

Hybrid control of networked embedded systems

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

Abstract

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

1 Introduction

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

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

In this overview paper we start by surveying and classifying the control problems that have been investigated in the hybrid systems literature (Section 2). We then discuss recent developments in four key application areas of hybrid control: control of power electronics (Section 3), industrial process control (Section 4), automotive control (Section 5) and communication systems (Section 6). We conclude the paper with a discussion of the open problems, research challenges and vistas (Section 7).

*Contact author, Laboratoire des Signaux et Systemes, Centre National de la Recherche Scientifique 3 rue Joliot Curie, 91192 Gif-sur-Yvette, France, lamnabhi@lss.supelec.fr

†The work carried out in the framework of the HYCON Network of Excellence, funded by the European Commission, contract number FP6-IST-511368.

2 An overview of hybrid control problems

2.1 Control problem classification

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

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

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

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

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

2.2 A simple hybrid control model

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

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

The dynamics of the state are determined through four functions: a vector field $f : Q \times X \times U \times D \rightarrow X$ that determines the continuous evolution, a reset map $r : Q \times Q \times X \times U \times D \rightarrow X$ that determines the outcome of the discrete transitions, “guard” sets $G : Q \times Q \times \Upsilon \times \Delta \rightarrow 2^X$ that determine when discrete transitions can take place, and “domain” sets $\text{Dom} : Q \times \Upsilon \times \Delta \rightarrow 2^X$ that determines when continuous evolution is possible¹. To avoid pathological situations (lack of solutions, deadlock, chattering, etc.) one needs to introduce technical assumptions on the model components. Typically, these include continuity assumptions on f and r , compactness assumptions on U and D , and convexity assumptions on $\bigcup_{u \in U} f(q, x, u, d)$ and $\bigcup_{d \in D} f(q, x, u, d)$, etc. As for continuous systems, these assumptions aim to ensure that for all $q \in Q$, $x_0 \in X$ and $u(\cdot)$, $d(\cdot)$ measurable functions of time, the differential equation $\dot{x}(t) = f(q, x(t), u(t), d(t))$ has a unique solution $x(\cdot) : \mathbb{R}_+ \rightarrow X$ with $x(0) = x_0$. Additional assumptions are often imposed to prevent deadlock, a situation where it is not possible to proceed by continuous evolution, or by discrete transition. Finally, in many publications assumptions are introduced to prevent what is called the Zeno phenomenon, a situation where the solution of the system takes an infinite number of discrete transitions in a finite amount of time. The Zeno phenomenon can prove particularly problematic for hybrid control problems, since it may be exploited either by the control or by the disturbance variables. For example, a controller may appear to meet a safety specification by forcing all trajectories of the system to be Zeno. This situation is undesirable in practice, since the specifications are met not because of successful controller design but because of modeling over-abstraction.

The solutions of this class of hybrid systems can be defined using the notion of a hybrid time set [39]. A hybrid time set $\tau = \{I_i\}_{i=0}^N$ is a finite or infinite sequence of intervals of the real line, such that for all $i < N$, $I_i = [\tau_i, \tau'_i]$ with $\tau_i \leq \tau'_i = \tau_{i+1}$ and, if $N < \infty$, then either $I_N = [\tau_N, \tau'_N]$, or $I_N = [\tau_N, \tau'_N)$, possibly with $\tau'_N = \infty$. Since the dynamical systems considered here are time invariant, without loss of generality we can assume that $\tau_0 = 0$.

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

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

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

$X \prod_{i=0}^N$, etc. that satisfy the following conditions:

- Discrete evolution: for $i < N$,
 1. $x_i(\tau'_i) \in G(q_i(\tau'_i), q_{i+1}(\tau_{i+1}), v_i(\tau'_i), \delta_i(\tau'_i))$.
 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 to $\dot{x}_i(t) = f(q_i(t), x_i(t), u_i(t), d_i(t))$ over the interval I_i starting at $x_i(\tau_i)$.
 4. $x_i(t) \in \text{Dom}(q_i(t), v_i(t), \delta_i(t))$ for all $t \in [\tau_i, \tau'_i]$.

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

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

An issue that arises is the type of controllers one allows for selecting the control inputs u and v . 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 $g : Q \times X \rightarrow 2^{T \times U}$. For controllers of this type, the runs of the closed-loop system can easily be defined as runs, $(\tau, q, x, v, u, \delta, d)$, of the uncontrolled system such that for all $I_i \in \tau$ and all $t \in I_i$, $(v_i(t), u_i(t)) \in g(q_i(t), x_i(t))$.

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

2.3 Stabilization of hybrid systems

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

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

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

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

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

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

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

Stability of hybrid systems has been extensively studied in recent years (see the overview papers [18, 36]). 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 [33, 53, 61, 64]. Stronger results are possible for special classes of systems, such as planar systems [63]. For non-deterministic systems, in [22] 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 classes of stochastic hybrid systems the reader is referred to [65].

2.4 Optimal control of hybrid systems

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

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

where $l : Q \times X \times U \times D \rightarrow \mathbb{R}$ is a function assigning a cost to the pieces of continuous evolution and $g : Q \times X \times Q \times X \times U \times D \times \Upsilon \times \Delta \rightarrow \mathbb{R}$ 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 [28, 54, 56]. 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 [10, 14]. Computational methods for solving the resulting variational and quasi-variational inequalities are presented in [42]. 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 [11] and the references therein).

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

2.5 Language specification problems

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

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

The name “safety specifications” (which has a formal meaning in computer science) intuitively refers to the fact that such specifications can be used to encode safety requirements in a system, to ensure that nothing bad happens, e.g., ensure that vehicles in an automated highway system (see the discussion in Section 6) do not collide with one another.

