the discrete switch positions are associated with different continuous-time dynamics. Moreover, both physical and safety constraints are present.

Power electronics circuits and systems have traditionally been controlled in industry using linear controllers combined with non-linear procedures such as pulse width modulation (PWM). The models used for controller design are a result of simplifications that include averaging the behavior of the system over time (to avoid modeling the switching) and linearizing around a specific operating point disregarding all constraints. As a result, the derived controller usually performs well only in a neighborhood around the operating point. To make the system operate in a reliable way for the whole operating range, the control circuit is subsequently augmented by a number of heuristic patches. The result of this procedure are large development times and the lack of theoretically backed guarantees for the operation of the system; in particular, no global stability guarantees can be given.

Recent theoretical advances in the field of hybrid systems, together with the availability of significant computational power for the control loops of power electronics systems, are inviting both the control and the power electronics communities to revisit the control issues associated with power electronics applications. Such an effort for a novel approach to controlling power electronics systems is outlined in this section, where we demonstrate the application of hybrid optimal control methodologies to power electronics systems. More specifically, we show how model predictive control (MPC) [55] can be applied to problems of induction motor drives and dc–dc conversion illustrating the procedure using two examples: the direct torque control (DTC) of three-phase induction motors and the optimal control of fixed-frequency switch-mode dc–dc converters.

The use of optimal control methodologies implies the solution of an underlying optimization problem. Given the high switching frequency that is used in power electronics applications and the large solution times that are usually needed for such optimization problems, solving this problem on-line may very well be infeasible. Depending on the application, this obstacle can be overcome in two ways; either by pre-solving off-line the optimization problem for the whole state-space using multi-parametric programming, a procedure that results in a polyhedral piecewise affine (PWA) controller that can be stored in a look-up table or by developing solution algorithms that are dedicated, tailored to the problem and can thus be executed within the limited time available. The first approach has been followed here for the optimal control of fixed-frequency dc–dc converters, whereas the second one has been applied to the DTC problem.

3.2. Optimal Control of Discrete Time Hybrid Systems

In the following, we restrict ourselves to the discrete-time domain, and we confine our models to (piecewise) affine dynamics rather than allowing general non-linear dynamics. This not only avoids a number of mathematical problems (like the Zeno behavior discussed in Section 2), but allows us to derive models for which we can pose analysis and optimal control problems that are computationally tractable. To model such discrete-time linear hybrid systems, we adopt mixed logical dynamical (MLD) [16] models and the PWA [76] framework. Other representations of such systems include Linear Complementarity (LC) systems, Extended LC (ELC) systems and Max-Min-Plus-Scaling systems that are, as shown in Ref. [42], equivalent to the MLD and PWA forms under mild conditions.

MPC [55] has been used successfully for a long time in the process industry and recently also for hybrid systems, for which, as shown in Ref. [16], MPC has proven to be particularly well suited. The control action is obtained by minimizing an objective function over a finite or infinite horizon subject to the evolution in time of the model of the controlled process and constraints on the states and manipulated variables. For linear hybrid systems, depending on the norm used in the objective function, this minimization problem amounts to solving a Mixed-Integer Linear Program (MILP) or Mixed-Integer Quadratic Program.

The major advantage of MPC is its straightforward design procedure. Given a (linear or hybrid) model of the system, one only needs to set up an objective function that incorporates the control objectives. Additional hard (physical) constraints can be easily dealt with by adding them as inequality constraints, whereas soft constraints can be accounted for in the objective function using large penalties. For details concerning the set up of the MPC formulation in connection with linear hybrid models, the reader is referred to Ref. [16]. Details about MPC can be found in Ref. [55].

To make the proposed optimal control strategies applicable to power electronics systems it is mandatory to overcome the obstacle posed by the large computation times occurring when solving the optimal control problem on-line. This can be achieved by pre-computing the optimal state-feedback control law off-line for all feasible states using the state vector as a parameter. For hybrid systems, such a method has been recently introduced, which is based on a PWA description of the controlled system and a linear objective function, using the 1- or ∞-norm. As shown in Ref. [21], the resulting optimal state-feedback control law is a PWA function of the state defined on a polyhedral partition of the feasible state-space. More specifically, the state-space is partitioned into polyhedral sets and for each of these sets the optimal control law is given as an affine function of the state. As a result, such a state-feedback controller can be implemented easily on-line as a look-up table.

3.3. Optimal DTC of Three-Phase Induction Motors

The rapid development of power semiconductor devices led to the increased use of adjustable speed induction motor drives in a variety of applications. In these systems, dc–ac inverters are used to drive induction motors as variable frequency three-phase voltage or current sources. One methodology for controlling the torque and speed of induction motor drives is DTC [81], which features very favorable control performance and implementation properties.

The basic principle of DTC is to exploit the fast dynamics of the motor's stator flux and to directly manipulate the stator flux vector such that the desired torque is produced. This is achieved by choosing an inverter switch combination that drives the stator flux vector to the desired position by directly applying the appropriate voltages to the motor windings. This choice is made usually with a sampling time $T_s = 25\,\mu$s using a pre-designed switching table that is traditionally derived in a heuristic way and, depending on the particularities of the application, addresses a number of different control objectives. These primarily concern the induction motor – more specifically, the stator flux and the electromagnetic torque need to be kept within pre-specified bounds around their references. In high power applications, where three-level inverters with Gate Turn-Off thyristors are used, the control objectives are extended to the inverter and also include the minimization of the average switching frequency and the balancing of the inverter's neutral point potential around zero. As mentioned in the Introduction section, because of the discrete switch positions of the inverter, the DTC problem is a hybrid control problem, which is complicated by the nonlinear behavior of the torque, length of stator flux and the neutral point potential (Fig. 1).

We aim at deriving MPC schemes that keep the three controlled variables (torque, flux and neutral point potential) within their given bounds, minimize the (average) switching frequency, and are conceptually and computationally simple yet yield a significant performance improvement with respect to the state of the art. More specifically, the term *conceptually simple* refers to controllers allowing for straightforward tuning of the controller parameters or even a lack of such parameters, and easy

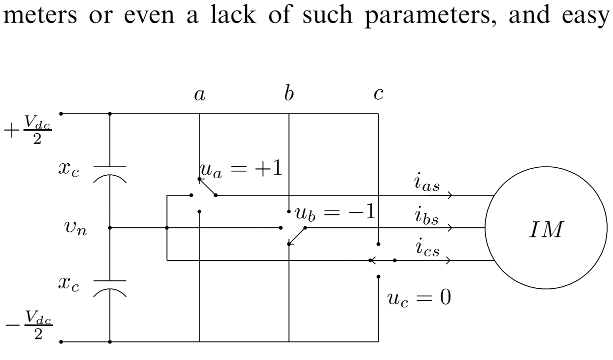

Fig. 1. The equivalent representation of a three-phase three-level inverter driving an induction motor.

adaptation to different physical setups and drives, whereas *computationally simple* implies that the control scheme does not require excessive computational power to allow the implementation on DTC hardware that is currently available or at least will be so within a few years.

An important first step is to derive discrete-time hybrid models of the drive tailored to our needs – or more specifically, models that are of low complexity yet of sufficient accuracy to serve as prediction models for our model-based control schemes. To achieve this, we have exploited in Ref. [69] a number of physical properties of DTC drives. These properties are the (compared with the stator flux) slow rotor flux and speed dynamics, the symmetry of the voltage vectors, and the invariance of the motor outputs under flux rotation. The low-complexity models are derived by assuming constant speed within the prediction horizon, mapping the states (the fluxes) into a 60° sector, and aligning the rotor flux vector with the *d*-axis of the orthogonal dq0 reference frame rotating with the rotational speed of the rotor [50]. The benefits of doing this are a reduction of the number of states from five to three, and a highly reduced domain on which the nonlinear functions need to be approximated by PWA functions.

Based on the hybrid models of the DTC drive, we have proposed in Refs [36,37,69] three novel control approaches to tackle the DTC problem, which are inspired by the principles of MPC and tailored to the peculiarities of DTC. For comparing with the industrial state of the art, we have used for all our simulations the Matlab/Simulink model of ABB's ACS6000 DTC drive [1] containing a squirrel-cage rotor induction motor with a rated apparent power of 2 MVA and a 4.3 kV three-leveldc-link inverter. This model was provided to us by ABB in the framework of our collaboration and its use ensures a realistic set-up.

3.3.1. DTC Based on Priority Levels

The first scheme [69] uses soft constraints to model the hysteresis bounds on the torque, stator flux and neutral point potential, and approximates the average switching frequency (over an infinite horizon) by the number of switch transitions over a short horizon. To make this approximation meaningful and to avoid excessive switching, one needs to enforce that switch transitions are only performed if absolutely necessary, i.e., when refraining from switching would lead to a violation of the bounds on the controlled variables within one time-step. This means that the controller has to postpone any scheduled switch transition until absolutely necessary. This strategy can be implemented by imposing a time-decaying penalty on the switch transitions, where switch transitions within the first time-step of the prediction interval result in larger penalties than those that are far in the future. Moreover, three penalty levels with corresponding penalties of different orders of magnitude provide clear controller priorities and make the fine-tuning of the objective function obsolete. To extend the prediction interval without increasing the computational burden, we propose to use a rather long prediction interval, but a short prediction horizon. This is achieved by finely sampling the prediction model with T_s only for the first steps, but more coarsely with a multiple of T_s for steps far in the future. This approach is similar to utilizing the technique of blocking control moves and leads to a time-varying prediction model with different sampling rates.

Simulation results demonstrating the behavior of the controlled variables under this control scheme are presented in Fig. 2. This control scheme not only leads to short commissioning times for DTC drives, but also it leads to a performance improvement in terms of a reduction of the switching frequency in the range of 20% with respect to the industrial state of the art,

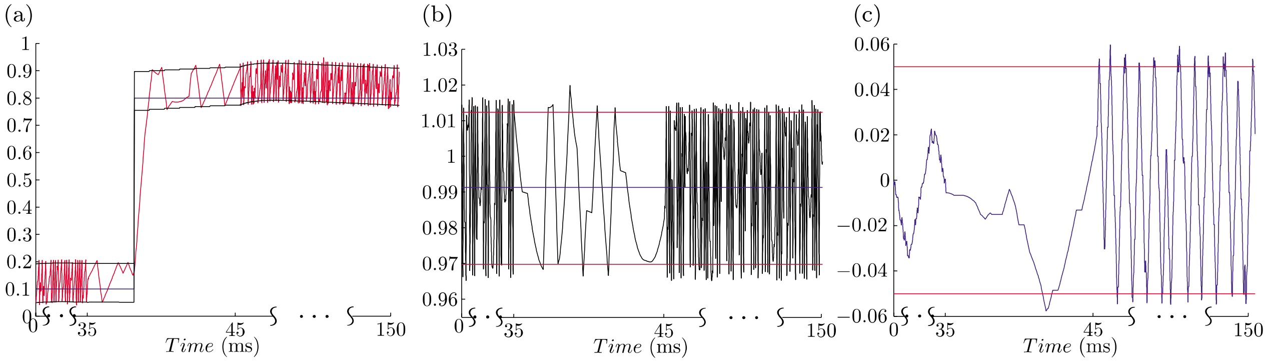

Fig. 2. Closed-loop simulation of the DTC scheme based on priority levels during a step change in the torque reference. The units in the figures are given in p.u. (a) Electromagnetic torque. (b) Stator flux. (c) Neutral point potential.

while simultaneously reducing the torque and flux ripples. Yet the complexity of the control law is rather excessive.

3.3.2. DTC Based on Feasibility and Move Blocking

The second scheme, presented in Ref. [36], exploits the fact that the control objectives only weakly relate to optimality but rather to feasibility, in the sense that the main objective is to find a control input sequence that keeps the controlled variables within their bounds, i.e., a control input sequence that is feasible. The second, weaker objective is to select among the set of feasible control input sequences the one that minimizes the average switching frequency, which is again approximated by the number of switch transitions over the (short) horizon. We therefore propose an MPC scheme based on feasibility in combination with a move blocking strategy, where we allow for switching only at the current time-step. For each input sequence, we determine the number of steps the controlled variables are kept within their bounds, i.e., remain feasible. The switching frequency is emulated by the cost function, which is defined as the number of switch transitions divided by the number of predicted time-steps an input remains feasible, and the control input is chosen so that it minimizes this cost function.

As shown in Ref. [36], the simplicity of the control methodology translates into a state-feedback control law with a complexity that is of an order of magnitude lower than the one of the first scheme, while the performance is improved.

3.3.3. DTC Based on Extrapolation

The third scheme [37] can be interpreted as a combination of the two preceding concepts. Specifically, we use a rather short horizon and compute for the input sequences over the horizon the evolution of the controlled variables using the prediction model. To emulate a long horizon, the "promising" trajectories are extrapolated and the number of steps is determined when the first controlled variable hits a bound. The cost of each input sequence is then determined by dividing the total number of switch transitions in the sequence by the length of the extrapolated trajectory. Minimizing this cost yields the optimal input sequence and the next control input to be applied.

The major benefits of this scheme are its superior performance in terms of switching frequency, which is reduced over the whole range of operating points by up to 50%, with an average reduction of 25%. This performance improvement is shown in Fig. 3, where the switching frequency of the developed control scheme is compared with the one achieved with ABB's currently employed approach [1]. Furthermore, the controller needs no tuning parameters.

