Proposal for a Tutorial Session at CDC-ECC'05

Title: Hybrid Control of Networked Embedded Systems

Submitted by *Françoise Lamnabhi-Lagarrigue*, HYCON Scientific Manager

One of the aims of the HYCON Network of Excellence (see www.ist-hycon.org) funded by the European Commission, is to strengthen and integrate the scientific and technological efforts going on in Europe on the huge area of embedded control systems. The main challenge is the mastering of the complexity and heterogeneity of such systems: Embedded controllers are often distributed, forcing one to consider networks of systems instead of stand-alone devices. Even if the components are simple, their networked interaction results in a complex behavior. In addition to the physically coupled, resource-constrained nature of embedded control systems, another constraint is the likely heterogeneity in nature and function of the interacting elements that make interoperability a key concern. Summarizing, the interaction of digital controllers, communication systems and physical plants originates complex dynamic behaviors that cannot be understood intuitively.

Hybrid systems provide the modeling framework for capturing the richness of behavior characteristics of embedded systems. The key feature of hybrid systems theory is their ability to rigorously describe devices where continuous parts (governed by differential or difference equations) and discrete parts (described by finite state machines, if-then-else rules, and temporal logic) interact over time. Therefore, hybrid systems theory is naturally tailored to model phenomena that switch between operating modes. Mode transitions are triggered by variables crossing specific thresholds (state events), by the elapse of certain time periods (time events), or by external inputs (input events).

To date, these difficulties have been mostly tackled by non-rigorous methods, supported by extensive simulation. Malfunction of the control system can lead to drastic performance degradation, severe damage to humans and the environment and cause significant economic losses. Moreover existing solutions generally make a number of assumptions that often do not hold in practice. Some of these solutions may actually succeed, and others may appear to have succeeded, at least for a time. Finally, because of its relatively recent development and, above all, of its multidisciplinary nature, hybrid systems science is currently fragmented across different communities with consequent, and often unaware, overlaps due to jargon barriers and lack of integration, communication and common standard. Therefore, it is fundamental to start the development of a new strong theoretical and technological basis for efficient design and management of these systems. These developments have to be done by intensively bridging academic (theory) and industrial (implementation) worlds: in one way it is important to understand which of the actual needs from the physical field and the awareness of implementation constraints are the most relevant theoretical questions and in the other way, it is also crucial to disseminate the envelope of robust, secure, optimal, performing methods which are enabled in industry.

Besides the Research and Integration (of different natures) activities of the HYCON NoE, the set of Dissemination activities is a very important vector. The objective of this tutorial session after a six months running is therefore to first overview the main recent research advances and highlight some of the open challenging problems and then to propose techniques and describe some of the main challenges in the four application domains studied by the HYCON consortium: Energy Management; Industrial Controllers; Automotive Electronics Design; and Communications Systems. The plan of the tutorial session is the following:

- **An Overview of Research Areas in Hybrid Control**,
  by *John Lygeros*
- **Model Predictive Control in Power Electronics: an Hybrid Systems Approach,**
  by *Tobias Geyer, Georgios Papafotiou and Manfred Morari*
- **Hybrid Control Techniques for the Design of Industrial Controllers,**
  by *Sebastian Engel and Olaf Stursberg*
- **Hybrid Systems in Automotive Electronics Design,**
  by *Andrea Balluchi, Luca Benvenuti and Alberto Sangiovanni-Vicentelli*
- **On Hybrid Control Problems in Communication Systems,**
  by *Fortunato Santucci and Karl Henrik Johansson*

An overview of research areas in hybrid control

John Lygeros

Abstract— **Hybrid systems have been an active area of research for a number of years. Recently a consensus is beginning to emerge among researchers about theoretical and applied problems related to control of hybrid systems that are both important and tractable. In this overview paper we survey recent research advances and highlight some of the open problems.**

I. 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 dynamics 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 of digital devices with 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, scheduling constraints, etc. 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. Motivated by the observation that embedded systems often also have to deal with an uncertain and potentially adverserial environment, researchers have in recent years extended their study of hybrid systems beyond continuous and discrete dynamics, to include probabilistic terms. This has led to the more general class of stochastic 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, automotive control, air traffic management, industrial process control, and manufacturing and robotics; advances in many of these application areas will be surveyed in the remaining papers of this tutorial.

The control problems that have arisen in these applications 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. Generally, according to the specification the problems can also be grouped into three classes:

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.

Work carried out in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368.

John Lygeros is with the Faculty of Electrical Engineering, University of Patras, 26500 Rio, Patras, Greece, `lygeros@ee.upatras.gr`

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 paper we provide an introduction to the problems addressed in all these areas. In Section III we formulate a number of hybrid stabilization problems, state the main approaches to solving these problems, and provide references to publications where more details can be found. In Sections IV and V we do the same with optimal control problems and language specification problems, respectively. To be able to clearly state the different control problems of interest, we start by introducing a simple hybrid system model (Section II). We stress that this hybrid model is meant to be used only for illustration purposes. It is not the model used in any of the references, nor does it claim to be a general model for controlled hybrid systems.

II. A SIMPLE HYBRID CONTROL MODEL

Hybrid control problems have been formulated for both continuous- and discrete-time systems. As usual, continuous-time problems present more technical difficulties. In this section we introduce a model suitable for formulating continuous-time control problems for deterministic hybrid systems. We also discuss briefly the simplifications that arise if discrete-time systems are considered and the complications involved in extending the model to stochastic systems.

A. Syntax: Non-deterministic systems

Since we are interested in hybrid dynamics, the dynamical systems we consider involve both a continuous state (denoted by x) and a discrete state (denoted by 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 υ to denote discrete controls, u to denote continuous controls, δ to denote discrete disturbances, and d to denote continuous disturbances.

The dynamics of the state are determined through four functions: a vector field f that determines the continuous evolution, a reset map r that determines the outcome of the discrete transitions, a "guard" set that determines when discrete transitions can take place, and a "domain" set Dom that determines when continuous evolution is possible. The following definition formalizes the details.

Definition 1 (Hybrid game automaton): A hybrid game automaton (HGA) characterizes the evolution of

- discrete state variables $q \in Q$ and continuous state variables $x \in X$,
- discrete control inputs $\upsilon \in \Upsilon$ and continuous control inputs $u \in U$ and
- discrete disturbance inputs $\delta \in \Delta$ and continuous disturbance inputs $d \in D$

by means of four functions

- a vector field $f : Q \times X \times U \times D \to X$,
- a domain set $\mathrm{Dom} : Q \times \Upsilon \times \Delta \to 2^X$,
- guard sets $G : Q \times Q \times \Upsilon \times \Delta \to 2^X$, and
- a reset function $r : Q \times Q \times X \times U \times D \to X$.

As usual, 2^X stands for the set of all subsets (power set) of X; in other words, Dom and G are set-valued maps. For simplicity, we assume that $X = \mathbb{R}^n$, $U \subseteq \mathbb{R}^m$, and $D \subseteq \mathbb{R}^p$ for integers n, m, and p. A similar definition can also be formulated for discrete-time hybrid systems, simply by considering f as a transition function rather than as a vector field. In this case the discrete-time hybrid system can be considered as a simple discrete-time system, with state space $Q \times X$ and a set-valued transition relation

$$R(q, x, u, d, \upsilon, \delta) = [\{q\} \times f(q, x, u, d)] \\ \cup \left[\bigcup_{\{q' \in Q \,:\, x \in G(q, q', \upsilon, \delta)\}} \{q'\} \times r(q, q', x, u, d) \right],$$

if $x \in \mathrm{Dom}(q, \upsilon, \delta)$ and

$$\bigcup_{\{q' \in Q \,:\, x \in G(q, q', \upsilon, \delta)\}} \{q'\} \times r(q, q', x, u, d)$$

otherwise. Even though this abstraction appears convenient and is suitable for certain classes of problems, it is often desirable to exploit additional structure by developing more detailed (rather than more abstract) models of discrete-time hybrid systems.

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. These assumptions aim to ensure, among other things, 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. A typical assumption to prevent this situation is that the set $\mathrm{Dom}(q, \upsilon, \delta)$ is open and if $x \notin \mathrm{Dom}(q, \upsilon, \delta)$ then $x \in \bigcup_{q' \in Q} G(q, q', \upsilon, \delta)$. 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. In addition, Zeno controllers require infinitely fast switching and cannot be implemented in practice. For these reasons, the Zeno phenomenon is usually forbidden by direct assumptions. In some cases, structural assumptions are introduced on the model to prevent Zeno solutions (e.g., by enforcing a lower bound on the time between discrete transitions or the time to traverse each discrete state cycle).

Many of the assumptions discussed here can be relaxed, replaced by other variants, or dropped altogether; for example, if we consider relaxed controls in optimal control problems, convexity and compactness assumptions are typically not needed. For discrete-time hybrid systems, most of these assumptions are unnecessary. For example, deadlock and the Zeno phenomenon are typically not issues for discrete-time systems.

B. Semantics: Solutions or runs

To formally define the solutions of this class of hybrid systems, we recall the following notion from [1].

Definition 2 (Hybrid time set): 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 $i < N$, $I_i = [\tau_i, \tau_i']$;
- if $N < \infty$, then either $I_N = [\tau_N, \tau_N']$, or $I_N = [\tau_N, \tau_N')$, possibly with $\tau_N' = \infty$;
- for all i, $\tau_i \leq \tau_i' = \tau_{i+1}$.

Since the dynamical systems considered here are time invariant, without loss of generality we can assume that $\tau_0 = 0$. It easy to see that, although more complicated than the usual time sets (the real numbers for continuous-time systems or the integers for discrete-time systems), hybrid time sets are reasonably well-behaved mathematical objects. For example, each hybrid time set is totally ordered, whereas the set of all hybrid time sets is partially ordered. One can therefore naturally define prefixes and suffixes of a hybrid time set, maximal elements of a collection of hybrid time sets, etc. For discrete-time hybrid systems, the introduction of hybrid time sets is unnecessary, since the set of integers or natural numbers can typically be used.

Roughly speaking, the solution of an HGA (often called a "run" or an "execution") is defined over a hybrid time set τ and involves 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).

The following concept helps to formalize the above discussion.

Definition 3 (Hybrid trajectory): Given a set of variables, a, that take values in a set A, a hybrid trajectory over this set of variables is a pair (τ, a) where $\tau = \{I_i\}_{i=0}^N$ is a hybrid time set and $a = \{a_i(\cdot)\}_{i=0}^N$ is a sequence of functions $a_i(\cdot) : I_i \to A$.

The solutions of the HGA can now be defined as hybrid trajectories over its state and input variables.

Definition 4 (Run): A run of an HGA is a hybrid trajectory $(\tau, q, x, \upsilon, u, \delta, d)$ over its state and input variables that satisfies 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 $t \in I_i$.
  3) $x_i(\cdot)$ is a solution of the differential equation

$$\dot{x}_i(t) = f(q_i(t), x_i(t), u_i(t), d_i(t))$$

  over the interval I_i starting at $x_i(\tau_i)$.

  4) $x_i(t) \in \text{Dom}(q_i(t), \upsilon_i(t), \delta_i(t))$ for all $t \in [\tau_i, \tau_i')$.

For discrete-time hybrid systems the definition of a run is again much simpler. A run can simply be defined as a finite or infinite sequence of states and inputs, $\{q_i, x_i, u_i, d_i, \upsilon_i, \delta_i\}_{i=0}^N$, such that for all i

$$(q_{i+1}, x_{i+1}) \in R(q_i, x_i, u_i, d_i, \upsilon_i, \delta_i).$$

C. Classification of control action

The preceding 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.

Notice that the model implicitly restricts the influence of the discrete inputs υ and δ to the timing and discrete destination of discrete transitions and the influence of the continuous inputs u and d to continuous evolution and the continuous destination of discrete transitions. At this level of generality all inputs could, in fact, be allowed to influence all aspects of the evolution of the system. Caution should be taken, however, when doing this, since experience suggests that it tends to severely complicate the technicalities associated with the definition of runs, ensuring that runs exist for all inputs, preventing chattering strategies, etc. Experience also suggests that this type of mixing of discrete and continuous inputs is rarely needed in practice.