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

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

In the automated highways context a minimal liveness type requirement is to make sure that the vehicles eventually arrive at their destination. Mixing different types of specifications 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. The approach was then extended to classes of hybrid systems such as timed automata (systems with continuous dynamics of the form $\dot{x} = 1$, [4]) and rectangular automata (systems with continuous dynamics of the form $\dot{x} \in [l, u]$ for fixed parameters l, u , [62]). For systems of this type, exact and automatic computation of the controllers may be possible using model checking tools [9, 16, 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 often be solved automatically for discrete time systems using methods from mathematical programming (see Section 3 for a discussion).

Extensions to general classes of hybrid systems in continuous time have been concerned primarily with computable numerical approximations of reachable sets using polyhedral approximations [3, 15, 51], ellipsoidal approximations [13], or more general classes of sets. A useful link in this direction has been the relation between reachability problems and optimal control problems with an l_∞ penalty function [58]. 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 [42].

3 Model predictive control in power electronics

3.1 Control problems in power electronics

Power electronics systems represent a well-established technology that has seen significant performance improvements over the last two decades. In general, these systems are used to transform electrical power from one – usually unregulated – form to another regulated one (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.

Recent theoretical advances in the field of hybrid systems, together with the availability of significant computational power for the control loops of power electronics systems, are inviting both the control and the power electronics communities to revisit the control issues associated with power electronics applications. Such an effort for a novel approach to controlling power electronics systems is outlined in this section, where we demonstrate the application of hybrid optimal control methodologies to power electronics systems. More specifically, we show how Model Predictive Control (MPC) [40] can be applied to problems of induction motor drives and dc-dc conversion illustrating the procedure using two examples: the Direct Torque Control (DTC) of three-phase induction motors and the optimal control of fixed-frequency switch-mode dc-dc converters.

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

3.2 Optimal control of discrete time hybrid systems

In the following, we restrict ourselves to the discrete-time domain, and we confine our models to (piece-wise) affine dynamics rather than allowing general nonlinear dynamics. This not only avoids a number of mathematical problems (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) [8] models and the PieceWise Affine (PWA) [55] 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 [30], equivalent to the MLD and PWA forms under mild conditions.

Model Predictive Control (MPC) [40] has been used successfully for a long time in the process industry and recently also for hybrid systems, for which, as shown in [8], 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.

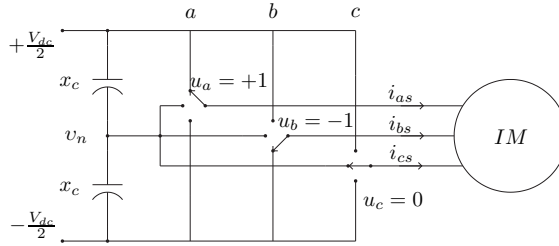


Figure 1: The equivalent representation of a three-phase three-level inverter driving an induction motor

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 [8] and [7]. Details about MPC can be found in [40].

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 [12], 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.

Optimal direct torque control of three-phase induction motors: The rapid development of power semiconductor devices led to the increased use of adjustable speed induction motor drives in a variety of applications. In these systems, dc-ac inverters are used to drive induction motors as variable frequency three-phase voltage or current sources. One methodology for controlling the torque and speed of induction motor drives is Direct Torque Control (DTC) [57], 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.

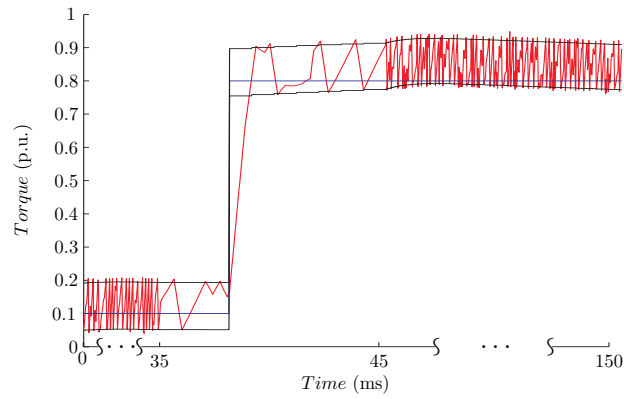
An important first step is to derive discrete-time hybrid models of the drive tailored to our needs – or more specifically, models that are of low complexity yet of sufficient accuracy to serve as prediction models for our model-based control schemes. To achieve this, we have exploited in [48, 50] a 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 [35]. 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 [24, 26, 50] three novel control approaches to tackle the DTC problem, which are inspired by the principles of MPC and tailored to the peculiarities of DTC. For comparing with the industrial state of the art, we have used for all our simulations the Matlab/Simulink model of ABB’s ACS6000 DTC drive [1] containing a squirrel-cage rotor induction motor with a rated apparent power of 2 MVA and a 4.3 kV three-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.

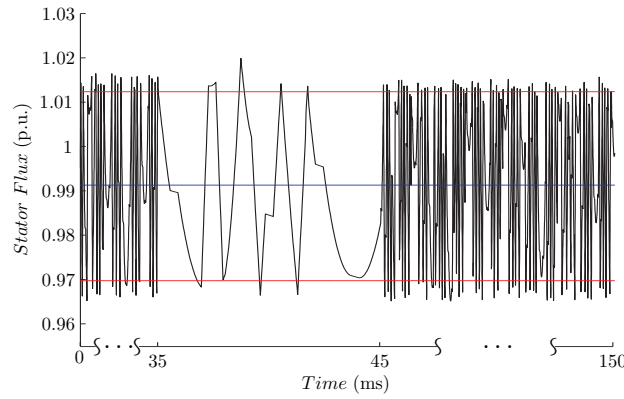
DTC based on priority levels: The first scheme [50] uses soft constraints to model the hysteresis bounds on the torque, stator flux and neutral point potential, and approximates the average switching frequency (over an infinite horizon) by the number of switch transitions over a short horizon. To make this approximation meaningful and to avoid excessive switching, one needs to enforce that switch transitions are only performed if absolutely necessary, i.e. when refraining from switching would lead to a violation of the bounds on the controlled variables within one time-step. This means that the controller has to postpone any scheduled switch transition until absolutely necessary. This strategy can be implemented by imposing a time-decaying penalty on the switch transitions, where switch transitions within the first time-step of the prediction interval result in larger penalties than those that are far in the future. Moreover, three penalty levels with corresponding penalties of different orders of magnitude provide clear controller priorities and make the fine-tuning of the objective function obsolete. To extend the prediction interval without increasing the computational burden, we propose to use a rather long prediction interval, but a short prediction horizon. This is achieved by finely sampling the prediction model with T_s only for the first steps, but more coarsely with a multiple of T_s for steps far in the future. This approach is similar to utilizing the technique of blocking control moves and leads to a time-varying prediction model with different sampling rates.