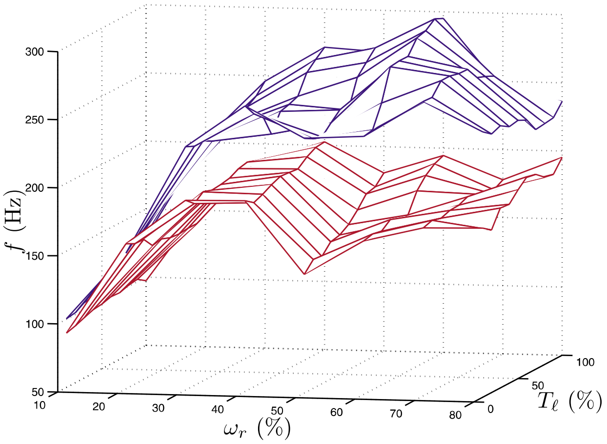

Fig. 3. Comparison of switching frequency f of ABB's DTC (upper surface) with respect to MPC based on extrapolation (lower surface) over the grid of operating points.

Summing up, at every discrete sampling instant, all control schemes use an internal model of the DTC drive to predict the output response to input sequences, choose the input sequence that minimizes an approximation of the average switching frequency, apply only the first element of the input sequence according to the receding horizon policy. Moreover, the proposed schemes are tailored to a varying degree to the specific DTC problem set-up. Starting from the first scheme, the complexity of the controllers in terms of computation times and the memory requirement for the controller hardware were steadily reduced by several orders of magnitude, while the performance was steadily improved. Since the switching losses of the inverter are roughly proportional to the switching frequency, the performance improvement in terms of the switching frequency reduction translates into energy savings and thus into a more cost efficient operation of the drive, which is especially important because high power applications are considered here. Most importantly, the last control scheme (based on extrapolation) is currently being implemented by our industrial partner ABB who has also protected this scheme by a patent application [37].

3.4. Optimal Control of dc–dc Converters

Switch-mode dc–dc converters are switched circuits that transfer power from a dc input to a load. They are used in a large variety of applications owing to their light weight, compact size, high efficiency and reliability. Since the dc voltage at the input is unregulated (consider, for example, the result of a coarse ac rectification) and the output power demand changes significantly over time constituting a time-varying load, the scope is to achieve output voltage regulation in the presence of input voltage and output load variations.

Fixed-frequency switch-mode dc–dc converters use semiconductor switches that are periodically switched on and off, followed by a low-pass filtering stage with an inductor and a capacitor to produce at the output a dc voltage with a small ripple. Specifically, the switching stage comprises a primary semiconductor switch that is always controlled, and a secondary switch that is operated dually to the primary one. For details the reader is referred to the standard power electronics literature (e.g. [61]).

The switches are driven by a pulse sequence of constant frequency (period), the switching frequency f_s (switching period T_s), which characterizes the operation of the converter. The dc component of the output voltage can be regulated through the duty cycle d, which is defined by $d = t_{on}/T_s$, where t_{on} represents the interval within the switching period during which the primary switch is in conduction. Therefore, the main control objective for dc–dc converters is to drive the primary switch with a duty cycle such that the dc component of the output voltage is equal to its reference. This regulation needs to be maintained despite variations in the load or the input voltage.

The difficulties in controlling dc–dc converters arise from their hybrid nature. In general, these converters feature three different modes of operation, where each mode is associated with a (different) linear continuous-time dynamic law. Furthermore, constraints are present resulting from the converter topology. In particular, the manipulated variable (duty cycle) is bounded between zero and one, and in the discontinuous current mode a state (inductor current) is constrained to be non-negative. Additional constraints are imposed as safety measures, such as current limiting and soft-starting, where the latter constitutes a constraint on the maximal derivative of the current during start-up. The control problem is further complicated by gross changes in the operating point owing to input voltage and output load variations, and model uncertainties.

Motivated by the hybrid nature of dc–dc converters, we have presented in Ref. [68] a novel approach to the modeling and controller design problem for fixed-frequency dc–dc converters, using a synchronous step-down dc–dc converter as an illustrative example (see Fig. 4). In particular, the notion of the ν-resolution model was introduced to capture the hybrid nature of the converter, which led to a PWA model that is valid for the whole operating regime and captures the evolution of the state variables within the switching period.

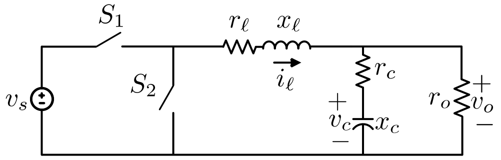

Fig. 4. Topology of the step-down synchronous converter.

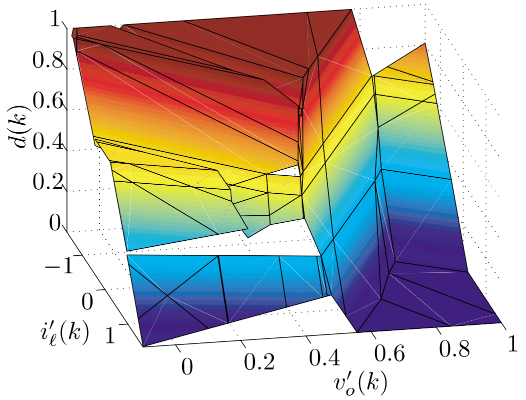

Fig. 5. State-feedback control law: the duty cycle $d(k)$ is given as a PWA function of the transformed state vector; dark blue corresponds to $d(k) = 0$ and dark red to $d(k) = 1$.

Based on the converter's hybrid model, we formulated and solved an MPC problem, with the control objective to regulate the output voltage to its reference, minimize changes in the duty cycle (to avoid limit cycles at steady state) while respecting the safety constraint (on the inductor current) and the physical constraint on the duty cycle (which is bounded by zero and one). This allows for a systematic controller design that achieves the objective of regulating the output voltage to the reference despiteinput voltage and output load variations while satisfying the constraints. In particular, the control performance does not degrade for changing operating points. Furthermore, we derived off-line the explicit PWA state-feedback control law with 121 polyhedra. This controller can be easily stored in a look-up table and used for the practical implementation of the proposed control scheme. The derived controller, for the set of converter and control problem parameters considered in Ref. [68], is shown in Fig. 5, where one can observe the control input $d(k)$ as a PWA function of the transformed states i'_ℓ (inductor current) and v'_o (output voltage).

The transformed states correspond to a normalization of the actual measured states over the input voltage. This allows us to account for changes in

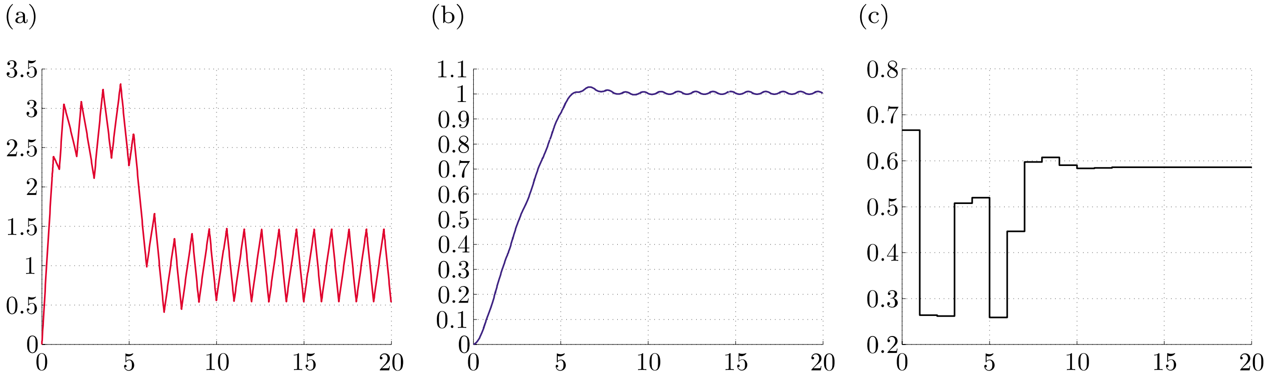

Fig. 6. Closed-loop response of the converter during start-up in nominal operation. All units are normalized, including the time scale where one time unit is equal to one switching period. (a) Inductor current $i_\ell(t)$. (b) Output voltage $v_o(t)$. (c) Duty cycle $d(t)$.

the input voltage that are an important aspect of the control problem. Moreover, the output load may change drastically (basically in the whole range from open- to short-circuit). This is addressed by adding an additional parameter to the control problem formulation and a Kalman filter is used to adjust it. For more details on these considerations and the reasoning behind the use of the output voltage as a state (rather than the capacitor voltage), the reader is referred to Ref. [35].

Regarding the performance of the closed-loop system, the simulation results in Fig. 6 show the step response of the converter in nominal operation during start-up. The output voltage reaches its steady state within 10 switching periods with an overshoot that does not exceed 3%. The constraint imposed on the current, the current limit, is respected by the peaks of the inductor current during start-up, and the small deviations observed are due to the approximation error introduced by the coarse resolution chosen for the ν-resolution model. The same holds for the small – in the range of 0.5% – steady-state error that is present in the output voltage.

Moreover, an *a posteriori* analysis shows that the considered state space is a positively invariant set under the derived optimal state-feedback controller. Most importantly, a piecewise quadratic Lyapunov function can be computed that proves exponential stability of the closed-loop system for the whole range of operating points.

4. Hybrid Control for the Design of Industrial Controllers

4.1. Hybrid Control Issues in Industrial Processes

Although continuous or quasi-continuous sampled data control has been the main topic of control education and research for decades, in industrial practice discrete-event or logic control is at least as important for the correct and efficient functioning of production processes than continuous control. A badly chosen or ill-tuned continuous controller only leads to a degradation of performance and quality as long as the loop remains stable, but a wrong discrete input (e.g., switching on a motor that drives a mass against a hard constraint or opening a valve at the wrong time) will most probably cause severe damage to the production equipment or even to the people on the shop floor, and to the environment. In addition, discrete and logic functions constitute the dominant part of the control software and are responsible for most of the effort spent on the engineering of control systems of industrial processes.

Generally, several layers of industrial control systems can be distinguished. The first and lowest layer of the hierarchy realizes safety and protection related discrete controls. This layer is responsible for the prevention of damage to the production equipment, the people working at the production site, and the environment and the population outside the plant. For example, a robot is shut down if someone enters its workspace or the fuel flow to a burner is switched off if no flame is detected within a short period after its start. Most of the safety-related control logic is consciously kept simple in order to enable inspection and testing of the correct function of the interlocks. This has the drawback that a part of the plant may be shut down if one or two of the sensors associated with the interlock system indicate a potentially critical situation while a consideration of the information provided by a larger set of sensors would have led to the conclusion that there was in fact no critical situation. As shutdowns cause significant losses of production, there is a tendency to install more sophisticated interlock systems which can no longer be verified by simply looking at the code or performing simple tests.

In the sequel, we do not distinguish between strictly safety-related and emergency-shutdown systems (which have to be presented to and checked by the authorities outside the plant) and more general protection systems which prevent damage or degradation of the equipment or unwanted situations causing large additional costs or the loss of valuable products, since from a design and verification point of view, there is no difference between the two. Clearly, the correct function of safety and protection related controls depends on the interaction of the discrete controller with the continuous and possibly complex plant dynamics.

The second layer of the control system is constituted by continuous regulation loops, e.g., for temperatures, pressures and speeds of drives. These loops receive their set-points or trajectories from the third layer, which is responsible for the sequence of operations required to process a part or a batch of material. On this layer, mostly discrete switchings between different modes of operation are controlled, but also continuous variables may be computed and passed to the lower-level continuous control loops. If these sequences are performed repeatedly in the same manner, they are usually realized by computer control. However, sequence control is mostly performed by the operators if large variations of the sequence of operations exist, as in some chemical or biochemical batch processes. The same is true for the start-up of production processes or for large transitions between operating regimes which usually do not occur very frequently.

On a fourth layer of the control hierarchy, the various production units are coordinated and scheduled to optimize the material flow. A major part of the control code (or of the task of the operators) on the sequential control layer is the handling of exceptions from the expected evolution of the production process: drills break, parts are not grasped correctly, controlled or supervised variables do not converge to their set-points, valves do not open or close, etc. Although there usually is only one correct sequence, a possibly different recovery sequence must be implemented for each possible fault. Exception handling in fact also is responsible for a large fraction of the code in continuous controllers (plausibility checks of sensor readings, strategies for the replacement of suspicious values, actuator monitoring, etc.).

Safety- and protection-related discrete controls and sequential discrete or mixed continuous-discrete controls are of key importance for the safe and profitable operation of present-day production processes. Their correctness and efficiency cannot be assessed by testing the logic independently as they are determined by their interaction with the (mostly) continuous dynamics of the physical system. This calls for systematic, model-based design and verification procedures that take the hybrid nature of the problem into account. In practice, however, discrete control logic is usually developed at best in a semi-formal manner. Starting from partial and partly vague specifications, code is developed, modified after discussions with the plant experts, simulated using a very crude plant model or with the programmer acting as the plant model, and then tested, debugged and modified during start-up of the plant. The main reason that this approach does not lead to complete failure is that for the most part logic control software from other projects is re-used and only small modifications and extensions are added. However, taking into account the low-level programming languages used and the lack of formal documentation, such software systems may become harder and harder to maintain.