Another 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 (for example, reachability 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.

D. Stochastic hybrid systems

The controlled hybrid system model presented above allows one to capture a number of interesting and important hybrid phenomena. Many of the deterministic and non-deterministic hybrid control problems considered in the literature can be recast in this framework. The model, however, does not contain any stochastic terms. The formal definition of stochastic hybrid models requires considerable mathematical overhead, even in the simplest cases. Here we briefly describe the types of stochastic phenomena that can appear in hybrid systems, only to familiarize the reader with the issues that arise; more details can of course be found in the references.

Stochastic terms can enter hybrid dynamics in a number of different places:

1) Continuous evolution may be governed by stochastic differential equations.
2) Discrete transitions may take place spontaneously, at a given, possibly state-dependent, rate (as they do for example in discrete Markov chains). Some authors also consider forced transitions, which take place whenever the continuous state tries to leave a given set (the equivalent of the Dom set introduced above).
3) The destination of discrete transitions may be given by a probability kernel.

As for deterministic and non-deterministic systems, one can also consider controls that influence the same places: for example, controls that steer continuous evolution through controlled diffusions, influence the rate at which discrete transitions take place, determine the boundaries at which they are forced, or influence the probability distribution that determines the destination of discrete transitions. Clearly, all these alternatives allow for the formulation of countless variants of control problems.

III. STABILIZATION OF HYBRID SYSTEMS

The problem of stabilizing hybrid systems is designing 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 $(\hat{q}, \hat{x})$,

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

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

$$f(q, \hat{x}, u, d) = 0 \text{ and } r(q, q', \hat{x}, u, d) = \hat{x}$$

for all $q, q' \in Q$. An equilibrium $\hat{x}$ naturally defines an invariant set $Q \times \{\hat{x}\}$.

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),

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

Stability of hybrid systems has been extensively studied in recent years (see the overview papers [2, 3]). 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 [4–7]. Stronger results are possible for special classes of systems, such as planar systems [8]. For non-deterministic systems, in [9] an approach to the practical exponential stabilization of a class of hybrid systems with disturbances is presented. For a brief overview of stabilization problems for stochastic hybrid systems the reader is referred to [10].

IV. 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 by means of a cost function. The objective of the controller is then to minimize this cost among all possible runs 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

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

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.

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, for example, [11–13]. 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 [14, 15]. This approach has also been extended to classes of stochastic hybrid systems; see, for example, [16, 17]. Computational methods for solving the resulting variational and quasi-variational inequalities are presented in [18]. For simple classes of systems (e.g., timed automata) 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, for example, [19–22].

A somewhat different optimal control problem arises when one tries to control hybrid systems using model predictive or receding horizon techniques. Generally, the aim here is to use a model to predict the future evolution of the system under different inputs and then employ optimization algorithms to select the inputs that promise the "best" future. The initial part of these inputs is applied to the system, a new measurement is taken (providing feedback), and the process is repeated. For hybrid systems, such a model predictive control approach has primarily been studied in discrete time; see, for example, [23, 24]. The toolbox of [25] provides functions for the numerical solution of hybrid model predictive control problems (and much more).

V. 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

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

The name "safety specifications" (which is given 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., in an air traffic management system to ensure that aircraft do not come closer to one another than a certain minimum distance.

Safety specifications are usually easy to meet (e.g., if aircraft never take off, mid-air 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

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

In the air traffic context a minimal liveness type requirement is to make sure that the aircraft eventually arrive at their destination. Mixing different types of specifications like 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, mostly under the name *synthesis problems*. The approach was then extended to classes of hybrid systems such as timed automata (systems with continuous dynamics of the form $\dot{x} = 1$, [26, 27]) and rectangular automata (systems with continuous dynamics of the form

$\dot{x} \in [l, u]$ for fixed parameters l, u, [28]). For systems of this type, exact and automatic computation of the controllers may be possible using model checking tools [29–31]. 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 be solved automatically in discrete time using methods from mathematical programming [25].

Extensions to general classes of hybrid systems in continuous time have been concerned primarily with computable numerical approximations of reachable sets using polyhedral approximations [32–35], ellipsoidal approximations [36], or more general classes of sets (e.g., defined using the solutions of the continuous system [37]). A useful link in this direction has been the relation between reachability problems and optimal control problems with an l_∞ penalty function [38, 39]. 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 [18].

VI. CONCLUDING REMARKS AND OPEN PROBLEMS

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. We conclude this overview by listing some of these problems (by no means an exhaustive list).

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 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. General principles (like the separation principle in linear systems) are probably too much to hope for in a general hybrid setting, but substantial progress may still be possible for specific subclasses.

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. Unlike continuous systems, however, even fundamental notions such as "information" and "strategy" are still the topic of debate in hybrid systems. It is hoped that advances in this front will eventually lead to a robust control theory for classes of uncertain hybrid systems.

Finally, stochastic hybrid systems pose a number of challenges. For example, the formulation and solution of language specifications (even of the simplest safety type) for stochastic hybrid systems is still to a large extent open. Progress in this area could come by blending results for stochastic discrete event systems with results on the l_∞ optimal control of stochastic systems.

REFERENCES

[1] J. Lygeros, K. Johansson, S. Simić, J. Zhang, and S. Sastry, "Dynamical properties of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 2–17, January 2003.

[2] D. Liberzon and A. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine*, vol. 19, pp. 59–70, October 1999.

[3] R. De Carlo, M. Branicky, S. Pettersson, and B. Lennarston, "Perspectives and results on the stability and stabilizability of hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1069–1082, July 2000.

[4] M. Wicks, P. Peleties, and R. De Carlo, "Switched controller synthesis for the quadratic stabilization of a pair of unstable linear systems," *European Journal of Control*, vol. 4, pp. 140–147, 1998.

[5] A. V. Savkin, E. Skafidas, and R. Evans, "Robust output feedback stabilizability via controller switching," *Automatica*, vol. 35, no. 1, pp. 69–74, 1999.

[6] J. Hespanha and A. Morse, "Switching between stabilizing controllers," *Automatica*, vol. 38, no. 11, pp. 1905–1917, November 2002.

[7] X. Xu and G. Zhai, "On practical stability and stabilization of hybrid and switched systems," in *Hybrid Systems Computation and Control*, ser. LNCS, R. Alur and G. Pappas, Eds. Berlin: Springer-Verlag, 2004, no. 2993, pp. 615–630.

[8] X. Xu and P. Antsaklis, "Stabilization of second order LTI switched systems," *International Journal of Control*, vol. 73, no. 14, pp. 1261–1279, September 2000.

[9] Y. Gao, J. Lygeros, M. Quincampoix, and N. Seube, "On the control of uncertain impulsive systems: Approximate stabilization and controlled invariance," *International Journal of Control*, 2005, to appear.

[10] C. Yuan and J. Lygeros, "Stabilization of a class of stochastic systems with Markovian switching," in *Mathematical Theory of Networks and Systems (MTNS04)*, Leuven, Belgium 2004.

[11] G. Grammel, "Maximum principle for a hybrid system via singular pertrubations," *SIAM Journal of Control and Optimization*, vol. 37, no. 4, pp. 1162–1175, 1999.

[12] H. J. Sussmann, "A maximum principle for hybrid optimal control problems," in *IEEE Conference on Decision and Control*, Phoenix, Arizona, December 7–10, 1999, pp. 425–430.

[13] M. Shaikh and P. Caines, "On the optimal control of hybrid systems: optimal trajectories, switching times and location schedules," in *Hybrid Systems: Computation and Control*, ser. LNCS, O. Maler and A. Pnueli, Eds. Berlin: Springer-Verlag, 2003, no. 2623, pp. 466–481.

[14] A. Bensoussan and J. Menaldi, "Hybrid control and dynamic programming," *Dynamics of Continuous, Discrete and Impulsive Systems*, no. 3, pp. 395–442, 1997.

[15] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, 1998.

[16] M. Ghosh, A. Arapostathis, and S. Marcus, "Optimal control of switching diffusions with application to flexible manufacturing systems," *SIAM Journal on Control Optimization*, vol. 31, no. 5, pp. 1183–1204, September 1993.

[17] A. Bensoussan and J. Menaldi, "Stochastic hybrid control," *Journal of Mathematical Analysis and Applications*, vol. 249, pp. 261–288, 2000.

[18] I. Mitchell, A. Bayen, and C. Tomlin, "Validating a Hamilton–Jacobi approximation to hybrid system reachable sets," in *Hybrid Systems: Computation and Control*, ser. LNCS, M. Di Benedetto and A. Sangiovanni-Vincentelli, Eds. Berlin: Springer-Verlag, 2001, no. 2034, pp. 418–432.

[19] E. Asarin and O. Maler, "As soon as possible: Time optimal control for timed automata," in *Hybrid Systems: Computation and Control*, ser. LNCS, F. W. Vaandrager and J. H. van Schuppen, Eds. Berlin: Springer-Verlag, 1999, no. 1569, pp. 19–30.

[20] R. Alur, S. La Torre, and G. Pappas, "Optimal paths in weighted timed automata," in *Hybrid Systems: Computation and Control*, ser. LNCS, M. Di Benedetto and A. Sangiovanni-Vincentelli, Eds. Berlin: Springer-Verlag, 2001, no. 2034, pp. 49–62.

[21] G. Berhmann, , A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager, "Minimum cost reachability for priced timed automata," in *Hybrid Systems: Computation and Control*, ser. LNCS, M. Di Benedetto and A. Sangiovanni-Vincentelli, Eds. Berlin: Springer-Verlag, 2001, no. 2034, pp. 147–161.

[22] O. Maler, “On optimal and suboptimal control in the presence of adversaries,” in *Proceedings of WODES04*, 2004, pp. 1–12.

[23] A. Bemporad and M. Morari, “Control of systems integrating logic dynamics and constraints,” *Automatica*, vol. 35, no. 3, pp. 407–427, March 1999.

[24] D. Mayne and S. Rakovic, “Model predictive control of contrained, piecewise affine, discrete time systems,” *International Journal of Robust and Nonlinear Control*, vol. 13, no. 3, pp. 261–279, 2003.

[25] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, “The multi-parametric toolbox (MPT),” in *Hybrid Systems Computation and Control*, ser. LNCS, R. Alur and G. Pappas, Eds. Berlin: Springer-Verlag, 2004, no. 2993, pp. 448–462.

[26] E. Asarin, O. Maler, and A. Pnueli, “Symbolic controller synthesis for discrete and timed systems,” in *Proceedings of Hybrid Systems II*, ser. LNCS, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. Berlin: Springer-Verlag, 1995, no. 999, pp. 1–20.

[27] M. Heymann, F. Lin, and G. Meyer, “Synthesis and viability of minimally interventive legal controllers for hybrid systems,” *Discrete Event Dynamic Systems: Theory and Applications*, vol. 8, no. 2, pp. 105–135, June 1998.

[28] H. Wong-Toi, “The synthesis of controllers for linear hybrid automata,” in *IEEE Conference on Decision and Control*, San Diego, California, December 10–12 1997, pp. 4607–4613.

[29] J. Bengtsson, K. Larsen, F. Larsson, P. Petterson, and W. Yi, “UPAAL: A tool suit for automatic verification of real-time systems,” in *Hybrid Systems III*, ser. LNCS. Berlin: Springer-Verlag, 1996, no. 1066, pp. 232–243.

[30] C. Daws, A. Olivero, S. Trypakis, and S. Yovine, “The tool KRONOS,” in *Hybrid Systems III*, ser. LNCS, R. Alur, T. Henzinger, and E. Sontag, Eds. Berlin: Springer-Verlag, 1996, no. 1066, pp. 208–219.

[31] T. A. Henzinger, P. H. Ho, and H. W. Toi, “A user guide to HYTECH,” in *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, ser. LNCS, E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, Eds., no. 1019. Berlin: Springer-Verlag, 1995, pp. 41–71.