Simulation results demonstrating the behavior of the controlled variables under this control scheme are presented in Fig. 2. This control scheme not only leads to short commissioning times for DTC drives, but 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 [48].

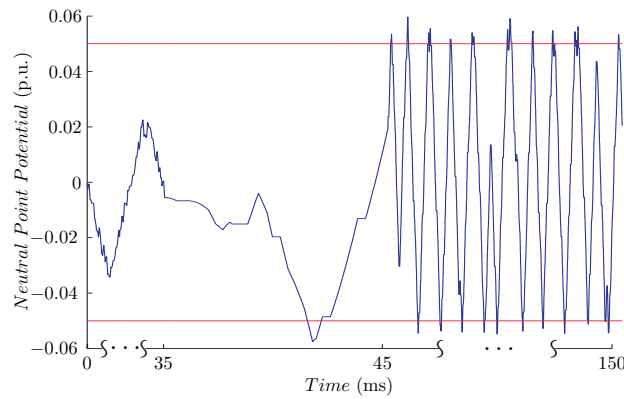
DTC based on feasibility and move blocking: The second scheme, presented in [24], exploits the fact that the control objectives only weakly relate to optimality but rather to feasibility, in the sense that the main objective is to find a control input sequence that keeps the controlled variables within their bounds, i.e. a control input sequence that is feasible. The second, weaker objective is to select among the set of feasible control input sequences the one that minimizes the average switching frequency, which is again approximated by the number of switch transitions over the (short) horizon. We therefore propose an MPC scheme based on feasibility in combination with a move blocking strategy, where we allow for switching only at the current time-step. For each input sequence, we determine the number of steps the controlled variables are kept within their bounds, i.e. remain feasible. The switching frequency is emulated by the cost function, which is defined as the number of switch transitions divided by the number



(a) Electromagnetic torque



(b) Stator flux



(c) Neutral point potential

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

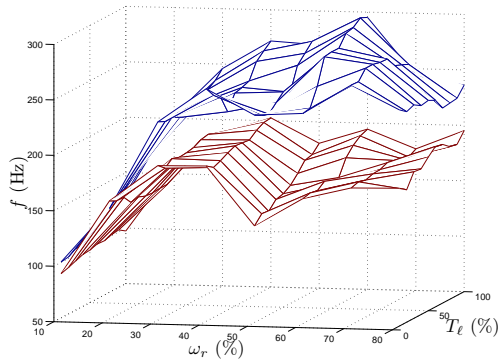


Figure 3: Comparison of switching frequency f of ABB’s DTC (upper surface) with respect to MPC based on extrapolation (lower surface) over the grid of operating points

of predicted time-steps an input remains feasible, and the control input is chosen so that it minimizes this cost function.

As shown in [24], 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.

DTC based on extrapolation:

The third scheme [26] can be interpreted as a combination of the two preceding concepts. Specifically, we use a rather short horizon and compute for the input sequences over the horizon the evolution of the controlled variables using the prediction model. To emulate a long horizon, the “promising” trajectories are extrapolated and the number of steps is determined when the first controlled variable hits a bound. The cost of each input sequence is then determined by dividing the total number of switch transitions in the sequence by the length of the extrapolated trajectory. Minimizing this cost yields the optimal input sequence and the next control input to be applied.

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

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 [26].

3.3 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

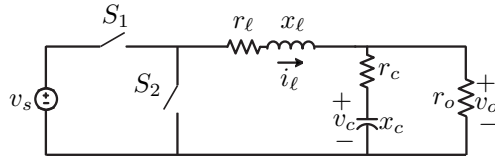


Figure 4: Topology of the step-down synchronous converter

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. [44]).

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 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 [25, 49] 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 [49], 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

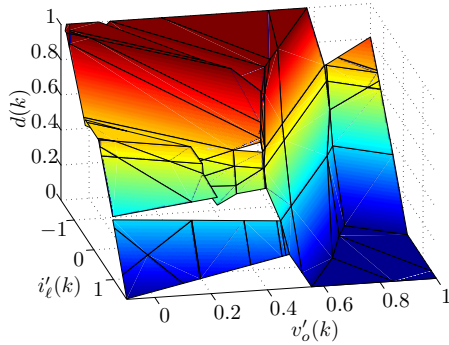


Figure 5: State-feedback control law: the duty cycle $d(k)$ is given as a PWA function of the transformed state vector; dark blue corresponds to $d(k) = 0$ and dark red to $d(k) = 1$

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 [23].

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.