4.2. Verification of Safety-Related Logic Controllers

In order to be accepted by practitioners, verification procedures for safety- and protection-related industrial controllers must be able to handle the control logic as it is implemented on the control hardware, usually a programmable logic controller (PLC) or a distributed control system. For the implementation of logic controls, the standard IEC-61131-3 [47] defines several standard formats. Among these, sequential function charts (SFCs) are best suited to represent sequential behaviors and the parallel (simultaneous) or alternative execution of program steps, and to structure logic control programs. Control code written in other IEC-61131-3 languages (Ladder Diagrams, Instruction List, Structured Text or Function Block Diagrams) can be embedded in SFC. According to Ref. [47], SFC consist of alternating sequences of steps and transitions, where actions are associated with steps and conditions with transitions. For an example, Fig. 7 shows the graphical representation of SFC, in which rectangles denote the steps (with actions blocks attached to the right) bold horizontal lines the transitions (including conditions) and vertical lines the flow of execution (from top to bottom). Action blocks contain a list of actions that are either simple manipulations of logical variables (most importantly the outputs to the plant), or activities that are limited to a specified period of time (or start after a given delay), or the activation of other SFC. The transition conditions may involve Boolean expressions of sensor readings and internal program variables.

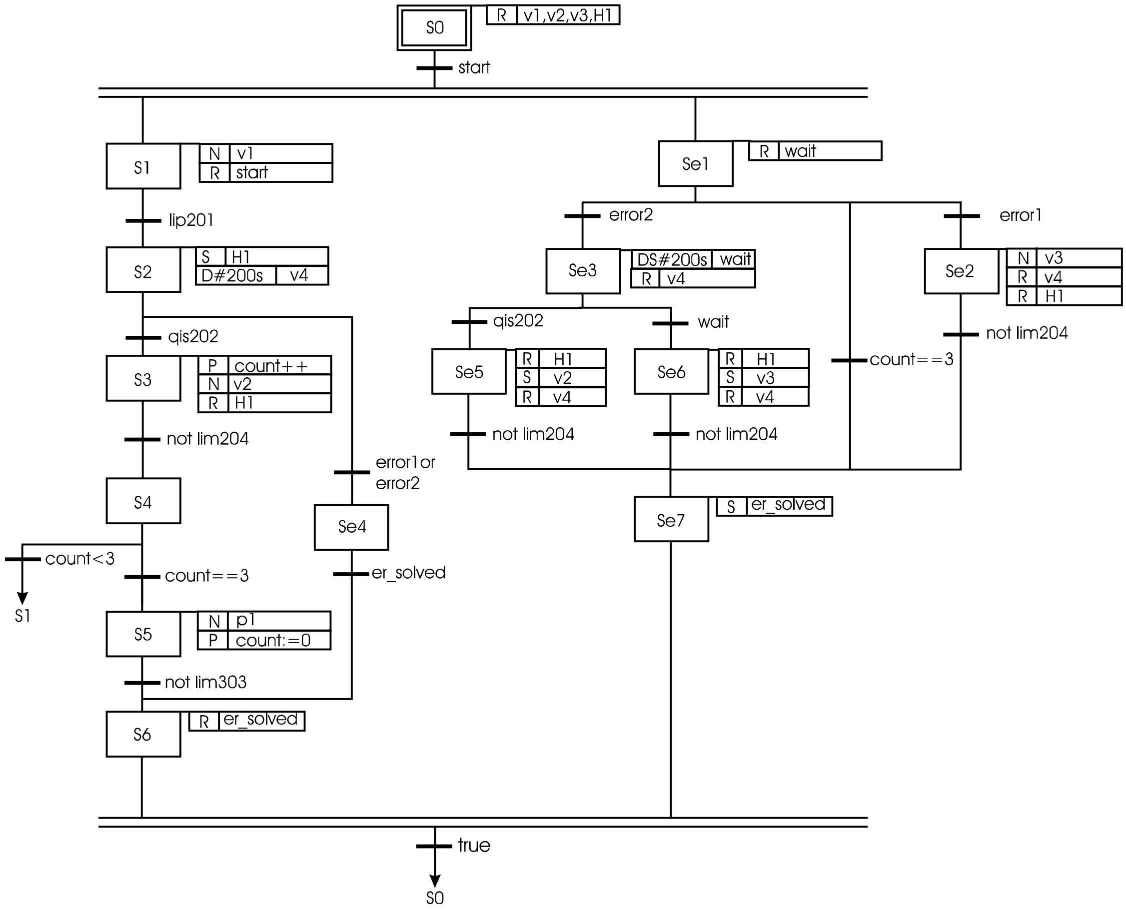

Fig. 7. Supervisory controller as SFC.

The goal of the verification for this type of logic controllers is to guarantee that the controller prevents the plant from reaching unwanted or dangerous states and/or ultimately steers it to the desired terminal state. Therefore, the plant dynamics must be described formally by a (untimed, timed or hybrid) automaton model, and a formal specification must be provided in a temporal logic framework (for example, see, [27]). Before model checking can be applied, the control logic (e.g., an SFC) must be represented as a state transition system. For logic control programs that contain timers or delayed actions, TA are the most suitable format. After composition of the plant model and the controller model, the overall model can be checked against the formal specification using one of the available tools, e.g., SMV for purely discrete models, UPPAAL for TA models or the tools sketched in Ref. [74] for hybrid models. The scheme of the overall procedure is shown in Fig. 8. In the sequel, we discuss the steps of the procedure in more detail for a specific approach that implements this general idea.

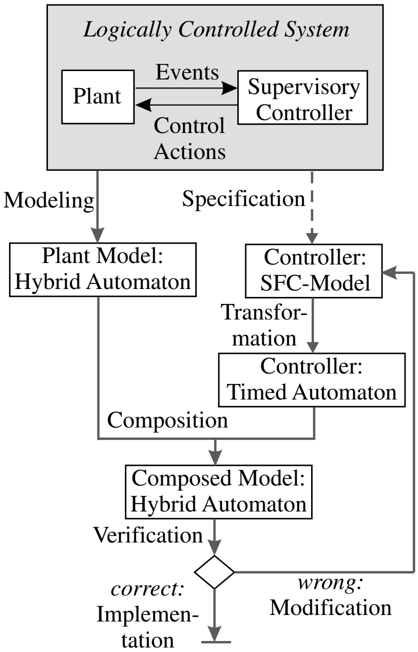

Fig. 8. Control design scheme.

4.2.1. Transformation of SFC into TA

As proposed in Refs [13,31], the transformation of a controller given as SFC into a set of TA can be accomplished by a procedure that first uses a graph grammar to partition the SFC into syntactical units. Such a unit is either a sequence of steps and transitions including alternative branches or a block representing parallel branches of the SFC. By scanning the SFC controller in a top-down manner, a structure of these two types of units is obtained such that a modular timed automaton model can be generated in a straightforward manner: each of the units is mapped into a single timed automaton, and the activation of the automata according to the execution of the SFC is established by synchronization labels. The state-transition structure of the automata follows directly from the step-transition sequences of the SFC. The transition conditions, which involve either inputs from the plant or internal variables of the SFC, are expressed by synchronization labels as well. Finally, the actions associated with the steps are modeled by separate automata, which can include clocks for the case of time-dependent action qualifiers. For modeling the actions, the procedure proposed in Ref. [13] uses a scheme that explicitly accounts for the cyclic scanning mode in which SFCs are executed on PLCs.

4.2.2. Model Composition and Verification

If the verification aims at analyzing that the controller drives the plant into particular sets of continuous states (or just prevents the plant from reaching them), representing the plant behavior by hybrid dynamic models, such as hybrid automata [44], is an appropriate choice. The communication between the controller and the plant model can be realized by synchronization of transitions, or by shared variables between both models. If the verification is carried out by the approach of abstraction-based and counterexample-guided model checking (see Ref. [26], and Ref. [74] for an overview of alternative techniques), the modular model is next transformed into a single composed hybrid automaton. The principle of abstraction-based and counterexample-guided model checking for verifying safety properties can be summarized as follows: an initial abstract model, given as a finite automaton, follows from abstracting away the continuous dynamics of the composed hybrid automaton. Applying model checking to the abstract model identifies behaviors (the *counterexamples*) for which the safety property is violated. In a validation step, it is analyzed whether for these particular behaviors counterexamples exist also for the hybrid automaton. If this applies, the procedure terminates with the result that the hybrid automaton does not fulfill the safety requirement. If none of the counterexamples for the abstract model can be validated for the hybrid automaton, the safety of the latter is proved. The validation step involves the evaluation of the continuous dynamics of the hybrid automaton, i.e., sets of reachable hybrid states are determined for locations encountered along the potential counterexample. Each time a counterexample of the abstract model is invalidated, the information about enabled or disabled transitions (according to the reachable hybrid states in the respective locations) is used to refine the abstract model.

If the verification reveals that the composed hybrid automaton satisfies all relevant requirements, the original SFC-model of the controller represents an implementable supervisory controller. Otherwise, the counterexample corresponding to the requirement violation must be examined in order to identify in which respect the SFC controller has to be modified.

4.2.3. Application to an Evaporation System

In order to illustrate the verification procedure, it is applied to the case study of a batch evaporation system [31,49]. As shown in Fig. 9, the system consists of two tanks (T1, T2) with heating devices, a condenser with cooling (C1), connecting pipes with valves (V1, V2, V3) and a pump (P1), as well as different sensors for liquid levels (LIS), temperatures (TI) and concentration (QIS). The intended operation is to evaporate the liquid from a mixture in T1 until a desired concentration is reached, to collect three batches of the product in T2, and to empty the latter afterwards through P1. Figure 7 shows a possible SFC-controller that not only realizes the desired procedure (left branch) but also includes exception routines (right branch) for the cases of evaporator breakdown (error1) and malfunction of the heating device of T1 (error2).

Since the SFC-controller contains two time-dependent actions (marked by ‘D#200s’ and ‘DS#200s’), it is transformed into a set of TA following the procedure sketched above. One possible verification objective is to check whether the controller avoids safety-critical states, which are a critically high and a critically low temperature of the mixture in T1, for the two failure cases. Assuming that a condenser malfunction occurs while the evaporation in T1 runs and T2 is partly filled, the relevant plant behavior can be restricted to three phases: P1, heating in T1 while T2 is drained; P2, draining of T2 without heating in T1; and P3, transferring the content of

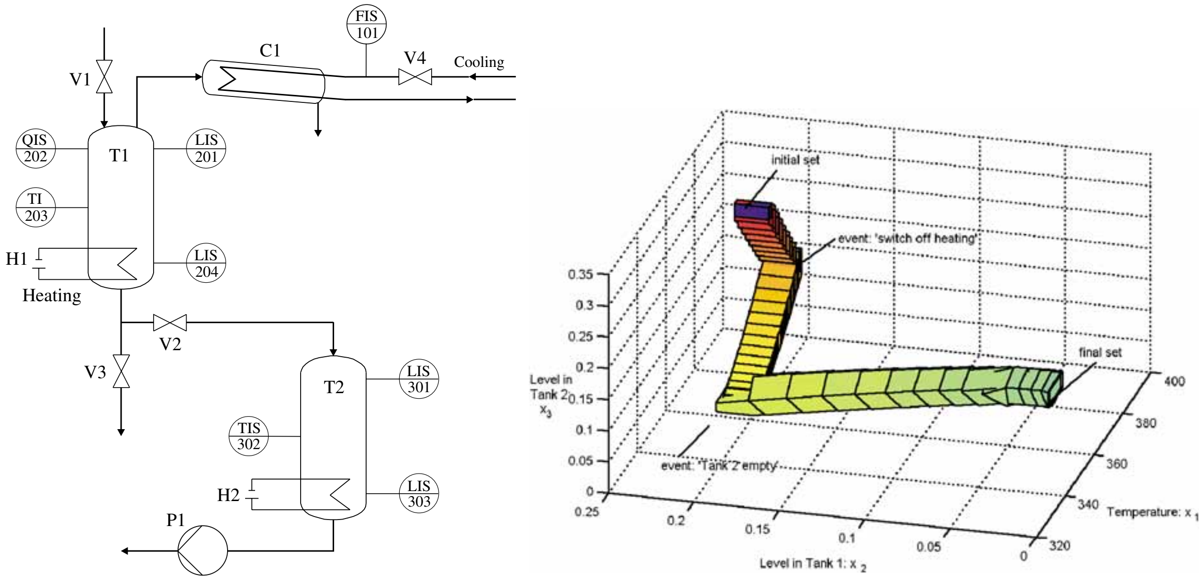

Fig. 9. Left part: Flowchart of the evaporation system. Right part: Reachable continuous set [the final set shows that a critically low temperature ($x_1 = 338\,\mathrm{K}$) is not reached before Tank 1 is empty ($x_2 \leq 0.01$ m)].

T1 into T2. The corresponding hybrid automaton contains nonlinear differential equations for the temperature of the liquid in T1, as well as the liquid levels in T1 and T2. The verification procedure described above was applied to the composition of all automata. As the set of reachable continuous states in Fig. 9 shows, a critically low temperature of 338 K is not reached before T1 is emptied, i.e., it can be concluded that the SFC-controller works as desired for this configuration.

4.3. Optimal Start-up and Shut-down of Industrial Plants

Although most processing systems are operated by a combination of continuous and discrete controls, both types of controllers are usually designed separately; however, operations such as start-up, shutdown or product change-over require the simultaneous consideration of both types of controls to avoid opposing effects. This section addresses the task of designing continuous and discrete controls in an integrated fashion by formulating an optimization problem for hybrid automata.

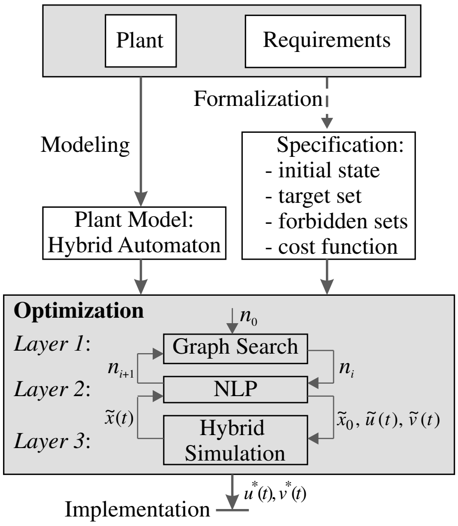

Fig. 10. Scheme of the optimization approach.