[32] M. Greenstreet and I. Mitchell, “Integrating projections,” in *Hybrid Systems: Computation and Control*, ser. LNCS, S. Sastry and T. Henzinger, Eds. Berlin: Springer-Verlag, 1998, no. 1386, pp. 159–174.

[33] A. Chutinam and B. Krogh, “Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations,” in *Hybrid Systems: Computation and Control*, ser. LNCS, F. W. Vaandrager and J. H. van Schuppen, Eds. Berlin: Springer-Verlag, 1999, no. 1569, pp. 76–90.

[34] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli, “Effective synthesis of switching controllers for linear systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1011–1025, July 2000.

[35] P. Saint-Pierre, “Approximation of viability kernels and capture basins for hybrid systems,” in *European Control Conference*, Porto, September 4-7, 2001, pp. 2776–2783.

[36] O. Botchkarev and S. Tripakis, “Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations,” in *Hybrid Systems: Computation and Control*, ser. LNCS, N. Lynch and B. H. Krogh, Eds. Berlin: Springer-Verlag, 2000, no. 1790, pp. 73–88.

[37] X. Koutsoukos, P. Antsaklis, J. Stiver, and M. Lemmon, “Supervisory control of hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1026–1049, July 2000.

[38] C. Tomlin, J. Lygeros, and S. Sastry, “A game theoretic approach to controller design for hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–969, July 2000.

[39] J. Lygeros, “On reachability and minimum cost optimal control,” *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

Model Predictive Control in Power Electronics: A Hybrid Systems Approach

Tobias Geyer, Georgios Papafotiou and Manfred Morari

***Abstract*— The field of power electronics poses challenging control problems that cannot be solved satisfactorily using traditional modelling and controller design approaches. The main difficulty arises from the hybrid nature of these systems due to the presence of semiconductor switches that operate with a high switching frequency and induce different modes of operation. Since the control techniques traditionally employed in industry feature a significant potential for improving the performance and the controller design, the field of power electronics invites the application of advanced hybrid systems methodologies. As will be shown in this paper, the computational power available today and the recent theoretical advances in the control of hybrid systems allows to tackle these problems in a novel way that improves the performance of the system, and is systematic and implementable. This is illustrated by two examples, namely the Direct Torque Control of three-phase induction motors and the optimal control of switch-mode dc-dc converters.**

I. INTRODUCTION

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 (e.g. consider the problem of unregulated dc to regulated dc conversion). 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 like 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 modelling 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.

Nowadays, however, the recent theoretical advances in the field of hybrid systems, together with the latest technology developments that have made available 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 paper, where we demonstrate the application of hybrid optimal control methodologies to power electronics systems. More specifically, we show how Model Predictive Control (MPC) [1] 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.

The paper is organized in the following way: Section II gives an overview of the theoretical framework that has been used, including the basic ideas behind the off-line solution of the optimal control problem. Subsequently, we present the new modelling and optimal control approaches to the DTC problem in Section III and to the control problem of dc-dc converters in Section IV. Conclusions and an outlook are provided in Section V.

II. OPTIMAL CONTROL OF 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 nonlinear dynamics. This not only avoids a number of mathematical problems

The authors are with the Automatic Control Laboratory, ETH Zentrum – ETL, Swiss Federal Institute of Technology, CH–8092 Zürich, Switzerland `geyer,papafotiou,morari@control.ee.ethz.ch`

(like Zeno behavior), 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) [2] models and the PieceWise Affine (PWA) [3] framework. Other representations of such systems include Linear Complementarity (LC) systems, Extended Linear Complementarity (ELC) systems and Max-Min-Plus-Scaling (MMPS) systems that are, as shown in [4], equivalent to the MLD and PWA forms under mild conditions.

Model Predictive Control (MPC) [1] has been used successfully for a long time in the process industry and recently also for hybrid systems, for which, as shown in [2], 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* (MIQP).

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 [2] and [5]. Details about MPC can be found in [1].

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. The details can be found in [6], where the authors report an algorithm that generates the solution by combining dynamic programming with multi-parametric programming and some basic polyhedral manipulations. As shown in [7], 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.

III. OPTIMAL DIRECT TORQUE CONTROL OF THREE-PHASE INDUCTION MOTORS

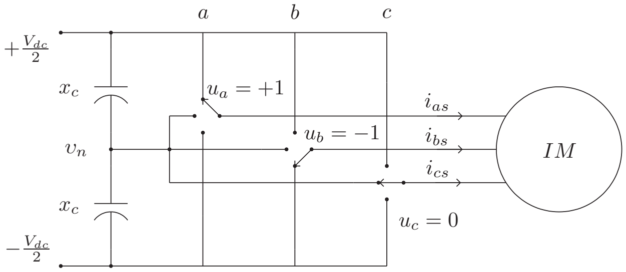


Fig. 1. The equivalent representation of a three-phase three-level inverter driving an induction motor

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 Direct Torque Control (DTC) [8], 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 (GTO) 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, 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.

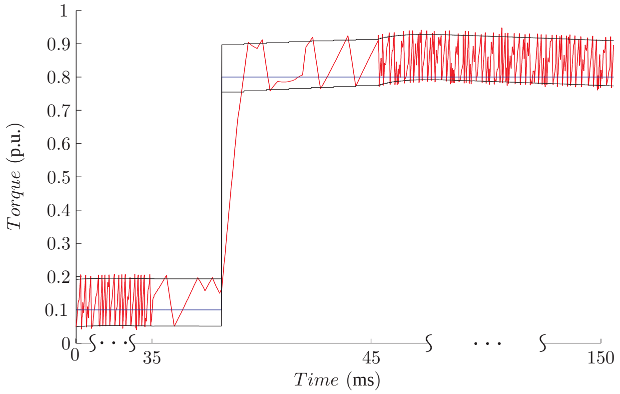
We aim at deriving MPC schemes that keep the three controlled variables (torque, flux, neutral point potential) within their given bounds, minimize the (average) switching frequency, and are conceptually and computationally simple yet yield a significant performance improvement with respect to the state 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 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 [9], [10] 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 degree 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 [11]. 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.

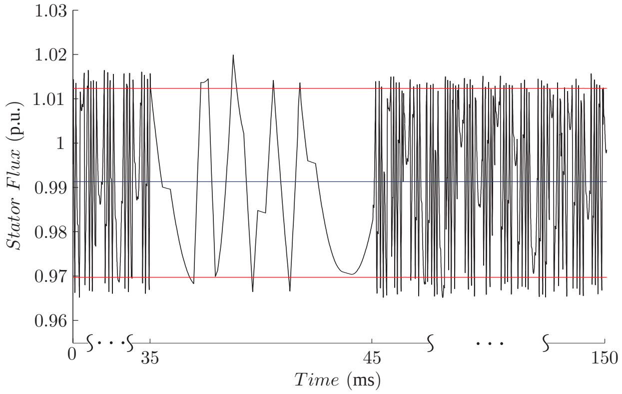
Based on the hybrid models of the DTC drive, we have proposed in [10], [12], [13] three novel control approaches to tackle the DTC problem, which are inspired by the principles of MPC and tailored to the peculiarities of DTC. For comparing with the industrial state of the art, we have used for all our simulations the Matlab/Simulink model of ABB's ACS6000 DTC drive [14] containing a squirrel-cage rotor induction motor with a rated apparent power of 2 MVA and a 4.3 kV three-level dc-link inverter. This model was provided to us by ABB in the framework of our collaboration and its use ensures a realistic set-up.

A. *DTC based on Priority Levels*

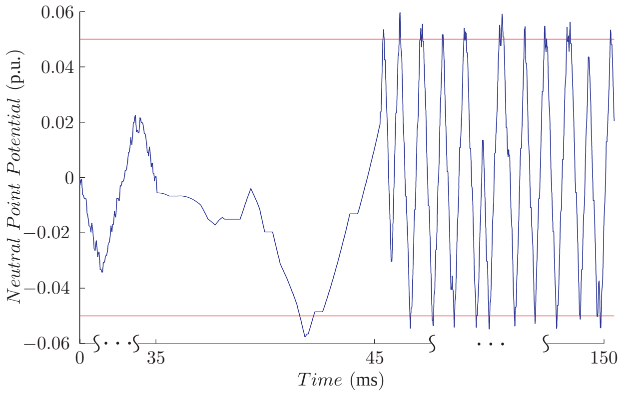
The first scheme [10] uses soft constraints to model the hysteresis bounds on the torque, stator flux and neutral point potential, and approximates the average switching frequency (over an infinite horizon) by the number of switch transitions over a short horizon. To make this approximation meaningful 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 then 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.



(a) Electromagnetic torque



(b) Stator flux



(c) Neutral point potential

Fig. 2. Closed-loop simulation of the DTC scheme based on priority levels during a step change in the torque reference

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 it also 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, while simultaneously reducing the torque and flux ripples. Yet the complexity of the control law is rather excessive [9].

B. *DTC based on Feasibility and Move Blocking*

The second scheme, presented in [12], exploits the fact that the control objectives only weakly relate to optimality

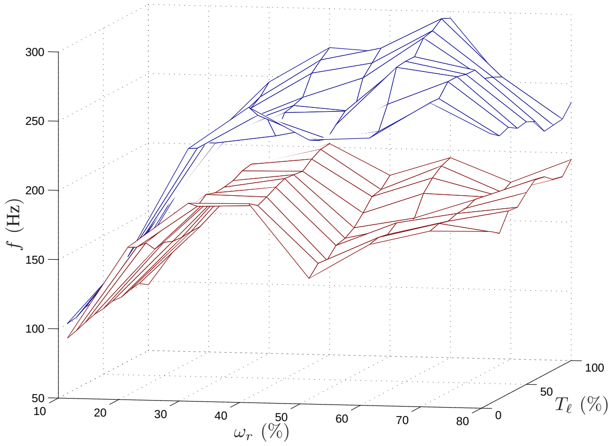


Fig. 3. Comparison of switching frequency f of ABB's DTC (upper surface) with respect to MPC based on extrapolation (lower surface) over the grid of operating points

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 [12], 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.

C. *DTC based on Extrapolation*

The third scheme [13] 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 [14]. Furthermore, the controller needs no tuning parameters.

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 [13].

IV. 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 due 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. [15]).

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 $d = \frac{t_{on}}{T_s}$, 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

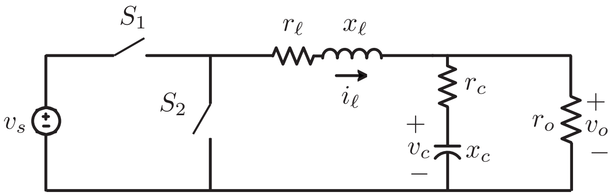


Fig. 4. Topology of the step-down synchronous converter

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 or 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 due to input voltage and output load variations, and model uncertainties.

Motivated by the hybrid nature of dc-dc converters, we have presented in [16], [17] a novel approach to the modelling 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.

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 despite input 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 [17], 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 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 [18].

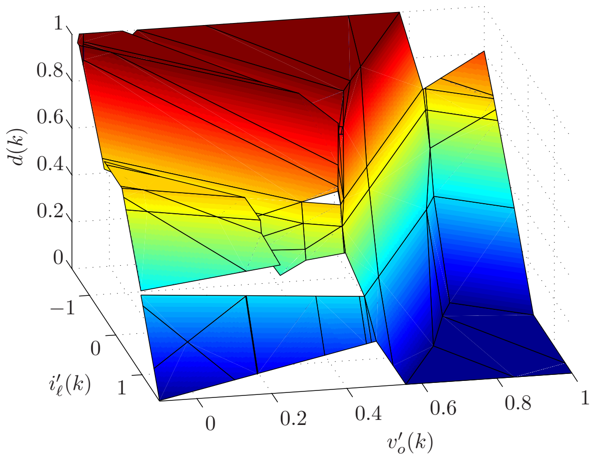


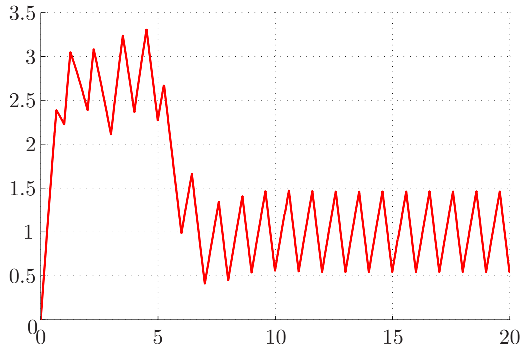
Fig. 5. State-feedback control law: the duty cycle $d(k)$ is given as a PWA function of the transformed state vector; dark blue corresponds to $d(k) = 0$ and dark red to $d(k) = 1$

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.