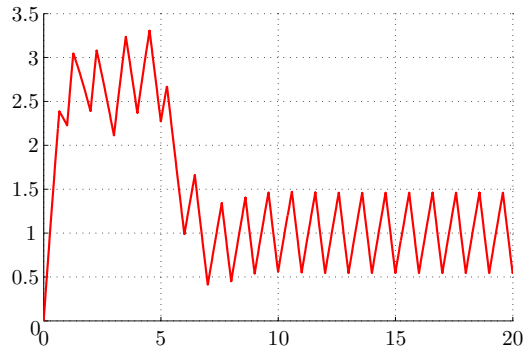
Moreover, an a posteriori analysis shows that the considered state space is a positively invariant set under the derived optimal state-feedback controller. Most importantly, a PieceWise Quadratic (PWQ) Lyapunov function can be computed that proves exponential stability of the closed-loop system for the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

4 Hybrid control for the design industrial controllers

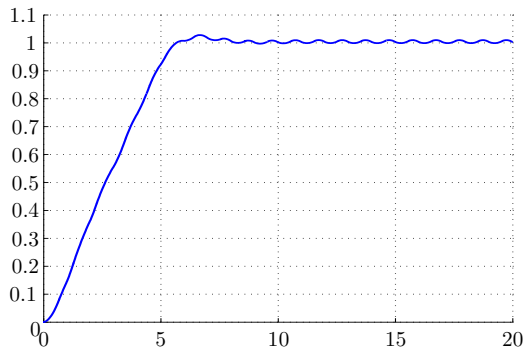
4.1 Hybrid control issues in industrial processes

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.

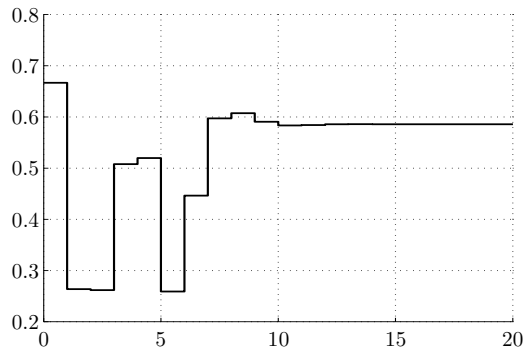
Generally, several layers of industrial control systems can be distinguished. The first and lowest layer



(a) Inductor current $i_l(t)$



(b) Output voltage $v_o(t)$



(c) Duty cycle $d(t)$

Figure 6: Closed-loop response during start-up in nominal operation

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

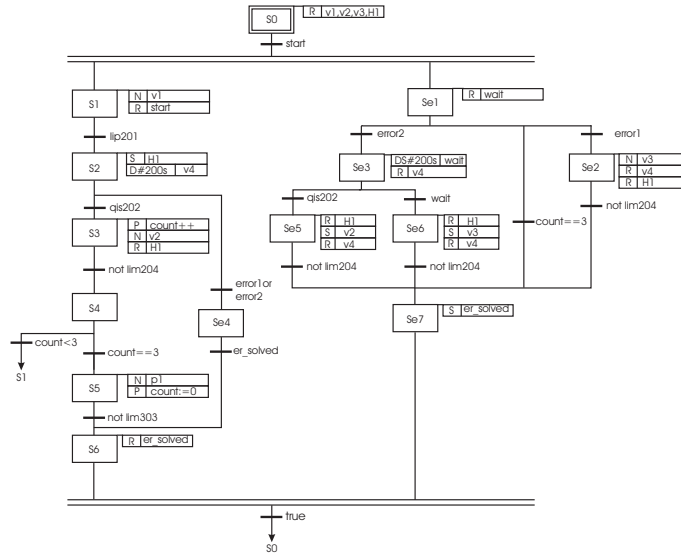


Figure 7: Supervisory controller as SFC.

the plant model, and then tested, debugged and modified during start-up of the plant. The main reason that this approach does not lead to complete failure is that for the most part logic control software from other projects is re-used and only small modifications and extensions are added. However, taking into account the low-level programming languages used and the lack of formal documentation, such software systems may become harder and harder to maintain.

4.2 Verification of safety related logic controllers

In order to be accepted by practitioners, verification procedures for safety and protection related industrial controllers must be able to handle the control logic as it is implemented on the control hardware, usually a programmable logic controller (PLC) or a distributed control system (DCS). For the implementation of logic controls, the standard IEC-61131-3 [?] 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 [?], SFC consist of alternating sequences of steps and transitions, where actions are associated with steps and conditions with transitions. For an example, Fig. 7 shows the graphical representation of SFC, in which rectangles denote the steps (with action 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. [?]). Before model checking can be applied, the control logic (e.g. an SFC) must be represented as a state transition system. For logic control programs that contain timers or delayed actions, 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

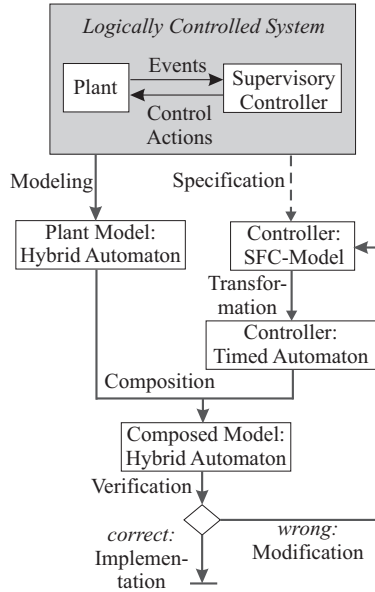


Figure 8: Control design scheme.

purely discrete models, UPPAAL for timed automata models, or the tools sketched in [?] for hybrid models. The scheme of the overall procedure is shown in Fig. 8. In the sequel, we discuss the steps of the procedure in more detail for a specific approach that implements this general idea.

Transformation of SFC into TA: As proposed in [?], 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 [?] uses a scheme that explicitly accounts for the cyclic scanning mode in which SFCs are executed on programmable logic controllers.

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 [?], 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 [?], and [?] 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.