Different approaches to the optimization of hybrid systems have been published in recent years, ranging from rather generic formulations to specific methods for certain subtypes of hybrid systems, for example, see Refs [23,24,38,67,73]. One branch of methods follows the idea of transforming the hybrid dynamics into a set of algebraic (in-)equalities that serve as constraints for a mixed-integer program [16,79]. As an alternative, the following section sketches a method that combines graph search with embedded nonlinear programming (NLP) and hybrid simulation [77,78].

4.3.1. Graph Search with Embedded NLP

Figure 10 provides an overview of the method: the starting point are the given plant dynamics and an informal listing of the requirements for the controlled behavior of the plant. The dynamics is represented by a deterministic hybrid automaton as introduced in Ref. [77], i.e., characterized by continuous and

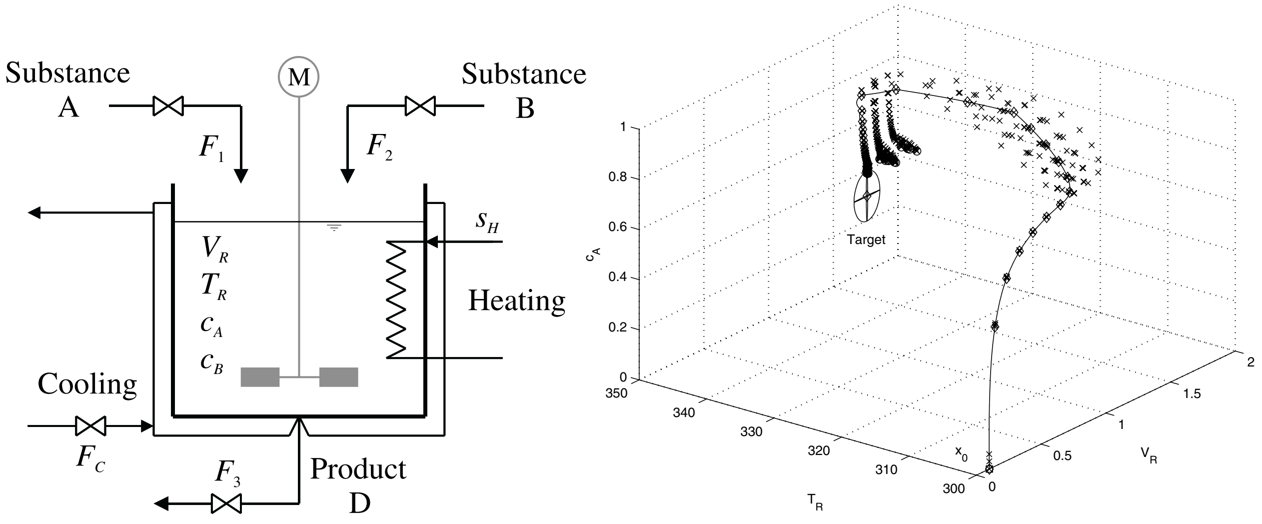

Fig. 11. Left: Scheme of the CSTR. Right: The optimal x-trajectory (solid line) projected in the (V_R, T_R, c_A)-space. Explored nodes are marked by crosses.

discrete input variables, autonomous switching between different continuous models and possible resets associated with transitions. The requirements are formalized by specifying the initialization of the hybrid model, a set of hybrid target states (in which the plant has to be driven by the controller) a set of hybrid forbidden states (that must never be encountered) and a cost criterion Ω. The latter specifies a performance measure, such as the startup time or the resource consumption during startup, which has to be minimized. Given the hybrid automaton and the specification, the following optimal control problem is posed:

$$\min_{\phi_u \in \Phi_u, \phi_v \in \Phi_v} \Omega(t_f, \phi_\sigma, \phi_u, \phi_v), \tag{1}$$

where ϕ_u and ϕ_v are the continuous and discrete input trajectories. $\phi_\sigma = (\sigma_0, \ldots, \sigma_f)$ is a feasible trajectory of hybrid states σ of the hybrid automaton (see Ref. [77] for more details). The solution of the optimization problem returns the input trajectories ϕ_u^*, ϕ_v^* that lead to a feasible run ϕ_σ^* which minimizes Ω.

The key idea of the optimization approach is to separate the optimization of the continuous and of the discrete degrees of freedom in the following sense: the discrete choices (i.e., the input trajectories ϕ_v) are determined by a graph search algorithm resembling the well-known principle of shortest-path search. For each node n_i contained in the search graph, an embedded optimization for the continuous degrees of freedom (and optionally for relaxed discrete degrees of freedom for future steps) is carried out. Within this embedded NLP, numerical simulation is employed to evaluate the hybrid dynamics of the hybrid automaton, leading to a cost evaluation for the corresponding evolution of the system. These costs are used in the graph search to apply a branch-and-bound strategy, i.e., upper (and lower) bounds on the optimal costs for the transition procedure are iteratively computed to prune branches of the search tree as early as possible.

4.3.2. Application to a Chemical Reactor

The method is illustrated by using the example of the start-up of a continuous stirred tank reactor (CSTR), as described in Ref. [79]. The system consists of a tank equipped with two inlets, a heating coil, a cooling jacket, a stirrer and one outlet (see Fig. 11). The inlets feed the reactor with two dissolved substances A and B which react exothermically to form a product D. The inlet flows F_1 and F_2 (with temperatures T_1 and T_2) can be switched discretely between two values each. The outlet flow F_3 is controlled continuously. In order to heat up the reaction mixture to a nominal temperature range with a high reaction rate, the heating can be switched on (denoted by a discrete variable $s_H \in \{0, 1\}$). By a continuously controlled cooling flow F_C an excess of heat can be removed from the tank once the reaction has started. The control objective for this system is to determine the input trajectories that drive the initially empty reactor into a desired range, in which the liquid volume V_R, the temperature T_R and the concentrations c_A and c_B have values in given intervals. Additionally, the regions of the state space where $T_R \geq 360$ or $V_R \geq 1.6$ are forbidden.

To model the system, the state vector is defined as $x := (V_R, T_R, c_A, c_B)^T$, the continuous input vector as $u := (F_3, F_C)^T$, and the discrete input vector as $v := (F_1, F_2, s_H)^T$. Depending on the continuous state, the system dynamics can be written as $\dot{x} = f(z, x, u, v)$ where:

- for z_1 with $V_R \in [0.1, 0.8]$:

$$f^I = \begin{pmatrix} F_1 + F_2 - F_3 \\ (F_1(T_1 - T_R) + F_2(T_2 - T_R))/V_R \\ \quad + F_C k_1 (T_C - T_R)(k_2/V_R + k_3) - k_4 q \\ (F_1 c_{A,1} - c_A(F_1 + F_2))/V_R + k_9 q \\ (F_2 c_{B,2} - c_B(F_1 + F_2))/V_R + k_{10} q \end{pmatrix}$$

- for z_2 with $V_R \in [0.8, 2.2]$:

$$f^{II} = \left(f_1^I, f_2^I + s_H k_6 (T_H - T_R)\left(k_7 - \frac{k_8}{V_R}\right), f_3^I, f_4^I\right)^T,$$

and $q = c_A c_B^2 \exp(-k_5/T_R)$. The separation into two V_R-regions accounts for the fact that the heating is only effective above $V_R = 0.8$. The initial state is $x_0 = (0.1, 300, 0, 0)^T$ and the target is given by z_2 and a hyper-ball with radius 0.1 around the continuous state $x_{tar} = (1.5, 345, 0.4, 0.2)^T$. The optimization was run with the cost criterion that the transition time for the startup procedure is minimized. The strategy chosen is that depth-first search is used until a first solution is found, then a breadth-first strategy is applied. Figure 11 shows the state trajectory representing the best solution obtained for a search comprising 400 nodes (obtained within 2 min of computation on a standard PC).

5. Hybrid Systems in Automotive Electronics Design

The design of embedded control systems for automotive applications has become very challenging in the last 5 years. Owing to the lack of an overall understanding of the interplay of subsystems and of the difficulties encountered in integrating very complex parts, system integration is a nightmare. Jurgen Hubbert, in charge of the Mercedes-Benz passenger car division, publicly stated in 2003: "The industry is fighting to solve problems that are coming from electronics and companies that introduce new technologies face additional risks. We have experienced blackouts on our cockpit management and navigation command system and there have been problems with telephone connections and seat heating". We believe that this state is the rule, not the exception, for the leading original equipment manufacturers (OEMs) in the environment today. The source of these problems is not only the increased complexity but also the difficulty of the OEMs in managing the integration and maintenance process with subsystems that come from different suppliers who use different design methods, different software architecture, different hardware platforms, different (and often proprietary) real-time operating systems. Therefore, the need for standards in the software and hardware domains that will allow plug-and-play of subsystems and their implementation are essential while the competitive advantage of an OEM will increasingly reside on essential functionalities (e.g., stability control).

Hence, to deliver more performing, less expensive and safer cars with increasingly tighter time-to-market constraints imposed by worldwide competitiveness, the future development process for automotive electronic systems must provide solutions to:

- the design of complex functionalities with tight requirements on safety and correctness;
- the design of distributed architectures consisting of several subsystems with constraints on non-functional metrics, such as cost, power consumption, weight, position and reliability;
- the mapping of the functionalities onto the components of a distributed architecture with tight real-time and communication constraints.

Most of the car manufacturers outsource the design and production of embedded controllers to suppliers (so-called Tier-1 companies), which in turn buy IC components and other devices by third parties (so-called Tier-2 companies). Embedded controllers are often developed by different Tier-1 companies and are requested to operate in coordination on a same model of a car. Moreover, in the development of an embedded controller, the supplier has to integrate some IPs (intellectual properties) provided by the car manufacturer at different levels of details (algorithms, legacy code) and, in the near future, possibly by third parties.

To cope with this challenging context, the design flow has to be significantly improved. Hybrid systems techniques can have an important role in this respect. Successful approaches to the design of control algorithms using hybrid system methodologies had been presented in the literature, e.g., cut-off control [11], intake throttle valve control [12], actual engaged gear identification [9] and adaptive cruise control [60]. However, despite the significant advances of the past few years, hybrid system methodologies are not mature yet for an effective introduction in the automotive industry. On the other hand, hybrid system techniques may have an important impact on several critical open problems in the overall design flow, beyond the classical controller synthesis step. In this section, we analyze the design flow for embedded controllers in the automotive industry, with the purpose of identifying challenges and opportunities for hybrid system technologies. In particular, in Section 5.1, an overview of the typical design flow for embedded controllers adopted by the automotive industry is presented with particular emphasis on the Tier-1 supplier problems. In Section 5.2, for each design step, we identify critical phases and bottleneck problems and we extract relevant open problems that hybrid system technologies may contribute to solve.

5.1. Design Scenario and Design Flow

In today cars, the electronic control system is a networked system with a dedicated electronic control unit (ECU) for each subsystem; for example, engine control unit, gear-box controller, ABS (anti-lock braking system), dashboard controller and VDC (vehicle dynamic control). The ECUs interact by asynchronous communication over a communication network specifically designed for automotive applications, such as CAN. Each ECU is a multirate control system composed of nested control loops, with frequency and phase drifts between fixed sampling-time actions and event driven actions.

The standard design flow of automotive ECUs adopted by Tier-1 companies (subsystem suppliers) is represented by the so-called *V-diagram* shown in Fig. 12. The top-down left branch represents the synthesis flow. The bottom-up right branch is the integration and validation flow. The synthesis flow, which will be analyzed in details in the next section, is articulated in the following steps:

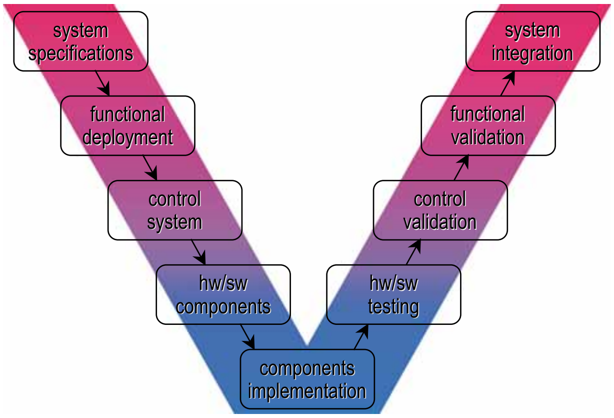

Fig. 12. Design and integration flow.

(A) *System specification*: formalization of system specification; coherence/feasibility analysis; completion of under-specified behaviors; abstraction of lower layers customer requirements; and risk assessment.
(B) *Functional deployment*: system decomposition; definition of subsystem specifications; design of control algorithm architecture; and definition of specifications for each control algorithm.
(C) *Control system*: plant modeling (model development, identification, validation); controller synthesis (plant model and specifications analysis, algorithm development, validation); and fast prototyping.
(D) *HW/SW components*: formal specifications for implementation; design of hardware and software architectures; hardware design; software development and automatic code generation; and RTOS.[2]

The synthesis flow terminates with the development of the components.