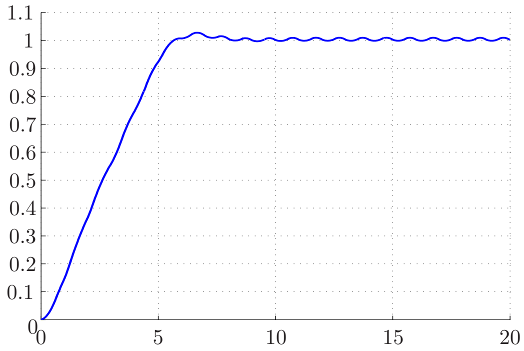
Moreover, an a posteriori analysis shows that the considered state space is a positively invariant set under the derived optimal state-feedback controller. Most importantly, a PieceWise Quadratic (PWQ) Lyapunov function can be computed that proves exponential stability of the closed-loop system for the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

V. CONCLUSIONS AND OUTLOOK

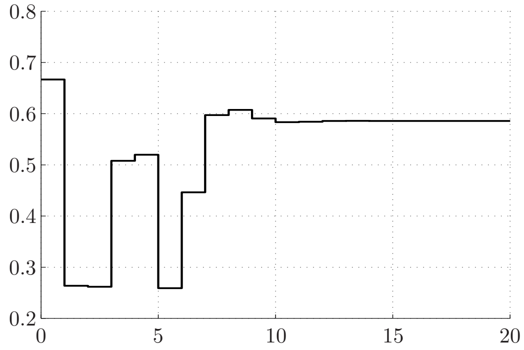
In this paper, we have outlined a number of new approaches to the control of power electronics circuits and systems that have been based on hybrid systems and optimal control methodologies. Two cases have been considered, namely the Direct Torque Control of three-phase induction motors and the optimal control of fixed-frequency dc-dc converters.



(a) Inductor current $i_\ell(t)$



(b) Output voltage $v_o(t)$



(c) Duty cycle $d(t)$

Fig. 6. Closed-loop response during start-up in nominal operation

The analysis has shown that hybrid system methods can be successfully applied to industrially relevant power electronics control problems, bringing benefits in terms of system design and performance. On the other hand, the major issue that arises is the complexity of the developed control algorithms. It is the opinion of the authors, however, that methods tailored to the specific problem under consideration can be developed. This fact, in combination with the continuous increase of the computational power that is available for the control of these systems, enables the control and power electronics communities to revisit some traditionally established methods in a more theoretically rigorous and systematic way.

VI. ACKNOWLEDGEMENTS

This work was supported by the two IST research projects of the European Comission IST-2001-33520 *Control and Computation (CC)* and FP6-IST-511368 *Hybrid Control (HYCON)*. The DTC project was also supported by ABB Switzerland Ltd., and the authors would like to thank Christian Stulz, Pieder Jörg and Petri Schroderus of ABB ATDD, Turgi, Switzerland, and Andreas Poncet of ABB Corporate Research, Baden-Dättwil, Switzerland, for their continuous advice.

REFERENCES

[1] J.M. Maciejowski. *Predictive Control*. Prentice Hall, 2002.
[2] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–427, March 1999.
[3] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automat. Contr.*, 26(2):346–358, April 1981.
[4] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.
[5] A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. In *Proc. American Control Conf.*, pages 1190–1194, Chicago, IL, June 2000.
[6] M. Baotić, F.J. Christophersen, and M. Morari. A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index. In *Proc. of the European Control Conf.*, Cambridge, UK, September 2003.
[7] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *LNCIS*. Springer, 2003.
[8] I. Takahashi and T. Noguchi. A new quick response and high efficiency control strategy for the induction motor. *IEEE Trans. Ind. Applicat.*, 22(2):820–827, September/October 1986.
[9] G. Papafotiou, T. Geyer, and M. Morari. Optimal direct torque control of three-phase symmetric induction motors. Technical Report AUT03-07, Automatic Control Laboratory ETH Zurich, `http://control.ee.ethz.ch`, 2003.
[10] G. Papafotiou, T. Geyer, and M. Morari. Optimal direct torque control of three-phase symmetric induction motors. In *Proc. 43rd IEEE Conf. on Decision and Control*, Atlantis, Bahamas, December 2004.
[11] P.C. Krause. *Analysis of Electric Machinery*. McGraw-Hill, NY, 1986.
[12] T. Geyer and G. Papafotiou. Direct torque control for induction motor drives: A model predictive control approach based on feasibility. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*, pages 274–290. Springer-Verlag, 2005.
[13] T. Geyer, G. Papafotiou, and M. Morari. Verfahren zum Betrieb einer rotierenden elektrischen Maschine. European patent application pending, 2004.
[14] ABB Asea Brown Boveri Ltd. Product webpage of ACS 6000. online document. `www.abb.com/motors&drives`.
[15] N. Mohan, T.M. Undeland, and W.P. Robbins. *Power Electronics: Converters, Applications and Design*. Wiley, 1989.
[16] T. Geyer, G. Papafotiou, and M. Morari. On the optimal control of switch-mode dc-dc converters. In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control*, volume 2993 of *LNCS*, pages 342–356. Springer-Verlag, 2004.
[17] G. Papafotiou, T. Geyer, and M. Morari. Hybrid modelling and optimal control of switch-mode dc-dc converters. In *IEEE Workshop on Computers in Power Electronics (COMPEL)*, Champaing, IL, USA, 2004.
[18] T. Geyer. *Low Complexity Model Predictive Control in Power Electronics and Power Systems*. PhD thesis, Automatic Control Laboratory ETH Zurich, Zurich, 2005.

Hybrid Control Techniques for the Design of Industrial Controllers

Sebastian Engell, Olaf Stursberg
Process Control Laboratory (BCI-AST)
University of Dortmund, 44221 Dortmund, Germany.
Email: {s.engell | o.stursberg}@bci.uni-dortmund.de

***Abstract*— This tutorial paper provides an overview of where techniques based on hybrid dynamic models are suitable or promising for designing controllers of industrial plants, in particular chemical processing systems. After summarizing the typical control tasks prevalent in the hierarchical automation structure of industrial plants, the paper focusses on two techniques employing hybrid models that recently have gained much attention by the research community: the algorithmic verification of safety-related discrete controls, and the optimal control of large transitions, like startup, shutdown, or product switch-over.**

***Index Terms*— Automation, Hybrid Dynamics, Optimal Control, Safety, Supervisory Control, Verification.**

I. INTRODUCTION

While 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 likely 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.

As an example of the complexity encountered, we mention an accident which happened some years ago in the chemical industry in Germany. The operators had forgotten to switch on the stirrer of a reactor while adding a second substance to it. The two substances did not mix well without stirring and the chemical reaction did not start as usual. When the operators realized their mistake (they could monitor this from the reactor temperature) they were aware of the fact that there was a potential for a strong reaction and the generation of a large amount of heat. Hence, in order to increase the transfer of heat to the cooling jacket, they switched the stirrer on. The two substances were mixed when the stirrer was switched on, and the reaction started vigorously, the mixture boiled, and the contents of the reactor contaminated the environment, leading to a large material and immaterial damage to the company.

The second layer of the control system is constituted by continuous regulation loops, e.g. for temperatures, pressures, 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. If there are a large variations of the sequence of operations or of the way in which the steps are performed, as in some chemical or biochemical batch processes, sequence control is mostly performed by the operators. The same is true for the start-up of production processes or for large transitions between operating regimes which usually do not occur too often.

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. While 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.

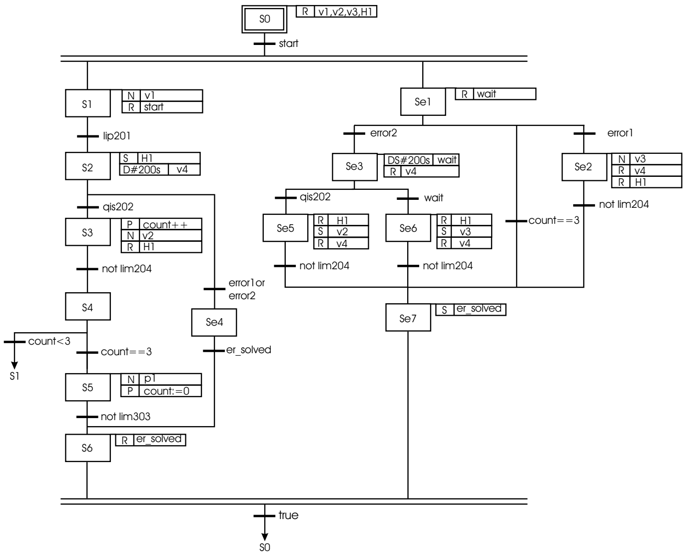


Fig. 1. Supervisory controller as SFC.

In the remainder of this paper, we try to highlight the potential of the application of hybrid systems and control techniques in the area of industrial controls. We focus on the two layers on which the hybrid nature of the controlled plant is most relevant, safety and protection related controls and sequence control. In the latter area, we describe some recent work on one of the most interesting problems, the control of large transitions in processing plants. This topic is most challenging because it requires taking continuous dynamics of considerable complexity into account as well as a large number of discrete and continuous variables over long horizons, rendering brute-force approaches not very promising.

II. 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 (DCS). For the implementation of logic controls, the standard IEC-61131-3 [1] defines several standard formats. Among these, sequential function charts (SFC) 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 [1], SFC consist of alternating sequences of steps and transitions, where actions are associated with steps and conditions with transitions. For an example, Fig. 1 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 which 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.

The goal of the verification of 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 (see e.g. [2]). Before model checking can be applied, the control logic (e.g. an SFC) must be represented as a state transition system. For

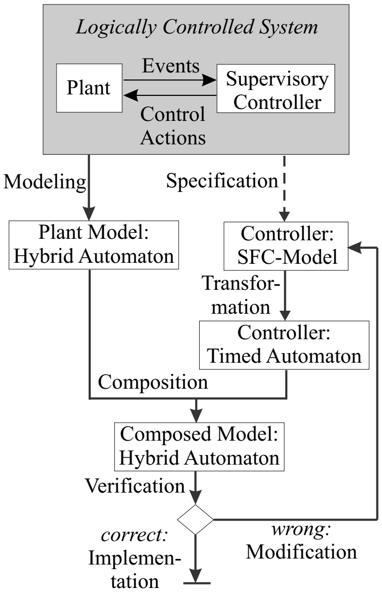


Fig. 2. Control design scheme.

logic control programs that contain timers or delayed actions, timed automata (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 timed automata models, or the tools sketched in [3] for hybrid models. The scheme of the overall procedure is shown in Fig. 2. In the sequel, we discuss the steps of the procedure in more detail for a specific approach that implements this general idea.

A. Transformation of SFC into TA

As proposed in [4], the transformation of a controller given as SFC into a set of timed automata 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 modelled by separate automata, which can include clocks for the case of time-dependent action qualifiers. For modeling the actions, the procedure proposed in [4] uses a scheme that explicitly accounts for the cyclic scanning mode in which SFCs are executed on programmable logic controllers.

B. Model Composition and Verification

In order to simplify the model, the part of the plant which is affected by the safety-related controller should be identified, and the behavior of this part is represented by a suitable model. 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) a hybrid dynamic model, like hybrid automata [5], 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 [6], and [3] 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 method 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 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 fulfil 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.