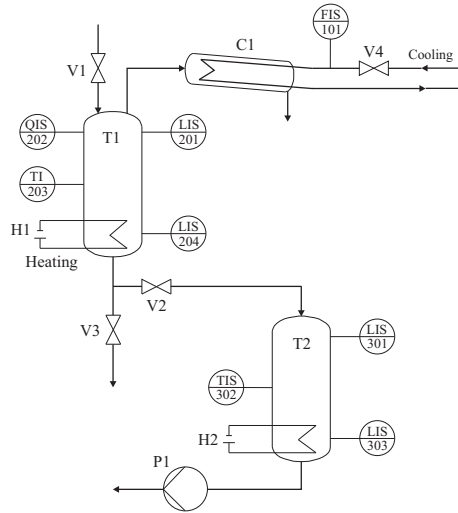


Figure 9: Flowchart of the evaporation system.

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.

Application to an evaporation system: In order to illustrate the verification procedure, it is applied to the case study of a batch evaporation system [?]. As shown in Fig. 9, the system consists of two tanks (T1, T2) with heating devices, a condenser with cooling (C1), connecting pipes with valves (V1, V2, V3) and a pump (P1), as well as different sensors for liquid levels (LIS), temperatures (TI), and concentration (QIS). The intended operation is to evaporate the liquid from a mixture in T1 until a desired concentration is reached, to collect 3 batches of the product in T2, and to empty the latter afterwards through P1. Figure 7 shows a possible SFC-controller which not only realizes the desired procedure (left branch) but also includes exception routines (right branch) for the cases of evaporator breakdown (error1) and malfunction of the heating device of T1 (error2).

Since the SFC-controller contains two time-dependent actions (marked by 'D#200s' and 'DS#200s'), it is transformed into a set of timed automata following the procedure sketched in Sec. ???. Figure 10 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. 11 shows, a critically low

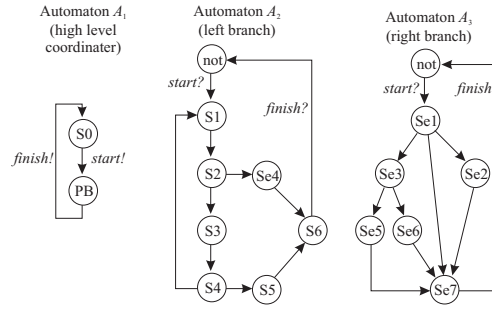


Figure 10: Separate automata to model the SFC structure (inputs / outputs and time conditions are omitted).

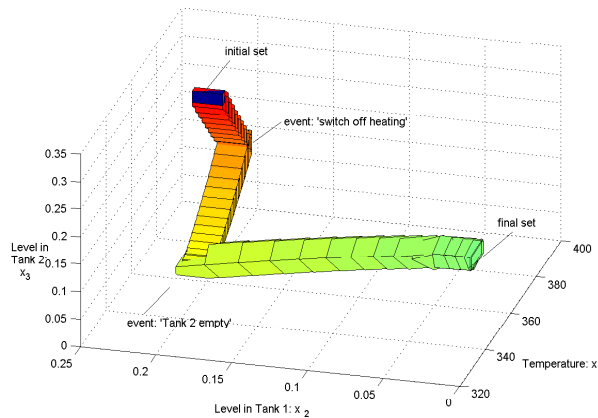


Figure 11: Reachable continuous set (the final set shows that a critically low temperature ($x_1 = 338K$) is not reached before Tank 1 is empty ($x_2 \leq 0.01m$)).

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.

4.3 Optimal start-up and shut-down 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. [14, ?, ?, ?, ?, ?]. 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 [?, ?]. 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 [?], it has been shown exemplarily for the approach in [?] that a drawback of this approach is the limited applicability for larger systems. As an alternative, the following section

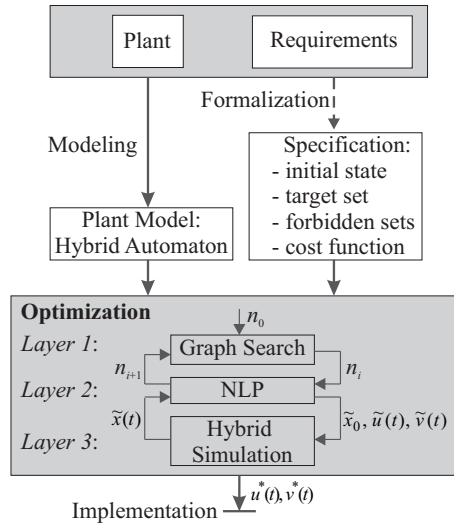


Figure 12: Scheme for the optimization approach.

sketches a method with the following characteristics [?, ?]:

- (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.

Graph search with embedded NLP: Figure 12 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 [?], 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:

$$\begin{aligned}
 & \min_{\phi_u \in \Phi_u, \phi_v \in \Phi_v} \Omega(t_f, \phi_\sigma, \phi_u, \phi_v) & (1) \\
 \text{s.t. } & \phi_\sigma = (\sigma_0, \dots, \sigma_f) \text{ with: } \sigma_0 = (z_0, x_0), \\
 & \sigma_f := (z(t_f), x(t_f)) \in \Sigma_{tar}, \text{ and for } \phi_\sigma \\
 & \text{applies in each phase of cont. evolution:} \\
 & \sigma \notin F_j \forall F_j \in F.
 \end{aligned}$$

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 [?] 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 .

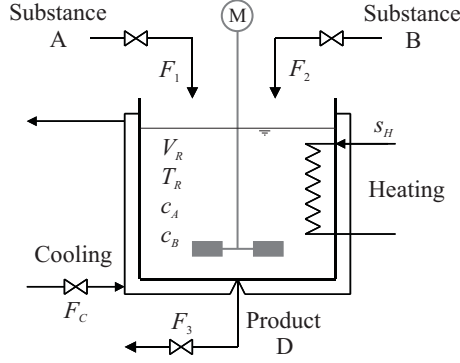


Figure 13: Scheme of the CSTR.

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.

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 [?]. The system consists of a tank equipped with two inlets, a heating coil, a cooling jacket, a stirrer, and one outlet (see Fig. 4.3). 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 $T_R \geq 360$ or $V_R \geq 1.6$ are forbidden.