The design of automotive ECUs is subject to very critical constraints on cost and time-to-market. Successful designs, in which costly and time-consuming re-design cycles are avoided, can only be achieved using efficient design methodologies that allow for component reuse at all layers of the design flow (see Ref. [3,10]) and for the evaluation of platform requirements at the early stages of the design flow. A *derivative design* approach is commonly implemented to minimize development time and cost by maximizing reuse (e.g., see Ref. [56]). According to this approach, control algorithms as well as electrical and mechanical components are obtained by derivation from available product generations that contain solutions developed in the past. *Model-based design* is widely adopted in the automotive industry. The description of specifications, functional architectures, algorithms and implementation architectures by models that are shared along the design chain allows – at least in principle – formal representation, analysis and full validation of the control system as well as automatic code generation. However, it is apparent that the today design chain should be refined to achieve higher degrees of integration and standardization.

---

[2] This layer is only sketched, since of little relevance to hybrid systems applications.

5.2. Synthesis Flow

In this section, we describe the synthesis part of the automotive design flow, emphasizing the aspects where we believe hybrid system techniques may have an important impact.

5.2.1. System Specifications

System specifications define requirements on performance, driveability, fuel consumption, emissions and safety. They are given in terms of a number of operation modes characterized by different controlled variables and objectives and regard both discrete and continuous behaviors. The degree of detail given by the OEMs in describing system specifications is not uniform. Some behaviors may result only vaguely specified while some others may be very detailed. Coherence and feasibilty of system specifications have to be analyzed to avoid redesign cycles. Hybrid systems can efficiently support system specification with:

- hybrid techniques and tools for model-based formal description and validation of system specifications integrated with requirements management, and analysis techniques for risk assessment;
- abstraction techniques based on hybrid models for projecting lower-layers customer requirements to the upper layers, in order to achieve a complete system representation at the specification layer.

5.2.2. Functional Deployment

In a first stage of the design, the system is decomposed into a collection of interacting components. This decomposition is clearly a key step towards a good quality design, since it leads to a design process that can be carried out as independently as possible for each component (see Ref. [3] for more details). A typical decomposition for engine control is shown in Fig. 13. The objectives and constraints that define the system specification are distributed among the components by the functional deployment process so that the composition of the behaviors of the components is guaranteed to meet the constraints and the objectives required for the overall controlled system. In a second stage of the functional deployment, for each function the architecture of control algorithms is defined, which includes their interconnection topology. Furthermore, for each control algorithm, desired closed-loop specifications are defined to achieve the requested behavior for each functional component. This process is mainly guided by the experience of system engineers, with little support of methodologies and tools. The sets of measurable and actuated quantities, which will constitute the sets of, respectively, inputs and outputs to the ECU, are often defined by the OEM that also specifies the sensors and the actuators to be used, since they have a major impact on the cost of the control system. In addition, OEM requirements may include details on the topology of the control algorithms architecture that further constrains the functional deployment process. Hybrid formalisms could support the description of

- the functional decomposition and the desired behavior for each functional component;
- the architecture of control algorithms, sensors and actuators, for each functional component;
- the desired requirements for each control algorithm that result from the functional deployment.

Hybrid techniques can also profitably be applied to develop methodologies and tools for the synthesis of functional behaviors from system specifications and for the validation of the specifications of the control algorithms with respect to the desired functional behaviors.

5.2.3. Control System

At the control system layer, the algorithms to be implemented in the architecture defined at the

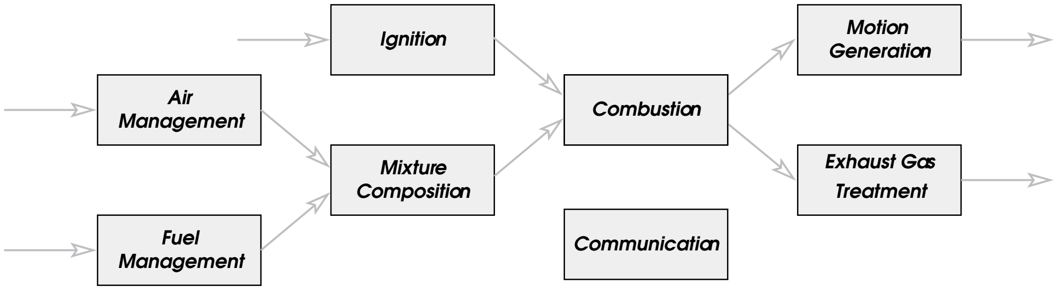

Fig. 13. Functional decomposition.

functional layer are designed. All control algorithms have to meet the assigned specification, so that their composition within a functional component exhibits the required behavior defined during functional deployment. Below, we briefly discuss the activities performed in the control system layer. It is worthwhile to note that in derivative design, when the algorithms are reused from previous designs, some activities may be either partially performed or skipped.

Plant modeling – Model development. Traditionally, control engineers adopt mean-value models to represent the behavior of automotive subsystems. However, the need for hybrid system formalisms to model the behavior of systems in automotive applications is apparent in many cases. Consider for instance a 4-stroke internal combustion engine. An accurate model of the engine has a natural hybrid representation because the cylinders have four modes of operation corresponding to the stroke they are in, while powertrain and air dynamics are continuous-time processes. In addition, these processes interact tightly. In fact, the timing of the transitions between two phases of the cylinders is determined by the continuous motion of the powertrain, which, in turn, depends on the torque produced by each piston. A detailed hybrid model of the engine has been presented in Ref. [7]. A second interesting example is the hybrid modeling of the driveline. In Ref. [8], a detailed model with up to 6048 discrete state combinations and 12 continuous state variables was presented. The hybrid model accurately represents discontinuities distributed along the driveline owing to engine suspension, clutch, gear, elastic torsional characteristic, tires, frictions and backlashes. The models of automotive subsystems are often highly nonlinear. In engine modeling, nonlinearities arise from fluid-dynamics and thermodynamics phenomena (e.g., volumetric efficiency, engine torque and emissions) and are usually represented by PWA maps. In conclusion, plant modeling in automotive requires extensive use of hybrid techniques, in both deterministic and stochastic frameworks, which include FSM, DES, DT CT and PDA, for representing interacting behaviors of different nature.

Plant modeling – Identification. In current practice, parameter identification is mostly based on steady-state measurements, obtained using either manually defined set-points or automatic on-line screening. Dynamic parameters are often obtained either analytically or from step responses. However, step response and other classical identification methods can be used to identify models representing standard continuous evolutions only, such as those exhibited by mean-value models. When applied to hybrid models, classical techniques can only be used to identify the plant model separately in each discrete mode. They hardly succeed in identifying parameters related to switching conditions and cannot be applied to black-box hybrid model identification. The availability of hybrid system identification techniques using transient data, including mode switching, would allow to increase identification accuracy, reduce the amount of experimental data needed and identify all parameters in the hybrid models. Efficient identification techniques for hybrid systems will also give the opportunity for modeling more complex hybrid behaviors that are currently abstracted owing to the difficulties in the identification process. The representation and identification of nonlinearities, as either PWA or polynomial functions (see Ref. [15]) could be significantly improved using efficient hybrid techniques to optimize the domain partition (possibly not grid-based) reducing model complexity; improve parameter identification accuracy; allow identification of high dimension nonlinearities $R^p \rightarrow R$, with $p > 2$.

Plant modeling – Validation. The selection of test patterns for model validation is a crucial issue in the validation process. Classical techniques allow to assess the richness of sets of test patterns for the validation of continuous models [75]. These techniques can be used in automotive applications to assess richness of validation patterns for continuous evolutions of the plant. Validation of hybrid models is a very complex task not sufficiently investigated in the literature. There is the need for methodologies for the assessment of the coverage of validation patterns and their automatic generation for hybrid models. Such problems can be formalized in the framework of reachability analysis. Interesting approaches have been proposed using the notions of structural coverage and data coverage.

Controller synthesis – Plant model and specifications analysis. Typically, before proceeding to the actual design of a control algorithm for a new application, some experimental data on a prototype of the system to be controlled are obtained using either open-loop control or some very elementary closed-loop algorithm. In addition, the assessment of classical structural properties, such as reachability, observability, stabilizability and passivity [14], on the plant model is of interest in this phase. Stability and robust stability margins, most critical perturbations and uncertainties, reachability and observability measures in the state space are important characteristics to be evaluated. Unfortunately hybrid system theory for system analysis is not fully developed. Some fundamental properties have not been formally defined yet and tests are not available for verifying most of the properties. Efficient implementation of the tests will be necessary for automatic evaluation on hybrid plant models, since often manual testing is prohibitively complex.

Controller synthesis – Algorithm development. Control algorithms are often characterized by many operation modes that are conceived to cover the entire life-time of the product: starting from in-factory operations before car installation, configuration, first power-on, power-on, functioning, power-off, connection to diagnostic tools. During standard functioning, control strategies can be either in a nominal operation mode or in some recovery mode. A significant number of algorithms are dedicated to the computation of switching conditions and controller initializations. Diagnostic algorithms, often required by the legislation, represent a major part of the strategies implemented in automotive ECUs.

As observed above, both specifications and accurate models of the plant are often hybrid. However, the methodology currently adopted in the automotive industry for algorithm development is rather crude in this respect. The continuous functionalities to be implemented in the controller are designed based on mean-value models of the plant, with some *ad hoc* solutions to manage hybrid system issues (such as synchronization with event-based behaviors). If the resulting behavior is not satisfactory under some specific conditions, then the algorithm is modified to detect critical behaviors and operate consequently (introducing further control switching). The discrete functionalities of the controller are designed by direct implementation of non-formalized specifications. Design methodologies for the synthesis of discrete systems, such as those developed for hardware design, are not employed. If the algorithm is not designed from scratch, but is obtained by elaborating existing solutions, as is often the case, then additional operation modes may be introduced to comply with the new specification. This results in a non-optimized controller structure. Structured approaches to the integrated design of the controller that allow to satisfy hybrid specifications considering hybrid models of the plant are not adopted as yet even though they have obvious advantages over the heuristics that permeate the present approaches.

Hybrid system techniques can significantly contribute to the improvement of control algorithm design in automotive applications. The introduction of hybrid synthesis techniques should be aimed at shortening the algorithm development time; reducing testing effort; reducing calibration parameters and provide automatic calibration techniques; improving closed-loop performances; guaranteeing correct closed-loop behavior and reliability; achieving and guaranteeing desired robustness; and reducing implementation cost. However, most of the analytical approaches so far proposed for controller design using hybrid system techniques are quite complex, require highly trained designers and long development time. For a profitable introduction of hybrid system design techniques, it is essential the methodologies to be supported by efficient tools that allow fast and easy designs. Hybrid MPC is a good example of the development of the methodology supported by successful efforts in design tool development [17].

Controller synthesis – Validation. Owing to complexity of the plant–controller interactions, the non-negligible effects of the implementation, the large uncertainties in the plant given by production diversity and aging, control algorithms validation is one of the hottest topics in automotive industry. Usually validation is obtained by expensive experimental results and extensive and time-consuming simulations of the closed-loop models. The designers, based on their experience, devise critical trajectories to test the behavior of the closed-loop system in the perceived worst-case conditions. A rough investigation on the robustness properties of control algorithms is obtained by screening the most critical parameters and uncertainties, and applying critical perturbations. Some approaches to automatic test patterns generation are under investigation, but not yet applied. To date, the quality of the control algorithm validation is not satisfactory and important improvements will be necessary to cope with the safety issues that will be raised by next generation x-by-wire systems. Ideally, validation and formal verification should be completely automatic. To the best of our knowledge, there is no tool available in the market for performance analysis, robust stability, and formal verification of both continuous and discrete specifications. Hybrid system techniques can contribute significantly to the improvement of the validation process:

- Validation has to be supported by tools for efficient simulations of hybrid models; (robust) stability and (robust) performance analysis; and invariant set and robust invariant set computations.
- Methodologies and tools should be developed for automatic validation against formalized hybrid specifications and safety relevant conditions, and automatic optimized test patterns generation reaching specified levels of coverage.

- Interesting validation problems are related to the computation of conservative approximations for the largest sets of parameter uncertainties (or calibration/implementation parameters) for which the desired performances are achieved.

5.2.4. Hardware/Software Components

The design of HW/SW implementation of ECUs follows to date the standard methodologies for hardware and software development. However, HW/SW implementation of the control algorithms may offer an interesting and little explored application of hybrid formalisms. In particular, we see value for hybrid methodologies at the boundary between control engineering and implementation design. The methodologies and the design tools in the control domain and the HW/SW implementation domains are often not sufficiently well integrated. The specification for the HW/SW implementation has to be model-based and has to include all the details necessary for a correct implementation of the algorithms i.e., complete description of the algorithm; specification of the computation accuracy both in the time domain and the value domain; execution order and synchronization; priorities in case of resource sharing (CPU, communication, etc.); communication specifications; and data storage requirements. The model-based algorithm description has to be integrated with tools for automatic code generation for software implementation and with tools for automatic synthesis for hardware design. Finally, the specification for the HW/SW implementation should ideally provide executable acceptance tests to be applied to the HW/SW implementation. Hybrid system techniques can be applied to the development of

- Methodologies and tools for the definition and validation of implementation constraints. The degradation of the execution of control algorithms owing to the implementation on bounded resource platforms has to be exported and modeled in an abstract way at the control system level to obtain constraints for the implementation from closed-loop analysis and executable acceptance tests.
- Tools suitable for the description of the implementation requirements includes the algorithm functional description; the computation accuracy and the other implementation requirements and constraints mentioned above; and the implementation acceptance tests for the validation of the HW/SW implementation.