C. Application to an Evaporation System

In order to illustrate the verification procedure, it is applied to the case study of a batch evaporation system [7]. As shown in Fig. 3, 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 3 batches of the product in T2, and to empty the latter afterwards through P1. Figure 1 shows a possible SFC-controller which not only realizes the

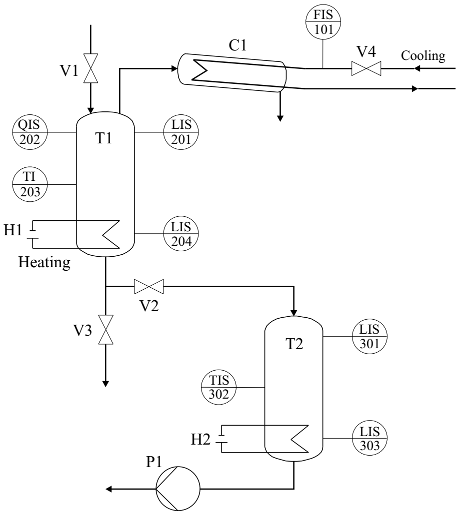


Fig. 3. Flowchart of the evaporation system.

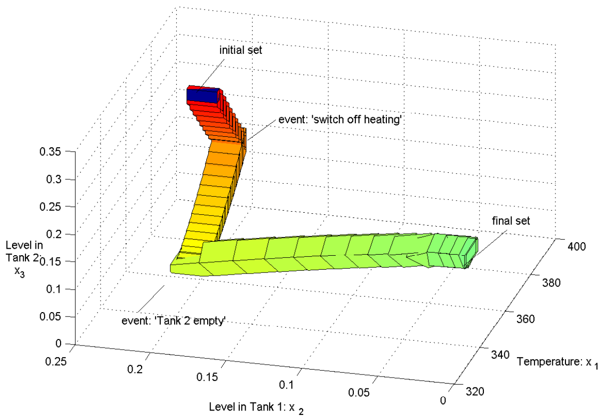


Fig. 5. Reachable continuous set (the final set shows that a critically low temperature ($x_1 = 338K$) is not reached before Tank 1 is empty ($x_2 \leq 0.01m$)) .

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 timed automata following the procedure sketched in Sec. II-A. Figure 4 shows the automata that represent the SFC structure. The complete TA model additionally contains automata that model the actions.

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, P3 - transferring the content of 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. 5 shows, a critically low temperature of 338K is not reached before T1 is emptied, i.e., it can be concluded that the SFC-controller works as desired for this configuration. This result was obtained within a computation time of around one minute on a PC with a 1.8 GHz Pentium-4 CPU.

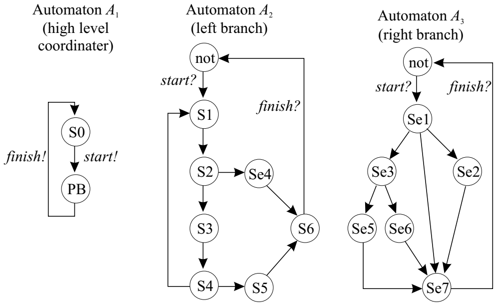


Fig. 4. Separate automata to model the SFC structure (inputs / outputs and time conditions are omitted).

III. OPTIMAL STARTUP AND SHUTDOWN OF INDUSTRIAL PLANTS

While most processing systems are operated by a combination of continuous and discrete controls, both types of controllers are usually designed separately – however, operations like 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. In particular, we consider the aspects of modeling the process dynamics by hybrid automata, formulating the transition procedure as an optimization problem, and computing the (optimal) control inputs efficiently.

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, see e.g. [8], [9], [10], [11], [12], [13]. 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 [14], [15]. If all constraints are written in linear form, mixed-integer linear (or quadratic) programming can be used for the solution, i.e., standard solvers that employ branch-and-bound strategies, where bounds are obtained from linear relaxations, can be used. In [16], it has been shown exemplarily for the approach in [15] that a drawback of this approach is the limited applicability for larger systems. As an alternative, the following section sketches a method with the following characteristics [17], [18]:

(a) the discrete degrees of freedom are determined by a graph search algorithm with problem specific heuristics to determine the optimal discrete control sequence with low effort,

(b) the continuous degrees of freedom are obtained from solving embedded nonlinear programming problems (NLP),
(c) the cost function is evaluated by hybrid simulation which takes care of the state-dependent structural changes of the model.

A. Graph Search with Embedded NLP

Figure 6 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 [17], i.e. characterized by continuous and 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:

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

$$\begin{aligned} s.t. \quad & \phi_\sigma = (\sigma_0, \ldots, \sigma_f) \text{ with: } \sigma_0 = (z_0, x_0), \\ & \sigma_f := (z(t_f), x(t_f)) \in \Sigma_{tar}, \text{ and for } \phi_\sigma \\ & \text{applies in each phase of cont. evolution:} \\ & \sigma \notin F_j \ \forall \ F_j \in F. \end{aligned}$$

where ϕ_u and ϕ_v are the continuous and discrete input trajectories. ϕ_σ is a feasible trajectory of hybrid states $\sigma = (z(t), x(t))$ consisting of a discrete location $z(t)$ and a continuous state $x(t)$ (see [17] for more details). Furthermore, t_f is the final time (with σ_f contained in the target set $\Sigma_{tar} = (z_{tar}, X_{tar})$), and F a collection of sets of forbidden hybrid states F_j. 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.

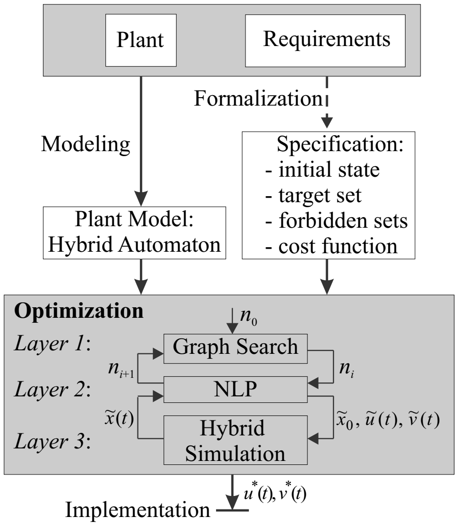


Fig. 6. Scheme for the optimization approach.

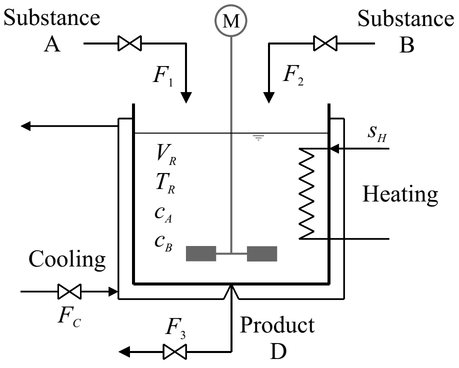


Fig. 7. Scheme of the CSTR.

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 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 nonlinear programming, 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.

B. 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 [15]. The system consists of a tank equipped with two inlets, a heating coil, a cooling jacket, a stirrer, and one outlet (see Fig. III-B). 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 desired temperature range with a high reaction rate, the heating can be switched on (denoted by a discrete variable $s_H \in \{0, 1\}$). The continuously controlled cooling flow F_C serves as a means to remove an excess of heat once the reaction has started. The objective for this system is to determine the input trajectories that drive the initially empty reactor into a desired operation in which the liquid volume V_R, the temperature T_R, and the concentrations c_A and c_B have reached nominal ranges. Additionally, the regions of the state space where $T_R \geq 360$

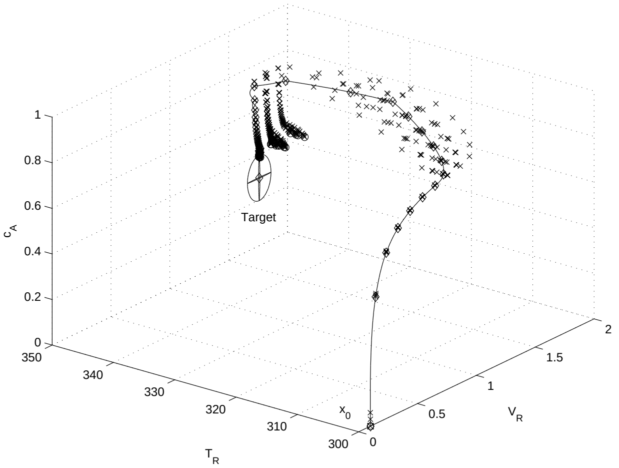


Fig. 8. CSTR: The optimal x-trajectory (solid line) projected in the (V_R, T_R, c_a)-space. Explored nodes are marked by crosses.

or $V_R \geq 1.6$ are forbidden.

To model the system, the state vector is defined as $x := (V_R, T_R, c_A, c_B)^\mathsf{T}$, the continuous input vector as $u := (F_3, F_C)^\mathsf{T}$, and the discrete input vector as $v := (F_1, F_2, s_H)^\mathsf{T}$. 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 $V_R \in]0.8, 2.2]$:

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

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 $x_0 = (0.1, 300, 0, 0)^\mathsf{T}$ and the target is given by z_2 and a hyper-ball with radius 0.1 around the continuous state $x_{tar} = (1.5, 345, 0.4, 0.2)^\mathsf{T}$. 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 III-B shows the state trajectory representing the best solution obtained for a search comprising 400 nodes. This result has been obtained within 2 minutes of computation time on a 2.0 GHz Pentium PC.

IV. CONCLUSIONS

The tasks of verifying properties like safety or goal attainment for industrial plants and of computing optimal control trajectories for procedures like startup or shutdown are two examples where the design procedure can be suitably supported by the use of hybrid models. At the time being, a number of successful applications of such techniques have been reported in literature – however, most of these applications refer to relatively small parts of industrial plants, or systems on a laboratory scale. The following two aspects seem most important to achieve that industrial control engineers include hybrid control techniques into their toolboxes: (a) the awareness of existing hybrid modeling techniques has to be increased, (b) the efficiency of methods for the analysis, design, and optimization of hybrid systems must be further improved to enhance the applicability to industrial-size problems. These are two main objectives of the Network of Excellence *Hybrid Control* (funded by the European Union), which includes one area of activities that explicitly aims at further developing hybrid control techniques based on case-studies provided by (mainly) the processing industries.

REFERENCES

[1] Intern. Electrotechnical Commission (Committee No. 65): “Programmable Controllers – Programming Languages, IEC 61131-3”, Ed. 2.0, 2003.

[2] E.M. Clarke, O. Grumberg, D.A. Peled: “Model Checking”, MIT Press, 1999.

[3] B.I. Silva, O. Stursberg, B.H. Krogh, S. Engell: “An Assessment of the Current Status of Algorithmic Approaches to the Verification of Hybrid Systems”, Proc. IEEE Conf. on Decision and Control, 2001, 2867-2874.

[4] N. Bauer, S. Engell, R. Huuck, S. Lohmann, B. Lukoschus, M.P. Remelhe, O. Stursberg: “Verification of PLC Programs given as Sequential Function Charts”, In: Integration of Software Specif. Techn. for Applic. in Eng., Spinger Series LNCS, Vol. 3147, 2004, 517-540.

[5] T.A. Henzinger: “The Theory of Hybrid Automata”, Proc. 11^{th} IEEE Symp. on Logic in Comp. Science, 1996, 278-292.

[6] E. Clarke, A. Fehnker, Z. Han, B.H. Krogh, J. Ouaknine, O. Stursberg, M. Theobald: “Abstraction and Counterexample-Guided Refinement in Model Checking of Hybrid Systems”, Int. Journal Foundations of Computer Science, Vol. 14(4), 2003, 583-604.

[7] S. Kowalewski, O. Stursberg, N. Bauer: “An Experimental Batch Plant as a Case Example for the Verification of Hybrid Systems”, European Journal of Control, Vol. 7, 2001, 366-381.

[8] M.S. Branicky, V.S. Borkar, S.K. Mitter: “A Unified Framework for Hybrid Control: Model and Optimal Control Theory”, IEEE Trans. Automatic Control, 1998, 43(1), 31-45.

[9] M. Broucke, M.D. Di Benedetto, S. Di Gennaro, A. Sangiovanni-Vincentelli: “Theory of Optimal Control Using Bisimulations”, In: Hybrid Systems: Comp. and Control, Springer, LNCS 1790, 2000, 89-102.