To model the system, the state vector is defined as $x := (V_R, T_R, c_A, c_B)^T$, the continuous input vector as $u := (F_3, F_C)^T$, and the discrete input vector as $v := (F_1, F_2, s_H)^T$. Depending on the continuous state,

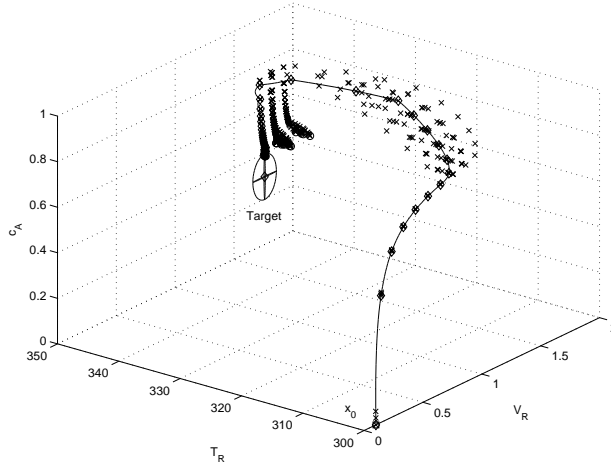


Figure 14: CSTR: The optimal x -trajectory (solid line) projected in the (V_R, T_R, c_a) -space. Explored nodes are marked by crosses.

the system dynamics can be written as $\dot{x} = f(z, x, u, v)$ where:

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

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

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

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

and $q = c_A c_B^2 \exp(-k_5/T_R)$. The separation into two V_R -regions accounts for the fact that the heating is only effective above $V_R = 0.8$. The initial state is $x_0 = (0.1, 300, 0, 0)^T$ and the target is given by z_2 and a hyper-ball with radius 0.1 around the continuous state $x_{tar} = (1.5, 345, 0.4, 0.2)^T$. The optimization was run with the cost criterion that the transition time for the startup procedure is minimized. The strategy chosen is that depth-first search is used until a first solution is found, then a breadth-first strategy is applied. Figure 4.3 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.

5 Hybrid systems in automotive electronics design

6 Hybrid control in communication systems

6.1 Hybrid control issues in communication systems

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 [20]. 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 re-cabling. In summary, we can envisage a networked embedded system as an eventually large set of sensors, controllers and actuators linked via wired and wireless communication channels. 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 [27, 47]. 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., [43]. An approach to jointly design control algorithms and the underlying communication network has been recently devised in [37], 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 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., [2] and [32]) can be found in the technical literature for communication systems and protocols. Therefore, in this section 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.

6.2 Layered architectures for networked systems

In the design of large-scale systems, it is crucial to have a design approach based on composition and modularity. This helps the designer to argue about the system and understand interactions and dynamics. Layered system architectures are common in many disciplines and widely used in practice. It is surprising that there is not much theory that supports the use [59]. 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 15, see [17, 60] 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 16. The top three OSI layers have been merged into one. The transport layer is based on either the 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

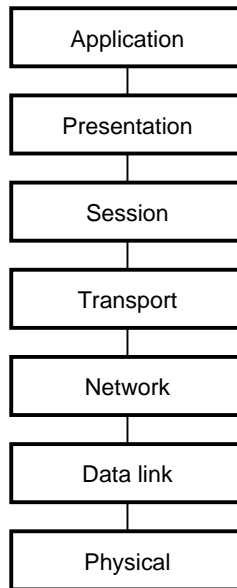


Figure 15: The OSI model for networking.

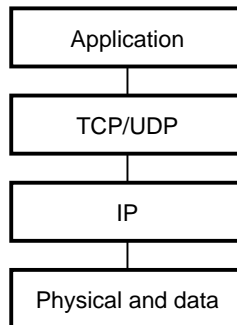


Figure 16: The layered architecture of the Internet is based on the OSI model.

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 event-triggered (discrete) dynamics. For further discussion on such a model for TCP, see [32], where the hybrid nature of TCP itself is also investigated. In Section 6.3, 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 [34]. 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 17, cf., [19, 59]. 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

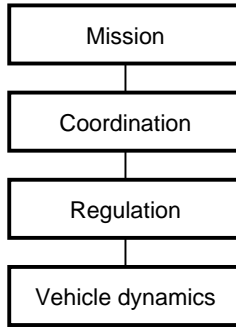


Figure 17: A layered control architecture for a multi-vehicle control system.

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 [59]. As an example, suppose the lowest layer of the architecture in Figure 17 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.

6.3 Illustrative examples

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

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

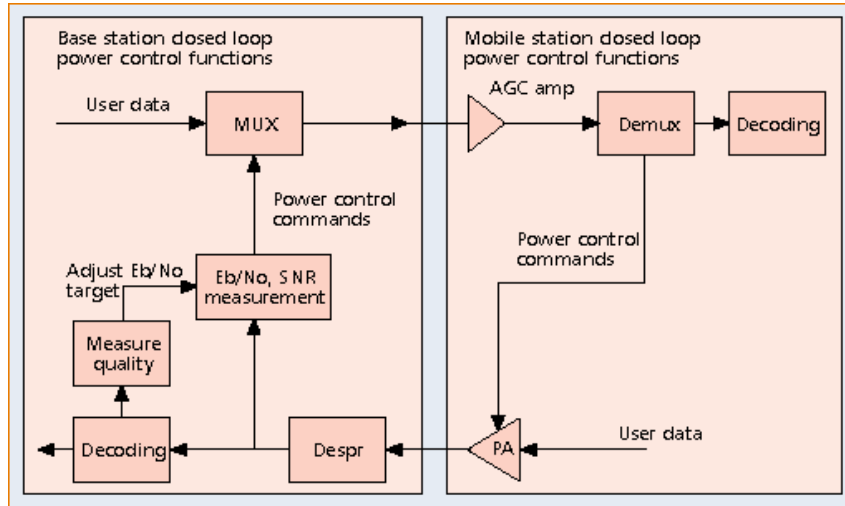


Figure 18: Power control of third generation wireless system. The closed-loop control system has hybrid dynamics in that there is a mixture of time-triggered and event-triggered signals.