6. Hybrid Control in Communication Systems

6.1. Cross-Fertilization of Control and Communication

The rapid technologies advances in embedded processors and networking has recently motivated interests and expectations for a large set of applications that rely on networked embedded systems [32]. Embedded processors are widely used in, e.g., automotive, entertainment and communication devices, and in a wide range of appliances. However, networking technologies, especially those based on the wireless medium, have also known a rapid growth, thus paving the way to conceive large sets of radio interconnected embedded devices. As microfabrication technology advances make it cheaper to build single sensor and actuator nodes, a large set of new applications can be envisaged in environment monitoring, smart agriculture, energy efficient heating, home automation, etc. Moreover, a major impact of wireless interconnections can be expected in industrial automation, where updating production lines will not induce anymore expensive and time-consuming re-cabling. In summary, we can envisage a networked embedded system as an eventually large set of sensors, controllers and actuators linked via wired and wireless communication channels. Although technology advances and prospected applications are progressing, it has to be recognized that developing sound methods for design and operations of such systems is a major research challenge [39,65]. In fact, traditional control theory typically relies on accurate and lossless feedbacks, with no time delay jitter. On the other hand, communication networks are designed for applications that typically are either delay tolerant (e.g., data transfer) or error tolerant (e.g., conversational services). Looking at the design problem from the communication side and thus keeping in mind the layered open system interconnection (OSI) model, we can cast the control over network problem as an application to be delivered over an underlying protocol stack.

A control application may require large communication channel capacities, if frequent and accurate feedbacks are required. In a shared resource environment this may induce larger delays, which might prevent meeting real-time constraints, whereas contextual information losses might prevent meeting safety constraints. Integrated design of channel coding and control algorithms is discussed in Ref. [59].

An approach to jointly design control algorithms and the underlying communication network has been recently devised in Ref. [52], where the problem has been cast according to a cross-layer paradigm that combines physical layer, media access control layer and control application. Modeling the various interacting components is not trivial even in simplified contexts, whereas it appears challenging if we also want to look at the wireless network as a useful ubiquitous computing resource for processing and decision: for example, distributed source coding and network coding can be intended as parts of novel computing paradigms that arise in the devised networking context.

A close link between communication and control also arises when we consider that control functionalities are omnipresent in communication systems, with critical examples such as the power control algorithms in cellular systems and the transport control protocol (TCP) in the Internet. In general, any modern communication system, which is targeted to provide a multitude of services, requires adequate control of its communication resources. The problem is exacerbated if we consider that end-to-end communications may often require inter-working among heterogeneous networks (e.g., wireless and wired), wherein the concept of ambient networks for coordinating control functionalities in different transport networks is currently emerging. Especially in the wireless context, where the scarce availability of spectrum slots forces us to handle resource sharing in the access portion of the network, development of effective techniques for management of network resources is recognized at least as important as the development of new transmission techniques that can counteract the hostile propagation channel and increase channel capacity (e.g., advanced channel coding and error recovery mechanisms, modulation techniques and diversity schemes). In fact, ultimate achievable spectral efficiency depends on efficient use of resources [e.g., assignment of codes to users and base stations (BSs), power levels, coverage handling through efficient beam-forming], which impact on the interference amount that each user signal has to counteract. Although the evident relevance of these control and scheduling problems, many of the mechanisms have not been designed using a model-based control framework, but merely heuristics and *ad hoc* solutions. When designing new communication protocols it is of fundamental importance to be able to assess the benefit of also transmitting status information related to the data transmission. In view of the increased system complexity this type of protocols imply, questions such as what information should be transmitted and the quantization of the gain, e.g., in terms of traffic predictability and reliability, needs to be addressed. These are core issues in any network communication system and they are being far from well understood to date. It is well known in control theory that old feedback information is of little use; on the contrary it tends to destabilize the system. The implication of this is that status information in a network is perishable and the influence of time delays is an important issue. Control theory has proven to be a suitable framework to analyze such aspects from a systems perspective.

A common need of the two facets of control in communication systems depicted above consists in (1) developing sound modeling of complex systems and environments, and (2) subsequently find suitable optimization and control strategies. Specifically, as it will be remarked throughout the examples below, hybrid systems theory intrinsically provides the mathematical basis for modeling the dynamics of our control systems. Although the suitability of such models have been proven and exploited recently in some domains, only few and limited attempts (e.g., Refs [2,45]) can be found in the literature for communication systems and protocols. Therefore, in this section we intend to emphasize how hybrid dynamics may actually arise in problems related to operation of communication systems. Specifically, we focus on wireless systems and provide some details on power control in interference-limited fading wireless channels and the behavior of TCP over a wireless interface. First, let us recall the layered architecture of communication systems.

6.2. Layered Architectures for Networked Systems

In the design of large-scale systems, it is crucial to have a design approach based on composition and modularity. This helps the designer to argue about the system and understand interactions and dynamics. Layered system architectures are common in many disciplines and widely used in practice. It is surprising that there is not much theory that supports the use [83]. An area that has gained tremendously from a standardized architecture is communication networks. The architecture is an important contributor to the Internet revolution. Here, we briefly discuss the OSI model for communication networks and discuss how it relates to networked embedded systems and hybrid dynamics.

The OSI reference model is shown to the left in Fig. 14, see Ref. [29,84] for details. The model is

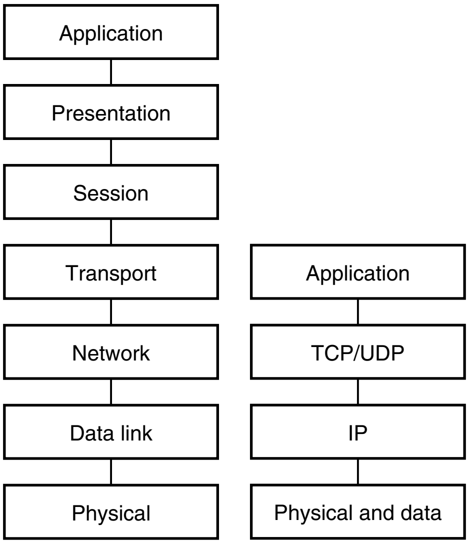

Fig. 14. The layered OSI model for communication networks to the left and the special instance of the Internet to the right.

decomposed of seven layers with specified network functions. The lowest layer is the physical layer, which is concerned with transmission of signals from a transmitter to a receiver across a physical medium. Choice of the modulation format is a typical aspect of the physical layer. The data link layer adds error correction on bit level to the unreliable point-to-point communication provided by the physical layer. The main function of the network layer is routing, i.e., to find out where to send packets (sequences of bits). This is typically performed by appending an address field to the packet. The transport layer handles messages. It forwards the messages between certain ports of the computers. The session layer sets up sessions between the computers, so that information can be exchanged. The presentation layer makes sure that the syntax used in different computers are translated and it also handles encryption and decryption. Finally, the application layer provides high-level functions needed for the user applications, e.g., file transfer. For the Internet architecture it is common to group some of the OSI layers together. The layered architecture of the Internet is shown in Fig. 14. The top three OSI layers have been merged into one. The transport layer is based on either the TCP or the user data protocol (UDP). The network layer is defined by the Internet protocol (IP).

Hybrid models are closely related to layered system architectures. The choice of mathematical modeling framework used in communication networks depends obviously on the purpose of the model. One way of classifying models is by linking them to layers of the OSI model. Models for the physical layer should capture radio signal propagation, interference, modulation, etc.; models corresponding to the data link layer are of information theoretic character, etc. Cross-layer design is an intensive area of development for particularly wireless networks. When two or more layers are considered, it is natural to be faced with a mixture of model classes. As an example, consider a continuous flow modeling the data transmission of the transport layer. It might be convenient to use such an abstraction, even if the data in reality are transmitted as finite messages at discrete instances of time. Routing decisions are of event-triggered nature and may depend on network changes or competing traffic. Hence, to analyze traffic flow over individual links, we might end up with a model having a hybrid nature with a mixture of time-triggered (continuous) dynamics and event-triggered (discrete) dynamics. For further discussion on such a model for TCP, see Ref. [45], where the hybrid nature of TCP itself is also investigated. Below we discuss a related model for TCP over wireless systems. It has recently been pointed out that caution needs to be taken in introducing new cross-layer mechanisms [48]. In understanding the interactions, such mechanisms may lead to, a rigorous modeling framework is important.

Hierarchical architectures are common also in many control applications, such as in air-traffic management, distributed process control systems, intelligent vehicle highway systems, mobile robotics, etc. For synthesizing controllers and verifying designs, it is useful to employ a hybrid systems framework for hierarchical control systems. Indeed, part of the motivation for developing hybrid systems theory comes from modeling hierarchical control systems [83].

6.3. Power Control in Interference-Limited Fading Wireless Channels

When considering interference-limited wireless systems, link performance is mainly determined by the signal-to-interference ratio (SIR) statistics. Random channel fluctuations and interfering signals ultimately determine link performance. This is especially true for those systems that are based on direct sequence/code division multiple access (DS/CDMA), in which user signals are allowed to overlap both in time and in frequency, being only distinguishable through spreading and scrambling codes. DS/CDMA is a basic access technique for the radio interface of third generation wireless systems, e.g., in so-called W-CDMA and CDMA2000. These systems have been defined for supporting heterogeneous traffic, with a variety of source rates and quality of service requirements. The achievement of large capacities and adequate performance in this context is a challenging task, and requires a proper allocation of system resources. Moreover, as the environment is time-varying, adaptive transmission techniques are envisaged, with various combinations of alternatives for power and rate allocation, coding formats, error recovery mechanisms, and so on.

Among various techniques, power control is an essential functionality to combat the near–far effect and let each user achieve its target SIR at every time. Apart from the open-loop component, in modern systems there is a closed-loop control. It consists of an outer loop and an inner loop. The outer loop adapts the target SIR based on link quality estimation, while the inner loop is responsible for power adaptation in order to meet the target SIR. Let us consider the reverse link in a multi-user system, i.e., mobile station (MS) to BS. The closed-loop acts for each user signal, so that there is a set of interacting loops, each one acting as follows (Fig. 15).

At each symbol time, an estimation of the SIR is performed, for example, with an averaging filter over a block of B symbol intervals and compared with the target level. Thus, a new estimate of the SIR is available at the filter output every B symbol time interval. The difference between the filter output and the target level is then used to decide which is the power correction to be applied at the MS. Specifically, such a difference is quantized in order to meet the available bandwidth for the power update command to send on a forward (BS to MS) link power control channel. For example, binary quantization could be used, where either the bit 1 is sent to increase the MS transmission power or the bit 0 is sent to decrease the power. After a delay, owing to propagation and processing, the command is received by the MS. The new transmitted power at the MS is obtained by applying the correction to the last transmitted power level. The transmitted power is kept constant until a new update command is received.

A well-founded view of power control is provided in Ref. [41], where it is evidenced that a system with quantized feedback is concerned. Hybrid dynamics also arise from that the target SIR updates are events that take place on a larger time scale with respect to regular (synchronous) transmission power updates forced by the inner loop. Moreover, power control cannot be considered alone in the adaptive transmission context we have envisaged. In fact, rate adaptation among a limited set of alternatives is allowed and jointly combined with target settings in the outer loop. In addition, adaptive coding formats also interact with power control and contribute to define the event-based component of the hybrid system. An attempt to model the complexity of interactions among all these components has been proposed in some recent papers [6,71]. In particular, in Ref. [6] a model is derived for the power-controlled and interference-limited wireless channel, and then evaluation of performances of forward error correction and hybrid automatic repeat request (ARQ) error control coding is performed over the abstracted channel model.

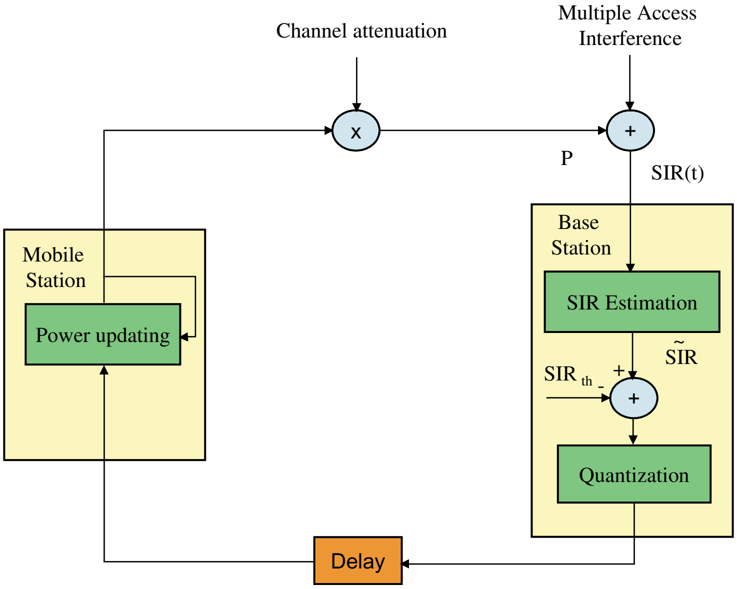

Fig. 15. Power control of third generation wireless system. The closed-loop control system has hybrid dynamics in that there is a mixture of time-triggered and event-triggered signals, and the communicated control command is quantized.

6.4. TCP/IP Over Wireless Systems

A sound layered communication architecture is important [48]. The tremendous growth of the Internet is to a large extent due to the architecture illustrated in Fig. 14. New technology and cross-layer algorithms may, however, challenge the separation of the layers. One example is given by wireless Internet, in which there are one or more wired links replaced by radio transmissions. In this case, as is shown below, the physical and data link layer may influence upper layers and thereby deteriorate performance.

Consider a single user that connects to the Internet through a mobile terminal. An illustration of the system is shown in Fig. 16, where four interacting feedback control loops are indicated. At the lowest level, the transmission power is controlled in order to keep the SIR at a desired level, as discussed previously. This is a fast inner loop (1) intended to reject disturbances in the form of varying radio conditions. On top of this, we have an outer power control loop (2) that tries to keep the frame error rate constant, by adjusting the target SIR of the inner loop. Next, we have a local link-layer retransmission of damaged radio frames through the automatic repeat request mechanism (3). Finally, the end-to-end congestion control of TCP (4) provides a reliable end-to-end transport for the application with built-in flow control.