[10] K. Gokbayrak, C. G. Cassandras: “Hybrid Controllers for Hierarchically Decomposed Systems”, In: Hybrid Systems: Comp. and Control, Springer, LNCS 1790, 2000, 117-129.

[11] M.S. Shaikh, P.E. Caines: “On the Optimal Control of Hybrid Systems”, In: Hybrid Systems: Comp. and Control, Springer, LNCS 2623, 2003, 466-481.

[12] M. Buss, O. von Stryk, R. Bulirsch, G. Schmidt: “Towards Hybrid Optimal Control”, Automatisierungstechnik, 9(2000), 448-459.

[13] J. Oldenburg, W. Marquardt, D. Heinz, D. Leineweber: “Mixed Logic Dynamic Optimization Applied to Batch Distillation Process Design”, AICHE Journal, 49(2003), 2900-2917.

[14] A. Bemporad, M. Morari: “Control of Systems Integrating Logic, Dynamics, and Constraints”, Automatica, 35(3), 407-427.

[15] O. Stursberg, S. Panek, J. Till, S. Engell: “Generation of Optimal Control Policies for Systems with Switched Hybrid Dynamics”, In: Modelling, Analysis, and Design of Hybrid Systems, Springer, LNCIS 279, 2002, 337-352.

[16] J. Till, S. Engell, S. Panek, O. Stursberg: “Applied Hybrid System Optimization - An Empirical Investigation of Complexity”, Control Engineering Practice, Vol. 12/10, 2004, 1291-1303.

[17] O. Stursberg: “Dynamic Optimization of Processing Systems with Mixed Degrees of Freedom”, In: Proc. 7th Int. Symposium on Dynamics and Control of Process Systems, 2004, ID: 164.

[18] O. Stursberg: “A Graph-Search Algorithm for Optimal Control of Hybrid Systems”, In: 43rd IEEE Conf. on Decision and Control, 2004, 1412-1417.

Hybrid Systems in Automotive Electronics Design

Andrea Balluchi[*], Luca Benvenuti[*†] and Alberto L. Sangiovanni–Vincentelli[*‡]
[*] PARADES, Via S. Pantaleo, 66, 00186 Roma, Italy, `balluchi,alberto@parades.rm.cnr.it`
[†] DIS, Università di Roma “La Sapienza” Via Eudossiana 18, 00184 Roma, Italy, `luca.benvenuti@uniroma1.it`
[†] EECS Dept., Univ. of California at Berkeley, CA 94720, USA, `alberto@eecs.berkeley.edu`

***Abstract*— Automotive is certainly one of the most attractive and promising application domains for hybrid system techniques. Indeed, some hybrid models and algorithms have already been successfully applied for automotive control designs. On the other hand, despite the significant advances achieved in the past few years, hybrid methods are in general still not mature enough for their effective introduction in the automotive industry design processes at large. In this paper, we take a broad view of the development process for embedded control systems in the automotive industry with the purpose of identifying challenges and opportunities for hybrid systems in the design flow. We identify critical steps in the design flow and extract a number of open problems where, in our opinion, hybrid system technology could play an important role.**

I. INTRODUCTION

Due to the lack of an overall understanding of the interplay of sub–systems and of the difficulties encountered in integrating very complex parts, system integration has become a nightmare in the automotive industry. 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 today environment. The source of these problems is clearly 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 (RTOS). Therefore, the need for standards in the software and hardware domains that will allow plug-and-play of sub-systems 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 functionality 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 functionality 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 design of control algorithms using hybrid system methodologies had been presented in the literature, e.g. cut-off control [6], intake throttle valve control [7], actual engaged gear identification [4], adaptive cruise control [12]. 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 that go beyond the classical controller synthesis step. In this paper, 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 II, 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 III, for each design step, we identify critical phases and bottle-neck problems and we extract relevant open problems that hybrid system technologies may contribute to solve.

This work has been carried out in the framework of the HYCON E.U. Network of Excellence (FP6-IST-511368) and has been partially supported by the CC E.U. Project (FP5-IST-2001-33520).

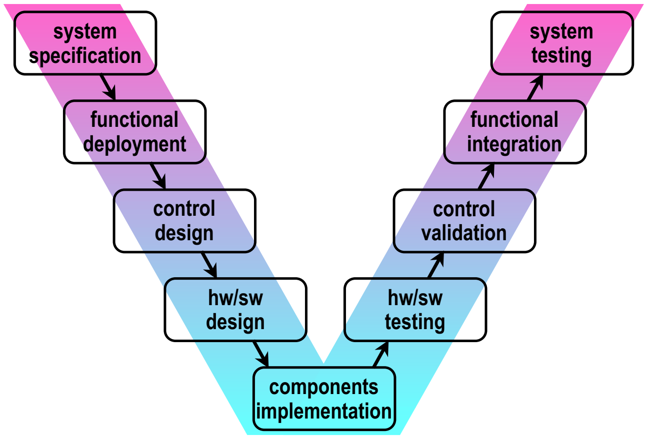


Fig. 1. Design and integration flow.

II. 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: e.g. 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. An ECU (for example, the engine control unit) may have more than one hundred I/O signals, may implement up to two hundreds control algorithms and share with the other related ECUs approximately fifty signals.

The standard design flow of automotive ECUs adopted by Tier–1 companies (subsystem suppliers) is the so–called "V-diagram" shown in Figure 1. The top–down left branch represents the synthesis flow. The bottom–up right branch is the integration and validation flow. The synthesis flow is articulated in the following steps:

A. *System specification*: formalization of system specification; coherence analysis; evaluation of feasibility; completion of under-specified behaviors; abstraction of lower layers customer requirements.
B. *Functional deployment*: system decomposition; definition of subsystem specifications; design of control algorithm architecture; 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, controller validation); fast prototyping.
D. *HW/SW components*: formal specifications for implementation; design of hardware and software architectures; hardware design; software development and automatic code generation; RTOS (Real–time Operating Systems)[1].

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 [1], [5]) and for evaluation of platform requirements at the early stages of the design flow. To do so, design methodologies should provide means for the:

- evaluation of the compliance of the reused component with the new context requirements;
- correct integration with other components;
- cost evaluation.

There is an increasing interest in the industrial community towards managing the complexity of the design and obtaining ECUs with guaranteed performances and reduced cost, by means of a model-based design approach. In this approach, specifications, functional architectures, algorithms, and implementation architectures are represented formally by models thus allowing, at least in principle, formal analysis and automatic synthesis.

III. 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.

A. System specification

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: in fact system specifications define switching conditions between operation modes as well as the desired continuous behavior for each mode.

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 so that the OEM imposes not only a system level requirement but also a particular solution to satisfy it.

Since these constraints are often the result of decisions based on insufficient analysis, the feasible design space may be empty thus causing unnecessary design cycles. We do believe that care must be exercised when constraint are entered at abstraction levels that are non appropriate with respect to the role of the company that specifies them.

The previous discussion shows that:

- tools for system specifications, requirements management and system design, validation and verification must be developed to deal with hybrid models;

---

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

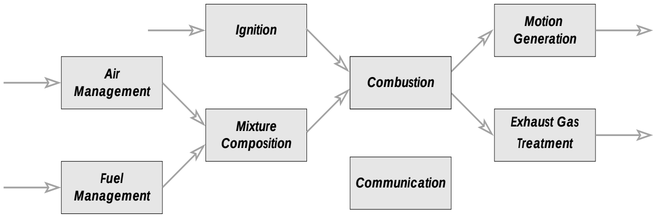


Fig. 2. Functional decomposition.

- since customer requirements contain details regarding several levels of the design flow, then to achieve a complete representation of the system at system specification level, abstraction techniques that deal with hybrid systems for projecting lower–levels specifications back to upper–levels must be developed;
- hybrid techniques and supporting tools to perform coherence and feasibility analysis at system specification level have to be developed as well.

B. Functional deployment

In a first stage of the design, the system is decomposed into a collection of interacting components. The decomposition, based on the understanding of the physical process of interest, 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 [1] for more details). A typical decomposition for engine control is shown in Figure 2. 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, the control algorithms architecture is defined. In particular, the set of control algorithms to be developed for each function and the topology of interconnection are determined. 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. In fact, the OEM often defines also sensors and actuators to be used, since they have a major impact on the cost of the control system. In addition, customer requirements may include details on the topology of the control algorithms architecture that further constrains the functional deployment process.

As a consequence, hybrid formalisms are required to 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 obtained from the functional deployment process.

Moreover, the development of methodologies and tools for the synthesis of functional behaviors from system specifications and for validation of the obtained control algorithm requirements w.r.t. the desired functional behaviors, are necessary.

C. Control system

At the control system level, the algorithms to be implemented in the architecture defined at the functional level 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.

In general, the design process for each control algorithm involves

1. Plant modeling: *a*) model development; *b*) identification; *c*) validation.
2. Controller synthesis: *a*) plant model and specifications analysis; *b*) algorithm development; *c*) controller validation.
3. Fast prototyping.

However, if part of the algorithms are re–used from previous designs, the entire three–step flow is often only partially performed.

In the following sections, the first two steps are discussed in details.

1.a) 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.

To demonstrate that this is indeed the case, let us consider for instance the behavior of an internal combustion engine, and the one of the fuel–injection and spark–ignition subsystems. 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 (which can be represented by a finite-state model) while power–train 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 power–train, which, in turn, depends on the torque produced by each piston. In [2], we showed that the engine can be modeled using a hybrid system composed of interacting finite–state machines, discrete–event systems and continuous–time systems. The hybrid nature of the behaviors is also evident if we look at the different types of input and output signals for the internal combustion engine, and the fuel injection and spark ignition systems. The hybrid nature of the behaviors is not limited to the input–output interfaces of the models. For instance, the model of an automotive drive line has several internal discrete–continuous interactions. In [3], 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 drive line due to engine suspension, clutch, gear, elastic torsional characteristic, tires, frictions and backlashes. Finally, 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, emissions) and are usually represented by piece–wise affine maps.

In conclusion, plant models development requires extensive use of hybrid modeling techniques:

- hybrid deterministic and stochastic formalisms, including FSM, DES, DT, CT, PDA, for representing interacting behaviors of different nature are essential;
- such hybrid formalisms should be supported by appropriate tools for hybrid model description and simulation.

1.b) 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 either obtained analytically (e.g. intake manifold model) 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 hybrid models. Efficient identification techniques for hybrid systems will also give the opportunity for modeling more complex hybrid behaviors that are currently abstracted due to the difficulties in the identification process.

Moreover, efficient hybrid techniques for the representation and identification of nonlinearities, as either piece–wise affine functions (see [9]) or piece–wise polynomial functions, would produce majors impact in the design:

- domain partition could be optimized (possibly not grid-based), achieving increased accuracy and reducing model complexity;
- parameter identification accuracy could be improved;
- higher dimension nonlinearities $R^p \to R$ could be represented and identified.

1.c) 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. These techniques can be used in automotive applications to assess richness of validation patterns for continuous evolutions of the plant. However, the problem remains open for hybrid model validations. This topic is further discussed in Section III.C.2.c, where automatic test pattern generation for controller validation is analyzed.

Validation of hybrid models is a very complex task not sufficiently investigated in the literature. In particular, the following open problems must be addressed:

- methodologies for automatic generation of extensive validation patterns for hybrid models;
- techniques for the assessment of the completeness of validation patterns. This problem can be formalized in the framework of reachability analysis and interesting approaches have been proposed using the concepts of structural coverage and data coverage.

2.a) Plant model and specifications analysis: Typically the design process of a control algorithm for a new application starts with the definition of a plant model based on the analysis of some experimental data obtained either with open–loop control or with some very elementary closed–loop algorithm. The assessment of classical structural properties, such as reachability, observability, stabilizability, passivity [8], on the plant model is of interest in this phase. In addition, quantitative analysis is very useful to understand the strengths and weaknesses of the design. It is interesting to obtain by performance and perturbations/uncertainties analysis an evaluation of quantities such as stability margins, most critical perturbations/uncertainties, robust stability margins, reachability and observability measures in the state space.