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 18.

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 [29], 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 [52]-[5]. In particular, in [5] 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

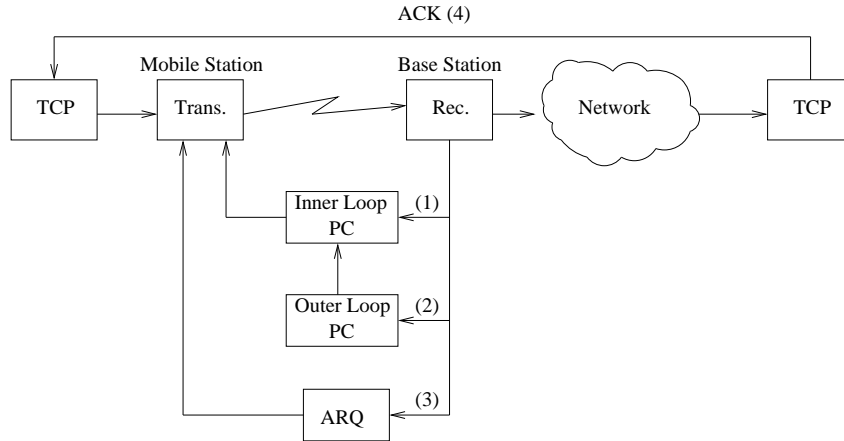


Figure 19: System overview of wireless Internet in a case when a mobile user connects to an Internet server through a TCP/IP session. Four of the feedback control loops that support the separation of the layers in the network architecture are indicated: the inner power control loop (1), outer power control loop (2), link-layer retransmission (3), and end-to-end congestion control loop (4).

abstracted channel model.

TCP/IP over wireless systems: A sound layered communication architecture is important, e.g., [34]. The tremendous growth of the Internet is to a large extent due to the architecture illustrated in Figure 16. 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 19, 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 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 [38]. It was recently shown that realistically modelled radio links influence the delay distribution of the TCP segments and that they induce spurious timeouts [45]. The performance degradation measured in throughput can be up to about 15%. The analysis is based on a hybrid model derived from Figure 19, where the power control loops are modelled through a Markov chain. The influence of a more detailed radio model was studied in [21].

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., [41]. 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 [45]. The fundamental limitations need to be investigated of the system. It was shown in [46] 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 considerable improvements of the throughput.

7 Concluding remarks

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

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

In terms of application to power electronics and power systems in general, methods tailored to the specific problems that arise in this area have to be developed. Progress in this direction, in combination with the continuous increase of 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.

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

In terms of communication networks, current research work is progressing along the two main tracks of control of networks and control over networks presented in Section ???. Specific interests include various aspects of distributed radio resource management in evolved third generation wireless systems, and efficient design and operations of ad-hoc wireless networks for control applications.

References

- [1] ABB Asea Brown Boveri Ltd. Product webpage of ACS 6000. online document. www.abb.com/motors&drives.
- [2] T. Alpcan and T. Basar. A hybrid systems model for power control in multicell wireless data networks. *Performance Evaluation*, 57:477, 2004.
- [3] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88(7):1011–1025, July 2000.
- [4] E. Asarin, O. Maler, and A. Pnueli. Symbolic controller synthesis for discrete and timed systems. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Proceedings of Hybrid Systems II*, number 999 in LNCS, pages 1–20. Springer-Verlag, Berlin, 1995.
- [5] F. Babich, F. Santucci, and F. Graziosi. Modeling of power-controlled interference-limited wireless channels. In *WPMC*. IEEE, 2002.
- [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. Cambridge, UK, September 2003.
- [7] A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. pages 1190–1194, Chicago, IL, June 2000.
- [8] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. 35(3):407–427, March 1999.
- [9] J. Bengtsson, K.G. Larsen, F. Larsson, P. Pettersson, and W. Yi. UPAAL: A tool suit for automatic verification of real-time systems. In *Hybrid Systems III*, number 1066 in LNCS, pages 232–243. Springer-Verlag, Berlin, 1996.
- [10] A. Bensoussan and J.L. Menaldi. Hybrid control and dynamic programming. *Dynamics of Continuous, Discrete and Impulsive Systems*, (3):395–442, 1997.
- [11] G. Berhmann, , A. Fehnker, T. Hune, K.G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum cost reachability for priced timed automata. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in LNCS, pages 147–161. Springer-Verlag, Berlin, 2001.
- [12] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer, 2003.
- [13] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In Nancy Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control*, number 1790 in LNCS, pages 73–88. Springer-Verlag, Berlin, 2000.
- [14] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: Model and optimal control theory. 43(1):31–45, 1998.
- [15] A. Chutinam and B. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In Frits W. Vaandrager and Jan H. van Schuppen, editors, *Hybrid Systems: Computation and Control*, number 1569 in LNCS, pages 76–90. Springer-Verlag, Berlin, 1999.
- [16] C. Daws, A. Olivero, S. Trypakis, and S. Yovine. The tool KRONOS. In R. Alur, T. Henzinger, and E. Sontag, editors, *Hybrid Systems III*, number 1066 in LNCS, pages 208–219. Springer-Verlag, Berlin, 1996.
- [17] J. D. Day and H. Zimmermann. The OSI reference model. *Proceedings of the IEEE*, 71:1334–1340, December 1983.