Cross-layer interactions may reduce the end-to-end throughput. For the wireless Internet scenario introduced above, the two nested power control loops are supposed to support the separation of the physical layer from the data link layer. The automatic repeat request should separate the data link layer from the network layer. TCP should separate the transport layer from the application by providing a virtual end-to-end connection between the mobile terminal and the Internet server. A timeout event in TCP occurs when a packet, or its acknowledgment, is delayed too long. The timeout mechanism is supposed to indicate severe congestion and thereby force TCP to reduce the sending rate drastically. Spurious timeouts, i.e., timeouts that are not due to network congestion, are known to sometimes occur if the lower layers are not working properly [53]. It was recently shown that realistically modeled radio links influence the delay distribution of the TCP segments and that they induce spurious timeouts [63]. The performance degradation measured in throughput can be up to ~15%. The analysis is based on a hybrid model derived from Fig. 16, where the power control loops are modeled through a Markov chain. The influence of a more detailed radio model was studied in Ref. [33].

There are a few proposals to improve TCP performance over radio links. One is to change the TCP algorithms to make them more robust to link irregularities [57]. Another is to engineer the link-layer, to give it properties that plain TCP handles well. In view of the discussion above on that caution needs to be taken in introducing new cross-layer mechanisms, it is not always desirable to optimize one layer of the network architecture for a specific application or operating condition. Another drawback with modifying TCP algorithms is that deployment of new algorithms affect all Internet end systems, which makes it a slow and costly process. Tuning the link properties is more practical from a deployment point of view, at least if the tuning can be performed before widespread adoption of a new link type. If possible, the radio links should be made as friendly as possible to a large class of data traffic [63]. The fundamental

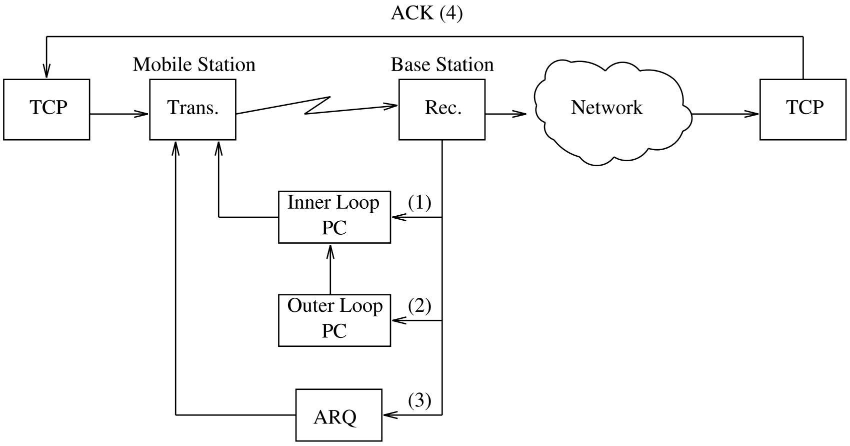

Fig. 16. System overview of wireless Internet in a case when a mobile user connects to an Internet server through a TCP/IP session. Four of the feedback control loops that support the separation of the layers in the network architecture are indicated: the inner power control loop (1), outer power control loop (2), link-layer retransmission (3) and end-to-end congestion control (4).

limitations need to be investigated of the system. It was shown in Ref. [64] that without any cross-layer signaling, the delay distribution could in a very simple way be adjusted by adding a suitable delay to certain TCP segments and thereby gain considerable improvements of the throughput. The design and implementation of new hybrid controller for improved user experience of wireless Internet was discussed in Ref. [62]. Information on radio bandwidth and queue length available in the so-called radio network controller (RNC), close to the BS, is used in a proxy that resides between the Internet and the cellular system. The hybrid control algorithm in the proxy sets the window size according to event-triggered information on radio bandwidth changes and time-triggered information on the queue length of the RNC. Figure 17 illustrates a typical improvement of bandwidth utilization of the proxy setup compared with the nominal setup to date. The available bandwidth of the wireless link is shown by the dashed line. The bandwidth variations are due to the varying conditions of the radio link. The dotted line shows the utilization for the nominal setup with TCP Reno (without proxy). The solid line shows the utilization with the new proxy solution. Note that none of the setups achieves full link utilization, but the proxy setup tracks the bandwidth variations much better than the nominal setup, and it adjusts more quickly to the available bandwidth. This results in faster response and better utilization of the radio link. The initial delay of ~0.4 s corresponds to the TCP connection establishment, and affects both setups equally. The oscillations in the nominal setup is a result of the bursty behavior of the Slow Start mechanism of standard TCP.

7. Concluding Remarks

The topic of hybrid control has attracted considerable attention from the research community in recent years. This has produced a number of theoretical and computational methods, which are now available to the designer and have been used successfully in a wide range of applications. There are still, however, many details that need to be clarified, as well as substantial problems that have not been studied in sufficient detail, both in theory and in applications. We conclude this overview by summarizing some of these issues.

From the point of view of theory, a number of interesting problems arise in the area of dynamic feedback, which is still unexplored to a large extent. The rapid development in the design of hybrid

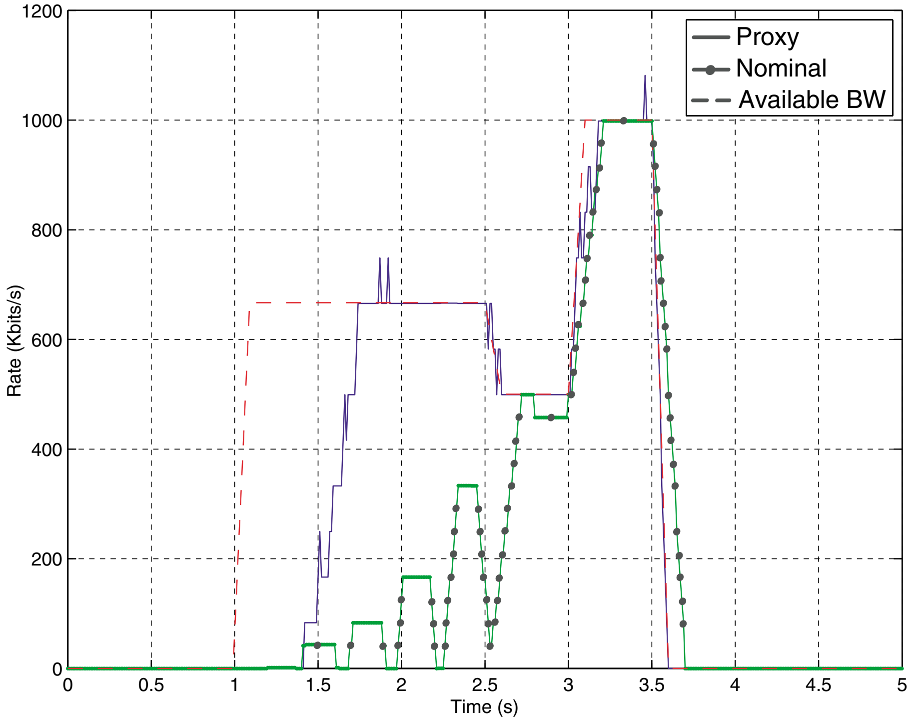

Fig. 17. Improved TCP over a wireless link through a new proxy-based hybrid controller. Available bandwidth (dashed) over the wireless link, compared to the actual utilization for the proxy setup (solid) and the nominal setup (dotted).

observers witnessed in recent years poses the question of how the system will perform if the state estimates that the observers produce are used in state feedback. A second area that, despite numerous contributions, still poses formidable problems is the area of hybrid games. As in the robust control of continuous systems, gaming appears in hybrid systems when one adopts a non-deterministic point of view to the control of uncertain systems. It is hoped that advances hybrid gaming will eventually lead to a robust control theory for classes of uncertain hybrid systems. Finally, stochastic hybrid systems pose a number of challenges. Progress in this area could come by blending results for stochastic discrete event systems with results on the l_∞ optimal control of stochastic systems.

In terms of application to power electronics and power systems in general, hybrid systems methods tailored to the specific problems that arise in this area need to be developed. Progress in this direction, in combination with the continuous increase in computational power that is available for the control of such systems, enables the control and power electronics communities to revisit some traditionally established methods in a more theoretically rigorous and systematic way.

For industrial processes, the tasks of verifying properties such as safety for industrial plants and of computing optimal control trajectories for startup and shutdown, presented in Section 4, are just two examples where industrial practice can be supported by the use of hybrid techniques. A number of successful applications of such techniques have been reported in the literature; however, most of these applications deal with relatively small parts of industrial plants, or systems on a laboratory scale. To extend the hybrid approach to large-scale industrial problems, practising engineers need to embrace hybrid control techniques and include them into their toolboxes. This in turn requires an increased awareness of existing hybrid techniques among practicing engineers and an increased efficiency of hybrid methods to enhance their applicability to industrial-size problems.

For automotive applications, we described critically the automotive electronic design flow in use now with the intention of underlining where hybrid methods can be of use to improve the quality of design. The quality of present products is far from being satisfactory in view of the rapid advances of integrated circuit and system technology, and of the ever increasing demands on functionality and time to market. Although we are optimistic that hybrid systems will be of good use in automotive electronics, the difficulties in propagating this approach to design cannot be overemphasized. Similar to industrial control problems, a coherent set of tools and a training approach should be developed to make hybrid systems and their relationship with embedded systems appealing to automotive engineers. The most obvious application of hybrid systems is for modeling and control at the highest level of abstraction, e.g., in engine control. However, we believe that a profitable application will also be at the boundary of control design and implementation engineering where the effects of limited resources and physics on the control performance has to be captured to verify the correctness of overall system (plant and controller).

Finally, current work on hybrid systems methods for communication networks is progressing along the two main tracks of control of networks and control over networks, discussed in Section 6. Specific interests include various aspects of distributed radio resource management in evolved third generation wireless systems, and efficient design and operations of *ad hoc* wireless networks for control applications.

Acknowledgements

The work presented in this paper was carried out in the framework of the HyCon Network of Excellence, supported by the European Commission under contract number FP6-IST-511368.

References

1. ABB Asea Brown Boveri Ltd. Product webpage of ACS 6000. Online document: www.abb.com/motors&drives
2. Alpcan T, Basar T. A hybrid systems model for power control in multicell wireless data networks. Perform Evaluation 2004; 57: 477–495
3. Antoniotti M, Balluchi A, Benvenuti L, Ferrari A, Pinello C, Sangiovanni-Vincentelli AL, Flora R, Nesci W, Rossi C, Serra G, Tabaro M. A top-down constraints-driven design methodology for powertrain control system. In: Proceedings of the global powertrain congress (GPC98) volume emissions, testing and controls, Detroit, MI. October, 1998, pp 74–84
4. Asarin E, Bournez O, Dang T, Maler O, Pnueli A. Effective synthesis of switching controllers for linear systems. Proc IEEE 2000; 88(7): 1011–1025
5. Asarin E, Maler O, Pnueli A. Symbolic controller synthesis for discrete and timed systems. In: Antsaklis P, Kohn W, Nerode A, Sastry S (eds), Proceedings of Hybrid Systems II, LNCS 999. Springer-Verlag, Berlin, 1995, pp 1–20
6. Babich F, Santucci F, Graziosi F. Modeling of power-controlled interference-limited wireless channels. In: Proceedings of the WPMC. IEEE, 2002
7. Balluchi A, Benvenuti L, Di Benedetto MD, Pinello C, Sangiovanni-Vincentelli AL. Automotive engine control and hybrid systems: challenges and opportunities. Proc IEEE 2000; 88(7): 888–912