Classical concepts and techniques for system analysis cannot be applied to hybrid systems (e.g. switching systems stability has no direct relation with subsystems poles). Unfortunately hybrid system theory has not been developed to a point to be trusted for model analysis:

- some fundamental properties have not been formally defined yet and tests are not available for verifying most of the properties;
- efficient implementation of tests will be necessary for automatic evaluation, since often manual testing is prohibitively expensive for hybrid system properties;
- analysis tools must be integrated with standard system engineering tools.

2.b) 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 at the nominal operation mode or at one of several recovery modes. A significant number of algorithms are dedicated to the computation of switching conditions and controller initializations.

A short and by no–means exhaustive list of control actions for which hybrid system design is particularly interesting is as follows: fuel injection, spark ignition, throttle valve control (especially with stepper motor), electromechanical intake/exhaust valve control, engine start-up and stroke detection, crankshaft sensor management, VGT and EGR actuation (hysteresis management), emission control (cold start-up, lambda on/off sensor feedback), longitudinal oscillations control (backlash and elasticity discontinuities), gear–box control (servo-actuation in traditional gear shift systems), cruise control and adaptive cruise control, diagnosis algorithms (signals and functionalities on-line monitoring), algorithms for fault-tolerance and safety and recovery (degraded mode activation).

Diagnostic algorithms represent a major part of the strategies implemented in automotive ECUs. For engine control, the implementation of diagnosis algorithms is enforced by legislation: OBDII (On Board Diagnosis II) in USA and EOBD (European On Board Diagnosis) in EU. In general, these requirements specify that every fault, malfunction or simple component degradation that leads to pollutant emissions over given thresholds should be diagnosed and signaled to the driver. This requirement has a significant impact on ECU design, since it implies the development of many on–line diagnostic algorithms [11].

Both specifications and accurate models of the plant are often hybrid in automotive applications but the methodology currently adopted for algorithm development is rather crude and can be summarized as follows. 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 controller 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 and corresponding tools for the synthesis of discrete systems are usually not employed. The discrete behavior of the controller is not obtained by automatic synthesis of a formalized specification, as for instance it is done in hardware design. 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;
- reducing implementation cost.

Most of the analytical approaches so far proposed for controller design using hybrid system techniques are quite complex. Usually, the application of these techniques requires designers that are trained in hybrid systems and necessitates long development times. As a consequence, the hybrid system design process results too expensive for the human resources commonly deployed in automotive system engineering. Hence, for a profitable introduction of hybrid system design techniques, it is essential that methodologies are supported by efficient tools that allow fast and easy designs. Hybrid model predictive control is a good example in hybrid system research where the development of the methodology was supported by a good effort in design tool development [10].

2.c) Controller validation: Control algorithms are validated in extensive, time-consuming and hence expensive 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 even if some of the critical maneuvers may be provided by the system specifications. Furthermore, a rough investigation on the robustness properties of control algorithms is obtained by screening the most critical parameters and uncertainties and applying critical perturbations. In the current design flow, there is no automatic approach to the validation of performance specifications. Some approaches for automatic test patterns generation are under investigation. To the best of our knowledge, there is no tool available in the market for performance analysis, robust stability, and formal verification for both continuous and discrete specification.

Due to complexity of the plant–controller interactions, the non negligible effects of the implementation, the large uncertainties in the plant given by product diversity and aging, validation of control algorithms is one of the hottest topics in automotive industry. Today, the quality of the validation step is not satisfactory and important improvements in validation 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. 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 closed–loop models;
  - stability and performance analysis;
  - robust stability and robust performance analysis;
  - invariant set and robust invariant set computations.
- Methodologies and tools should be developed for
  - automatic validation against formalized hybrid performance specifications;
  - automatic validation of safety relevant conditions;
  - automatic optimized test patterns generation reaching specified level of coverage.
- Interesting validation problems are related to the computation of conservative approximations for the largest sets of

  - parameter uncertainties,
  - calibration parameters,
  - implementation parameters (e.g. sampling–period, latency, jitter, computation precision, etc.),

  for which the desired performances are achieved.
- Some classes of algorithms that require intensive and complex validation are
  - diagnosis algorithms;
  - safety critical algorithms;
  - algorithms preventing the system to stall (e.g. idle speed control).

D. Hardware/Software components

The design of HW/SW implementation of ECUs follows today 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 as a more rigorous design approach is advocated for reducing errors. 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 and SW implementation domains are often not integrated; this situation is the frequent cause of design errors. The specification for the HW/SW implementation must include all details necessary for a correct implementation of the algorithms, i.e., they must provide:

- complete description of the algorithm;
- specification of the computation accuracy;
  - in the value domain: precision for each computation chain (for fixed–point arithmetic implementation), threshold detection bounds, etc.);
  - in the time domain: bounds for latency, jitter, delay in event detection, etc.
- execution order and synchronization;
- priorities in case of resource sharing (CPU, communication, etc);
- communication specifications;
- data storage requirements, e.g., variables in EEPROM.

In addition, the specification for the HW/SW implementation should be derived from executable models, according to the model–based design approach. These models should also 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 that can guarantee that the computation accuracy obtained in the HW/SW implementation is good enough. In particular,

- Tools suitable for the description of the implementation requirements of the algorithms have to:
  - support the specification of the algorithm behavior, the computation accuracy and the other implementation requirements and constraints mentioned above;
  - support description of implementation acceptance tests;
  - be efficiently integrated with software and hardware development tools and tools for automatic code generation.
- Methodologies and tools for defining and validating implementation constraints should be developed:
  - the degradation of the execution of control algorithms due to the implementation on bounded resource platforms has to be exported and modeled at the control system level to obtain constraints for the implementation;
  - these constraints should be formally specified in the HW/SW implementation requirements along with executable acceptance tests;
  - tools should support the validation of the HW/SW implementation by running the acceptance tests.

It is in the analysis of the effects of implementation on the behavior of the control algorithms, in the construction of abstracted models of the implementation platform and in the constraint propagation that we see great value in hybrid technology.

IV. CONCLUSIONS

We described critically the automotive electronic design flow in use today 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. While 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. A coherent set of tools and 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 most 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).

V. ACKNOWLEDGMENTS

We wish to thank Alberto Ferrari and Pierpaolo Murrieri of PARADES; Gabriele Serra, Giacomo Gentile and Walter Nesci, of Magneti Marelli Powertrain (Bologna, I); Paolo Ferracin of CNH (Modena, I); Gilberto Burgio of Ford Motor Company (Aachen, G); M.D. DiBenedetto of the University of L'Aquila, for the many discussions on the topic.

REFERENCES

[1] M. Antoniotti, A. Balluchi, L. Benvenuti, A. Ferrari, C. Pinello, A. L. Sangiovanni-Vincentelli, R. Flora, W. Nesci, C. Rossi, G. Serra, and M. Tabaro. A top-down constraints-driven design methodology

for powertrain control system. In *Proc. GPC98, Global Powertrain Congress*, volume Emissions, Testing and Controls, pages 74–84, Detroit, Michigan, USA, October 1998.

[2] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, C. Pinello, and A. L. Sangiovanni-Vincentelli. Automotive engine control and hybrid systems: Challenges and opportunities. *Proceedings of the IEEE*, 88, ”Special Issue on Hybrid Systems” (invited paper)(7):888–912, July 2000.

[3] A. Balluchi, L. Benvenuti, C. Lemma, P. Murrieri, and A. L. Sangiovanni-Vincentelli. Hybrid models of an automotive driveline. Tech. rep., PARADES, Rome, I, December 2004.

[4] A. Balluchi, L. Benvenuti, C. Lemma, A. L. Sangiovanni-Vincentelli, and G. Serra. Actual engaged gear identification: a hybrid observer approach. In *to be presented at 16th IFAC World Congress*, Prague (CZ), July 2005.

[5] A. Balluchi, M. D. Di Benedetto, A. Ferrari, G. Gaviani, G. Girasole, C. Grossi, W. Nesci, M. Pennese, and A. L. Sangiovanni-Vincentelli. Design of a motorcycle engine control unit using an integrated control-implementation approach. In *Proc. 1st IFAC Workshop on ”Advances in Automatic Control”*, pages 218–225, Salerno, Italy, April 2004.

[6] A. Balluchi, M. D. Di Benedetto, C. Pinello, C. Rossi, and A. L. Sangiovanni-Vincentelli. Hybrid control in automotive applications: the cut-off control. *Automatica*, 35, Special Issue on Hybrid Systems:519–535, March 1999.

[7] M. Baotic, M. Vasak, M. Morari, and N. Peric. Hybrid theory based optimal control of electronic throttle. In *Proc. of the IEEE American Control Conference, ACC 2003*, pages 5209–5214, Denver, Colorado, USA, June 2003.

[8] A. Bemporad, G. Bianchini, F. Brogi, and F. Barbagli. Passivity analysis and passification of discrete-time hybrid systems. 2005. Submitted.

[9] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Trans. Automatic Control*, 2004. Accepted for publication as a regular paper.

[10] A. Bemporad, M. Morari, and N. L. Ricker. *Model Predictive Control Toolbox for Matlab – User’s Guide*. The Mathworks, Inc., 2004. http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/.

[11] J. Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Number 3 in Series on Asian Studies in Computer and Information Science. Kluwer International, 1999.

[12] R. M obus, M. Baotic, and M. Morari. Multi-object adaptive cruise control. In O. Maler and Eds. A. Pnueli, editors, *Hybrid Systems: Computation and Control, HSCC 2003*, volume 2623 of *Lecture Notes in Computer Science*, pages 359–374. Springer Verlag, 2003.

On Hybrid Control Problems in Communication Systems

Karl Henrik Johansson and Fortunato Santucci

***Abstract*— The importance of a strong research agenda on the convergence of communications and control has been emphasized by several researchers recently. The purpose of this paper is to evidence through concrete application examples how the envisioned synergy can actually be exploited. The focus is on control of wireless communication networks. In order to address in a rigorous and exhaustive way the complexity of interactions that usually arise in these systems, we try to devise how hybrid modelling may intrinsically provide the theoretical framework to formulate problems and provide partial solutions.**

I. INTRODUCTION

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 [1]. Embedded processors are widely used in, e.g., automotive, entertainment and communication devices, and in a wide range of appliances. On the other side, 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 micro-fabrication 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 recabling. 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. A wireless sensor network can be intended as a reduced version of such systems. While 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 [3], [2]. In fact, traditional control theory typically relies on detailed (accurate) and lossless feedbacks, and time jitter is not considered as well. On the other side, communication networks are designed for applications that typically are either delay tolerant (e.g., data transfer) or error tolerant (e.g., for 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, e.g., frequent and accurate feedbacks are required. In a shared resource environment this may induce larger delays, that might prevent meeting real-time constraints, while contextual information losses might prevent meeting safety constraints. Integrated design of channel coding and control algorithms is discussed in, e.g., [4]. An approach to jointly design control algorithms and the underlying communication network has been recently devised in [5], where the problem has been cast according to a cross-layer paradigm that combines physical layer, media access control (MAC) layer and control application. Modelling the various interacting components is not trivial even in simplified contexts, while 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.

While networked embedded systems are concerned with a communication network to provide service to a control application, 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, that 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

---

The work was partially supported by European Commission through the Network of Excellence HYCON, by the Swedish Foundation for Strategic Research through an Individual Grant for the Advancement of Research Leaders, and by the Swedish Research Council.

K. H. Johansson is with Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, Sweden. `kallej@s3.kth.se`

F. Santucci is with Center of Excellence DEWS, University of L'Aquila, L'Aquila, Italy. `santucci@ing.univaq.it`

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, power levels, coverage handling through efficient beam-forming) that 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 today being far from well understood. 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.