- [18] R. De Carlo, M. Branicky, S. Pettersson, and B. Lennarston. Perspectives and results on the stability and stabilizability of hybrid systems. 88(7):1069–1082, July 2000.
- [19] 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.
- [20] D. Estrin, editor. *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. U.S. National Research Council, 2001.
- [21] 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.
- [22] 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*, 77(16):1393–1407, 2004.
- [23] T. Geyer. *Low Complexity Model Predictive Control in Power Electronics and Power Systems*. PhD thesis, Zurich, 2005.
- [24] T. Geyer and G. Papafotiou. Direct torque control for induction motor drives: A model predictive control approach based on feasibility. volume 3414 of *Lecture Notes in Computer Science*, pages 274–290. Springer-Verlag, 2005.
- [25] T. Geyer, G. Papafotiou, and M. Morari. On the optimal control of switch-mode dc-dc converters. volume 2993 of *Lecture Notes in Computer Science*, pages 342–356. Springer-Verlag, 2004.
- [26] T. Geyer, G. Papafotiou, and M. Morari. Verfahren zum Betrieb einer rotierenden elektrischen Maschine. European patent application pending, 2004.
- [27] S. Graham, G. Baliga, and P.R. 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.
- [28] G. Grammel. Maximum principle for a hybrid system via singular perturbations. 37(4):1162–1175, 1999.
- [29] F. Gunnarsson and F. Gustafsson. Power control in wireless communications networks—from a control theory perspective. In *IFAC World Congress*, 2002.
- [30] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. 37(7):1085–1091, July 2001.
- [31] T. A. Henzinger, P. H. Ho, and H. Wong Toi. A user guide to HYTECH. In E. Brinksma, W. Cleveland, K. Larsen, T. Margaria, and B. Steffen, editors, *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, number 1019 in LNCS, pages 41–71, Berlin, 1995. Springer-Verlag.
- [32] J. Hespanha, S. Bohacek, K. Obraczka, and J. Lee. Hybrid modeling of TCP congestion control. In M. di Benedetto and A. L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 2001.
- [33] J. Hespanha and A.S. Morse. Switching between stabilizing controllers. *Automatica*, 38(11):1905–1917, November 2002.
- [34] V. Kawadia and P. R. Kumar. A cautionary perspective on cross-layer design. *IEEE Wireless Communications*, 12(1):3–11, 2005.
- [35] P.C. Krause. *Analysis of Electric Machinery*. McGraw-Hill, NY, 1986.

- [36] D. Liberzon and A.S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19:59–70, October 1999.
- [37] X. Liu and A. Goldsmith. Wireless network design for distributed control. In *Proc. 43rd IEEE Conference on Decision and Control*, 2004.
- [38] R. Ludwig and R. H. Katz. The eifel algorithm: Making tcp robust against spurious retransmissions. *ACM Computer Communication Review*, 30(1), January 2000.
- [39] J. Lygeros, K.H. Johansson, S.N. Simić, J. Zhang, and S.S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2–17, January 2003.
- [40] J.M. Maciejowski. *Predictive Control*. Prentice Hall, 2002.
- [41] 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.
- [42] I. Mitchell, A.M. Bayen, and C.J. Tomlin. Validating a Hamilton–Jacobi approximation to hybrid system reachable sets. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, number 2034 in LNCS, pages 418–432. Springer-Verlag, Berlin, 2001.
- [43] S. Mitter. Control with limited information: the role of systems theory and information theory. *IEEE Information Theory Society Newsletter*, 50, 2000. ISIT 2000 Plenary Talk.
- [44] N. Mohan, T.M. Undeland, and W.P. Robbins. *Power Electronics: Converters, Applications and Design*. Wiley, 1989.
- [45] N. Möller and K. H. Johansson. Influence of power control and link-level retransmissions on wireless TCP. In *Quality of Future Internet Services*, volume 2811 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.
- [46] 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.
- [47] R. M. Murray, editor. *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems*. SIAM, 2003.
- [48] G. Papafotiou, T. Geyer, and M. Morari. Optimal direct torque control of three-phase symmetric induction motors. Technical Report AUT03-07, <http://control.ee.ethz.ch>, 2003.
- [49] 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)*, Champaign, IL, USA, 2004.
- [50] 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.
- [51] P. Saint-Pierre. Approximation of viability kernels and capture basins for hybrid systems. In *European Control Conference*, pages 2776–2783, Porto, September 4-7, 2001.
- [52] F. Santucci, G. Durastante, F. Graziosi, and C. Fischione. Power allocation and control in multimedia cdma wireless systems. *Kluwer Telecommunication Systems*, 23:69–94, 2003.
- [53] A. V. Savkin, E. Skafidas, and R.J. Evans. Robust output feedback stabilizability via controller switching. *Automatica*, 35(1):69–74, 1999.
- [54] M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: optimal trajectories, switching times and location schedules. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, number 2623 in LNCS, pages 466–481. Springer-Verlag, Berlin, 2003.

- [55] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automat. Contr.*, 26(2):346–358, April 1981.
- [56] Hector J. Sussmann. A maximum principle for hybrid optimal control problems. pages 425–430, Phoenix, Arizona, December 7–10, 1999.
- [57] 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.
- [58] C.J. Tomlin, J. Lygeros, and S.S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–969, July 2000.
- [59] P. Varaiya. Smart cars on smart roads: Problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.
- [60] J. Walrand and P. Varaiya. *High-performance communication networks*. Morgan Kaufmann, 2nd edition, 2000.
- [61] 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*, 4:140–147, 1998.
- [62] H. Wong-Toi. The synthesis of controllers for linear hybrid automata. pages 4607–4613, San Diego, California, December 10–12 1997.
- [63] X. Xu and P.J. Antsaklis. Stabilization of second order LTI switched systems. *International Journal of Control*, 73(14):1261–1279, September 2000.
- [64] X. Xu and G. Zhai. On practical stability and stabilization of hybrid and switched systems. In R. Alur and G.J. Pappas, editors, *Hybrid Systems Computation and Control*, number 2993 in LNCS, pages 615–630. Springer-Verlag, Berlin, 2004.
- [65] 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.