8. Balluchi A, Benvenuti L, Lemma C, Murrieri P, Sangiovanni-Vincentelli AL. Hybrid models of an automotive driveline. Technical report, PARADES, Rome, I, December, 2004
9. Balluchi A, Benvenuti L, Lemma C, Sangiovanni-Vincentelli AL, Serra G. Actual engaged gear identification: a hybrid observer approach. In to be presented at 16th IFAC world congress, Prague, Czech Republic. July, 2005
10. Balluchi A, Di Benedetto MD, Ferrari A, Gaviani G, Girasole G, Grossi C, Nesci W, Pennese M, Sangiovanni-Vincentelli AL. Design of a motorcycle engine control unit using an integrated control-implementation approach. In: Proceedings of the 1st IFAC workshop on advances in automatic control, Salerno, Italy. April, 2004, pp 218–225
11. Balluchi A, Di Benedetto MD, Pinello C, Rossi C, Sangiovanni-Vincentelli AL. Hybrid control in automotive applications: the cut-off control. Automatica 1999; 35: 519–535
12. Baotic M, Vasak M, Morari M, Peric N. Hybrid theory based optimal control of electronic throttle. In: Proceedings of the IEEE American control conference (ACC 2003), Denver, CO. June, 2003, pp 5209–5214
13. Bauer N, Engell S, Huuck R, Lohmann S, Lukoschus B, Remelhe MP, Stursberg O. Verification of PLC programs given as sequential function charts. In: Proceedings of the integration of software specification techniques for applications in engineering, LNCS 3147. Springer, 2004, pp 517–540
14. Bemporad A, Bianchini G, Brogi F, Barbagli F. Passivity analysis and passification of discrete-time hybrid systems. In: Proceedings of the IFAC world congress, Prague, Czech Republic, 2005, Submitted
15. Bemporad A, Garulli A, Paoletti S, Vicino A. A bounded-error approach to piecewise affine system identification. IEEE Trans Autom Control, 2004. Accepted for publication as a regular paper
16. Bemporad A, Morari M. Control of systems integrating logic, dynamics and constraints. Automatica 1999; 35(3): 407–427
17. Bemporad A, Morari M, Ricker NL. Model predictive control toolbox for Matlab – user's guide. The Mathworks, Inc., 2004, http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/
18. Bengtsson J, Larsen KG, Larsson F, Petterson P, Yi W. UPAAL: a tool suit for automatic verification of real-time systems. In: Proceedings of the conference on hybrid systems III, LNCS 1066, Springer-Verlag, Berlin, 1996, pp 232–243
19. Bensoussan A, Menaldi JL. Hybrid control and dynamic programming. Dynam Continuous Discrete Impulsive Sys 1997; 3: 395–442
20. Berhmann G, Fehnker A, Hune T, Larsen KG, Pettersson P, Romijn J, Vaandrager F. Minimum cost reachability for priced timed automata. In: Di Benedetto M, Sangiovanni-Vincentelli A (eds), Proceedings of the conference on hybrid systems: computation and control, LNCS 2034. Springer-Verlag, Berlin, 2001, pp 147–161
21. Borrelli F. Constrained optimal control of linear and hybrid systems, volume 290 of Lecture Notes in Control and Information Sciences. Springer, 2003
22. Botchkarev O, Tripakis S. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In: Lynch N, Krogh BH (eds), Proceedings of the conference on hybrid systems: computation and control, LNCS 1790. Springer-Verlag, Berlin, 2000, pp 73–88
23. Branicky MS, Borkar VS, Mitter SK. A unified framework for hybrid control: model and optimal control theory. IEEE Trans Autom Control 1998; 43(1): 31–45
24. Broucke M, Di Benedetto MD, Di Gennaro S, Sangiovanni-Vincentelli A. Theory of optimal control using bisimulations. In: Proceedings of the conference on hybrid systems: computation and control, LNCS 1790. Springer, 2000, pp 89–102
25. Chutinam A, Krogh B. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In: Vaandrager FW, van Schuppen JH (eds), Proceedings of the hybrid systems: computation and control, LNCS 1569. Springer-Verlag, Berlin, 1999, pp 76–90
26. Clarke E, Fehnker A, Han Z, Krogh BH, Ouaknine J, Stursberg O, Theobald M. Abstraction and counter-example-guided refinement in model checking of hybrid systems. Int J Foundations Comput Sci 2003; 14(4): 583–604
27. Clarke EM, Grumberg O, Peled DA. Model checking. MIT Press, 1999
28. Daws C, Olivero A, Trypakis S, Yovine S. The tool KRONOS. In: Alur R, Henzinger T, Sontag E (eds), Proceedings of the conference on Hybrid Systems III, LNCS 1066, Springer-Verlag, Berlin, 1996, pp 208–219
29. Day JD, Zimmermann H. The OSI reference model. Proc IEEE 1983; 71: 1334–1340
30. De Carlo R, Branicky M, Pettersson S, Lennarston B. Perspectives and results on the stability and stabilizability of hybrid systems. Proc IEEE 2000; 88(7): 1069–1082
31. Engell S, Lohmann S, Stursberg O. Verification of embedded supervisory controllers considering hybrid dynamics. Int J Softw Eng Knowl Eng 2005; 15(2): 307–312
32. Estrin D (ed.) Embedded, everywhere: a research agenda for networked systems of embedded computers. US National Research Council, 2001
33. Fischione C, Graziosi F, Santucci F, Möller N, Johansson KH, Hjalmarsson H. Analysis of TCP over WCDMA wireless systems under power control, MAI and link level error recovery. In: Proceedings of the conference on IWCT, 2005
34. Gao Y, Lygeros J, Quincampoix M, Seube N. On the control of uncertain impulsive systems: approximate stabilization and controlled invariance. Int J Control 2004; 77(16): 1393–1407
35. Geyer T. Low complexity model predictive control in power electronics and power systems. PhD thesis, ETH, Zurich, 2005
36. Geyer T, Papafotiou G. Direct torque control for induction motor drives: a model predictive control approach based on feasibility. In: Morari M, Thiele L (eds), Proceedings of the conference on hybrid systems: computation and control, LNCS 3414. Springer-Verlag, 2005, pp 274–290

37. Geyer T, Papafotiou G, Morari M. Verfahren zum Betrieb einer rotierenden elektrischen Maschine. European patent application pending, 2004
38. Gokbayrak K, Cassandras CG. Hybrid controllers for hierarchically decomposed systems. In: Proceedings of the conference on hybrid systems: computation and control, LNCS 1790. Springer, 2000, pp 117–129
39. Graham S, Baliga G, Kumar PR. Issues in the converging of control with communication and computing: proliferation, architecture, design, services, and middleware. In: Proceedings of the 43rd IEEE conference on decision and control, 2004
40. Grammel G, Maximum principle for a hybrid system via singular pertrubations. SIAM J Control Optim 1999; 37(4): 1162–1175
41. Gunnarsson F, Gustafsson F. Power control in wireless communications networks – from a control theory perspective. In: Proceedings of the IFAC World Congress, 2002
42. Heemels WPMH, De Schutter B, Bemporad A. Equivalence of hybrid dynamical models. Automatica 2001; 37(7): 1085–1091
43. Henzinger TA, Ho PH, Wong Toi H. A user guide to HYTECH. In: Brinksma E, Cleaveland W, Larsen K, Margaria T, Steffen B (eds), Proceedings of the conference on tools and algorithms for the construction and analysis of systems (TACAS 95), LNCS 1019. Berlin, Springer-Verlag, 1995, pp 41–71
44. Henzinger TA. The theory of hybrid automata. In: Proceedings of the 11th IEEE symposium on logic in computer science. 1996, pp 278–292
45. Hespanha J, Bohacek S, Obraczka K, Lee J. Hybrid modeling of TCP congestion control. In: di Benedetto M, Sangiovanni-Vincentelli AL (eds), Proceedings of the conference on hybrid systems: computation and control, LNCS 2034. Springer-Verlag, NY, 2001
46. Hespanha J, Morse AS. Switching between stabilizing controllers. Automatica 2002; 38(11): 1905–1917
47. International Electrotechnical Commission. Programmable controllers – programming languages. IEC 61131-3, Technical Communication No. 65, 2003
48. Kawadia V, Kumar PR. A cautionary perspective on cross-layer design. IEEE Wireless Commun 2005; 12(1): 3–11
49. Kowalewski S, Stursberg O, Bauer N. An experimental batch plant as a case example for the verification of hybrid systems. Eur J Control 2001; 7: 366–381
50. Krause PC. Analysis of electric machinery. McGraw-Hill, NY, 1986
51. Liberzon D, Morse AS. Basic problems in stability and design of switched systems. IEEE Control Syst Mag 1999; 19: 59–70
52. Liu X, Goldsmith A. Wireless network design for distributed control. In: Proceedings of the 43rd IEEE conference on decision and control, 2004
53. Ludwig R, Katz RH. The eifel algorithm: making TCP robust against spurious retransmissions. ACM Comp Commun Rev 2000; 30(1)
54. Lygeros J, Johansson KH, Simić SN, Zhang J, Sastry SS. Dynamical properties of hybrid automata. IEEE Trans Autom Control 2003; 48(1): 2–17
55. Maciejowski JM. Predictive control. Prentice Hall, 2002
56. Martin G. The future of high-level modelling and system level design: some possible methodology scenarios. In Proceedings of the 9th IEEE/DATC electronic design processes workshop (EDP2002), Monterey, CA. April, 2002
57. Mascolo S, Casetti C, Gerla M, Sanadidi MY, Wang R. TCP Westwood: bandwidth estimation for enhanced transport over wireless links. In: Proceedings of the conference on MobiCom, Rome, Italy, 2001
58. Mitchell I, Bayen AM, Tomlin CJ. Validating a Hamilton–Jacobi approximation to hybrid system reachable sets. In: Di Benedetto M, Sangiovanni-Vincentelli A (eds), Proceedings of the conference on hybrid systems: computation and control, LNCS 2034. Springer-Verlag, Berlin, 2001, pp 418–432
59. Mitter S. Control with limited information: the role of systems theory and information theory. IEEE Inform Theory Soc Newslett 2000; 50
60. Mobus R, Baotic M, Morari M. Multi-Object adaptive cruise control. In: Maler O, Pnueli A (eds), Proceedings of the conference on hybrid systems: computation and control (HSCC 2003), LNCS 2623. Springer Verlag, 2003, pp 359–374
61. Mohan N, Undeland TM, Robbins WP. Power electronics: converters, applications and design. Wiley, 1989
62. Cabrera Molero I, Möller N, Petersson J, Skog R, Arvidsson Å, Flärdh O, Johansson KH. Cross-layer adaptation for TCP-based applications in WCDMA systems. In: Proceedings of the IST mobile and wireless communications summit, Dresden, Germany, 2005
63. Möller N, Johansson KH. Influence of power control and link-level retransmissions on wireless TCP. In: Proceedings of the conference on quality of future internet services, LNCS 2811. Springer-Verlag, 2003
64. Möller N, Johansson KH, Hjalmarsson H. Making retransmission delays in wireless links friendlier to TCP. In: Proceedings of the 43rd IEEE conference on decision and control, 2004
65. Murray RM (ed.). Control in an information rich World: report of the panel on future directions in control, dynamics, and systems. SIAM, 2003
66. Network of Excellence HyCon. Taming heterogeneity and complexity of networked embedded systems. European Commission, FP6-IST-511368. Available at: www.ist-hycon.org
67. Oldenburg J, Marquardt W, Heinz D, Leineweber D. Mixed logic dynamic optimization applied to batch distillation process design. AICHE J 2003; 49: 2900–2917
68. Papafotiou G, Geyer T, Morari M. Hybrid modelling and optimal control of switch-mode dc-dc converters. In: Proceedings of the IEEE workshop on computers in power electronics (COMPEL), Champaing, IL, 2004
69. Papafotiou G, Geyer T, Morari M. Optimal direct torque control of three-phase symmetric induction motors. In: Proceedings of the 43rd IEEE conference on decision and control, Atlantis, Bahamas, December, 2004
70. Saint-Pierre P. Approximation of viability kernels and capture basins for hybrid systems. In: Proceedings of the European control conference, Porto. September 4–7, 2001, pp 2776–2783
71. Santucci F, Durastante G, Graziosi F, Fischione C. Power allocation and control in multimedia cdma wireless systems. Kluwer Telecommun Syst 2003; 23: 69–94

72. Savkin AV, Skafidas E, Evans RJ. Robust output feedback stabilizability via controller switching. Automatica 1999; 35(1): 69–74
73. Shaikh MS, Caines PE. On the optimal control of hybrid systems: optimal trajectories, switching times and location schedules. In: Maler O, Pnueli A (eds), Proceedings of the conference on hybrid systems: computation and control, LNCS 2623. Springer-Verlag, Berlin, 2003, pp 466–481
74. Silva BI, Stursberg O, Krogh B, Engell S. An assessment of the current status of algorithmic approaches to the verification of hybrid systems. In: Proceedings of the 40th IEEE conference on decision and control, 2001, pp 2867–2874
75. Smith R, Dullerud GE. Continuous-time control model validation using finite experimental data. IEEE Trans Autom Control 1996; 41(8): 1094–1105
76. Sontag ED. Nonlinear regulation: the piecewise linear approach. IEEE Trans Autom Control 1981; 26(2): 346–358
77. Stursberg O. Dynamic optimization of processing systems with mixed degrees of freedom. In Proceedings of the 7th International IFAC symposium on dynamics and control of process systems, 2004, p 164
78. Stursberg O. A graph-search algorithm for optimal control of hybrid systems. In: Proceedings of the 43rd IEEE conference on decision and control. 2004, pp 1412–1417
79. Stursberg O, Panek S, Till J, Engell S. Generation of optimal control policies for systems with switched hybrid dynamics. In: Proceedings of the conference on modelling, analysis, and design of hybrid systems, LNCS 279. Springer, 2002, pp 337–352
80. Sussmann HJ. A maximum principle for hybrid optimal control problems. In: Proceedings of the IEEE conference on decision and control, Phoenix, AZ. December 7–10, 1999, pp 425–430
81. Takahashi I, Noguchi T. A new quick response and high efficiency control strategy for the induction motor. IEEE Trans Ind Appl 1986; 22(2): 820–827
82. Tomlin CJ, Lygeros J, Sastry SS. A game theoretic approach to controller design for hybrid systems. Proc IEEE 2000; 88(7): 949–969
83. Varaiya P. Smart cars on smart roads: problems of control. IEEE Trans Autom Control 1993; 38(2): 195–207
84. Walrand J, Varaiya P. High-performance communication networks. 2nd edn. Morgan Kaufmann, 2000
85. Wicks M, Peleties P, De Carlo R. Switched controller synthesis for the quadratic stabilization of a pair of unstable linear systems. Eur Control 1998; 4: 140–147
86. Wong-Toi H, The synthesis of controllers for linear hybrid automata. In: Proceedings of the IEEE conference on decision and control, San Diego, CA. December 10–12, 1997, pp 4607–4613
87. Xu X, Antsaklis PJ. Stabilization of second order LTI switched systems. Int J Control 2000; 73(14): 1261–1279
88. Xu X, Zhai G. On practical stability and stabilization of hybrid and switched systems. In: Alur R, Pappas GJ (eds), Proceedings of the conference on hybrid systems computation and control, LNCS 2993. Springer-Verlag, Berlin, 2004, pp 615–630