As somehow evidenced, a common need of the two facets depicted above consists in (i) developing sound modelling of complex systems and environments and (ii) subsequently find suitable optimization and control strategies. Specifically, as it will be remarked throughout the examples, hybrid systems theory may intrinsically provide the mathematical basis for modelling the dynamics of our control systems. While the suitability of such models have been proven and exploited recently in, e.g., the automotive domain, only very few and limited attempts (e.g., [6] and [7]) can be found in the technical literature for communication systems and protocols. Therefore, in this paper we intend to emphasize how hybrid dynamics may actually arise in many problems related to operation of communication systems. Specifically, we focus on wireless systems and provide some details on the following problems: power assignment and control in interference-limited fading wireless channels and modelling the behaviour of TCP over a wireless interface.

The outline of the paper is as follows. In Section II we briefly discuss layered architectures of communication and control systems. We review the cross-layer vision of the OSI model and discuss relations to the hybrid systems approach. In Section III we deal with two specific examples, and provide the guidelines for interpreting their hybrid nature. Finally, conclusions and future perspectives are provided in Section IV.

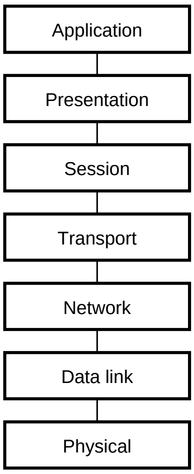


Fig. 1. The OSI model for networking.

II. 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 [11]. 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 a model for hierarchical control.

The OSI reference model is shown in Figure 1, see [8], [9] for details. The model is 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 done 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 Figure 2. The top three OSI layers have been merged into one. The transport layer is based on either the

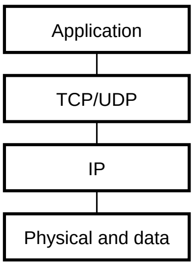


Fig. 2. The layered architecture of the Internet is based on the OSI model.

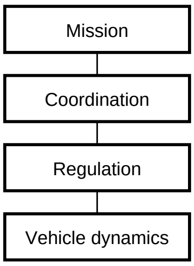


Fig. 3. A layered control architecture for a multi-vehicle control system.

transport control protocol (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 modelling 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 data in reality is 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 even-triggered (discrete) dynamics. For further discussion on such a model for TCP, see [7], where the hybrid nature of TCP itself is also investigated. In Section III, 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 [10]. 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. An example of a layered architecture for a multi-vehicle control system is shown in Figure 3, cf., [11], [12]. The bottom layer consists of the open-loop vehicle dynamics. The second layer is a set of local feedback control laws that regulate the vehicle dynamics, i.e., based on local sensor information provide the vehicle actuators with suitable control commands. The regulation layer provides the coordination layer with a set of maneuvers (e.g., goto way-point, hold maneuver, follow vehicle). The mission layer supervises a team of vehicles by giving each of them sets of maneuvers to execute. The mission layer handles also inter-vehicle communication and error recovery.

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 [11]. As an example, suppose the lowest layer of the architecture in Figure 3 can be modelled as the open-loop system

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x), \end{aligned}$$

where x represents the state of the vehicle (position, heading, etc.), u the controls (steering, throttle, etc.) and y the sensor signals. The regulation layer might be given as

$$u = c_k(y,r),$$

where c_k represents a family of (possibly dynamic) controllers indexed by k, and r reference values and other external variables affecting the controls. Both k and r depend on the maneuver imposed by the coordination layer, e.g., for a goto maneuver c_k could correspond to the implementation of a time-optimal controller and r the way-point. The coordination layer is conveniently modelled as a discrete-event system, for which each state correspond to the execution of a maneuver. Transition takes place either if a maneuver is completed or some other task is given by the mission layer. The integration of the three lower layers of the multi-vehicle control system is hence suitable to model as a hybrid system.

An important extension to the simple hierarchical control model discussed here is the corresponding information and sensing hierarchy. In a networked embedded system, the interaction between control actuation and sensing and information processing is crucial. Under many circumstances, sensing and information processing might be done independent of control (e.g., consider a surveillance robots utilizing a building automation system). This is a conceptually more intrigue system to handle and these are not explicitly captured by the hierarchical control model.

III. ILLUSTRATIVE EXAMPLES

In this section two application examples on wireless communication are presented.

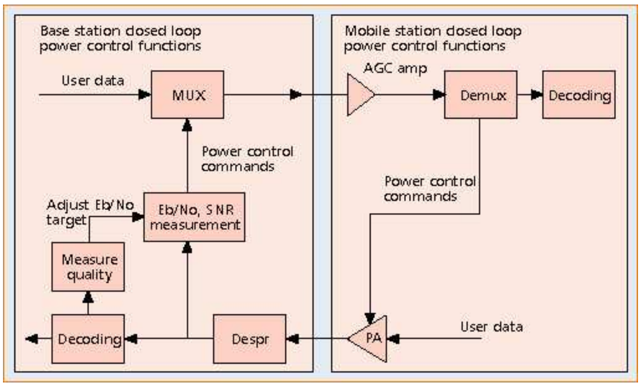


Fig. 4. Power control of third generation wireless system. The closed-loop control system has hybrid dynamics in that there is a mixture of time-triggered and event-triggered signals.

A. Power Control

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 e.g. DS/CDMA, where different 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. 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, that usually consists of an outer loop and an inner loop: they will be sketched in the broader context of next sub-section. However, it is important to remark that the outer loop eventually adapts the target SIR based on link quality estimation. The inner loop is instead responsible for power adaptation at the transmit side in order to meet the required SIR. Let us consider the reverse (Mobile Station (MS) to Base Station (BS)) link in a multi-user system. The closed loop acts for each user signal, so that there is a set of interacting loops, each one acting as follows, see Figure 4. At each symbol time, the received power (or SIR) is e.g. averaged over a block of B symbol intervals (integrate and dump) and compared to the target level. The difference between the filter output and the target level is used to decide which is the power correction to be applied at the MS. A new estimate of the received power is available at the filter output every B bit time intervals. A power update command is then sent on a forward (BS to MS) link power control channel. After a delay, due 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 [13], where it is evidenced that a system with quantized feedback is concerned. We want to emphasize here that the existence of a hybrid dynamics is certainly evident when we remark that 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 can not 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 a hybrid dynamic. Although not explicitly evidenced in the hybrid framework, an attempt to model the complexity of interactions among all these components has been proposed in some recent papers [14]-[15]. In particular, in [15] a model is derived (abstracted) for the power controlled and interference limited wireless channel, and then evaluation of performances of forward error correction (FEC) and hybrid automatic repeat request (ARQ) error control coding is performed over the abstracted channel model.

B. TCP/IP over Wireless Systems

A sound layered communication architecture is important, e.g., [10]. The tremendous growth of the Internet is to a large extent due to the architecture illustrated in Figure 2. 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 Figure 5, 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 in previous sub-section. 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

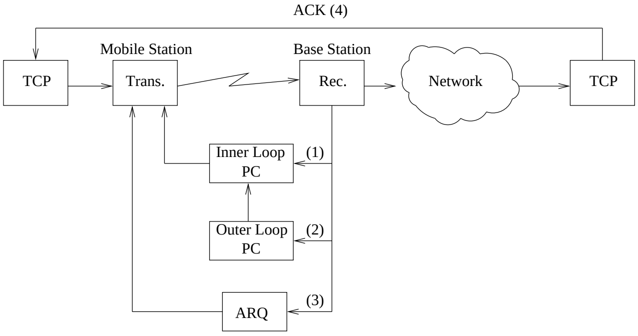


Fig. 5. System overview of wireless Internet in a case when a mobile user connects to an Internet server through a TCP/IP session. Four of the feedback control loops that support the separation of the layers in the network architecture are indicated: the inner power control loop (1), outer power control loop (2), link-layer retransmission (3), and end-to-end congestion control (4).

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 acknowledgement, 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 [19]. It was recently shown that realistically modelled radio links influence the delay distribution of the TCP segments and that they induce spurious timeouts [16]. The performance degradation measured in throughput can be up to about 15%. The analysis is based on a hybrid model derived from Figure 5, where the power control loops are modelled through a Markov chain. The influence of a more detailed radio model was studied in [18].

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, e.g., [20]. 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 done 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 [16]. The fundamental limitations need to be investigated of the system. It was shown in [17] that without any cross-layer signalling, the delay distribution could in a very simple way be adjusted by adding a suitable delay to certain TCP segments and thereby gain considerably improvements of the throughput.

IV. CONCLUSIONS

Through some application examples, we have illustrated the the importance of a research agenda on the convergence between communications and control. Specifically, we have described how some relevant control problems in wireless communications could be usefully cast in terms of hybrid systems for consistent modelling of significant interactions. Current research work is progressing along the two main tracks of control of networks and control over networks, with specific interests on 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 research agenda outlined in the paper forms the basis for a workpackage on control and communication in the EU FP6 Network of Excellence HYCON. Contributions by the partners are gratefully acknowledged. The work presented in Section III is based on joint work with colleagues and students at KTH and DEWS.

REFERENCES

[1] D. Estrin, Ed., *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. U.S. National Research Council, 2001.

[2] S. Graham, G. Baliga, and P. Kumar, “Issues in the converging of control with communication and computing: Proliferation, architecture, design, services, and middleware,” in *Proc. 43rd IEEE Conference on Decision and Control*, 2004.

[3] R. M. Murray, Ed., *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems*. SIAM, 2003.

[4] S. Mitter, “Control with limited information: the role of systems theory and information theory,” *IEEE Information Theory Society Newsletter*, vol. 50, 2000, iSIT 2000 Plenary Talk.

[5] X. Liu and A. Goldsmith, “Wireless network design for distributed control,” in *Proc. 43rd IEEE Conference on Decision and Control*, 2004.

[6] T. Alpcan and T. Basar, “A hybrid systems model for power control in multicell wireless data networks,” *Performance Evaluation*, vol. 57, p. 477, 2004.

[7] J. Hespanha, S. Bohacek, K. Obraczka, and J. Lee, “Hybrid modeling of TCP congestion control,” in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, M. di Benedetto and A. L. Sangiovanni-Vincentelli, Eds. Springer-Verlag, New York, 2001, vol. 2034.

[8] J. D. Day and H. Zimmermann, “The OSI reference model,” *Proceedings of the IEEE*, vol. 71, pp. 1334–1340, Dec. 1983.

[9] J. Walrand and P. Varaiya, *High-performance communication networks*, 2nd ed. Morgan Kaufmann, 2000.

[10] V. Kawadia and P. R. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3–11, 2005.

[11] P. Varaiya, “Smart cars on smart roads: Problems of control,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 195–207, 1993.

[12] J. de Sousa, K. H. Johansson, A. Speranzon, and J. Silva, “A control architecture for multiple submarines in coordinated search missions,” in *IFAC World Congress*, Prague, Czech Republic, 2005, to appear.

[13] F. Gunnarsson and F. Gustafsson, “Power control in wireless communications networks—from a control theory perspective,” in *IFAC World Congress*, 2002.

[14] F. Santucci, G. Durastante, F. Graziosi, and C. Fischione, “Power allocation and control in multimedia cdma wireless systems,” *Kluwer Telecommunication Systems*, vol. 23, pp. 69–94, 2003.

[15] F. Babich, F. Santucci, and F. Graziosi, “Modeling of power-controlled interference-limited wireless channels,” in *WPMC*. IEEE, 2002.

[16] N. Möller and K. H. Johansson, “Influence of power control and link-level retransmissions on wireless TCP,” in *Quality of Future Internet Services*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2003, vol. 2811.

[17] N. Möller, K. H. Johansson, and H. Hjalmarsson, “Making retransmission delays in wireless links friendlier to TCP,” in *Proc. 43rd IEEE Conference on Decision and Control*, 2004.

[18] C. Fischione, F. Graziosi, F. Santucci, N. Möller, K. H. Johansson, and H. Hjalmarsson, “Analysis of TCP over WCDMA wireless systems under power control, MAI and link level error recovery,” in *IWCT*, 2005.

[19] R. Ludwig and R. H. Katz, “The eifel algorithm: Making tcp robust against spurious retransmissions,” *ACM Computer Communication Review*, vol. 30, no. 1, January 2000.

[20] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCP Westwood: bandwidth estimation for enhanced transport over wireless links,” in *MobiCom*, Rome, Italy, 2001.