Automotive Engine Control and Hybrid Systems: Challenges and Opportunities

ANDREA BALLUCHI, LUCA BENVENUTI,
MARIA DOMENICA DI BENEDETTO, SENIOR MEMBER, IEEE, CLAUDIO PINELLO, AND
ALBERTO LUIGI SANGIOVANNI-VINCENTELLI, FELLOW, IEEE

Invited Paper

The design of engine control systems has been traditionally carried out using a mix of heuristic techniques validated by simulation and prototyping using approximate average-value models. However, the ever increasing demands on passengers' comfort, safety, emissions, and fuel consumption imposed by car manufacturers and regulations call for more robust techniques and the use of cycle-accurate models. We argue that these models must be hybrid because of the combination of time-domain and event-based behaviors. In this paper, we present a hybrid model of the engine in which both continuous and discrete time-domain as well as event-based phenomena are modeled in a separate but integrated manner. Based on this model, we formalize the specification of the overall engine control by defining a number of hybrid control problems. To cope with the difficulties arising in the design of hybrid controllers, a design methodology is proposed. This methodology consists of a relaxation of the hybrid problem by simplifying some of its components to obtain a solvable problem, and then deriving a solution to the original control problem by appropriately modifying the control law so obtained to take into consideration the original specifications and models. The effectiveness of this approach is illustrated on three challenging problems: fast force-transient control, cutoff control, and idle speed control.

Keywords*—*Engine control, hybrid dynamical systems, modeling, models of computation.

Manuscript received November 8, 1999; revised April 13, 2000. This work was supported in part by the Project on Advanced Research on Architectures and Design of Electronic Systems (PARADES), by CNR P.F. MADESS II, and by GSRC.

A. Balluchi is with PARADES, Rome 00186, Italy (e-mail: balluchi@parades.rm.cnr.it).

L. Benvenuti is with PARADES, Rome 00186 Italy and also with the Dipartimento di Ingegneria Elettrica, Università dell'Aquila, L'Aquila 67040, Italy (e-mail: lucab@parades.rm.cnr.it).

M. D. Di Benedetto is with the Dipartimento di Ingegneria Elettrica, Università dell'Aquila, L'Aquila 67040, Italy (e-mail: dibenede@ing.univaq.it).

C. Pinello is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: pinello@eecs.berkeley.edu).

A. L. Sangiovanni-Vincentelli is with PARADES, Rome 00186, Italy and also with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: alberto@eecs.berkeley.edu).

Publisher Item Identifier S 0018-9219(00)06457-4.

NOMENCLATURE

V_α	Throttle motor input voltage.
α	Throttle valve angle.
p	Intake manifold pressure.
θ	Crankshaft angle.
n	Crankshaft speed.
α_e	Drive-line torsion angle.
ω_p	Wheel revolution speed.
a	Vehicle acceleration.
j	Vehicle jerk.
T	Torque generated by the engine.
T^i	Torque generated by the ith cylinder.
T_l	Load torque on the crankshaft.
q^i	Fuel loaded in the ith cylinder.
$\tilde{q}^i$	Desired loaded fuel for the ith cylinder.
m^i	Air loaded in the ith cylinder.
ϕ^i	Position of the ith cylinder.
φ^i	Spark advance of the ith cylinder.
$\tilde{\varphi}^i$	Desired spark advance for the ith cylinder.
η	Ignition efficiency.
m_C	Air trapped in the cylinder in compression stroke.
m_E	Air trapped in the cylinder in expansion stroke.

I. INTRODUCTION

Hybrid systems have been the subject of intensive study in the past few years by both the control and the computer-science communities. Particular emphasis has been placed on a unified representation of hybrid models rooted in rigorous mathematical foundations [1]–[7]. Some classical problems such as reachability analysis [8], [9], stability, and safety [10]–[12] have been investigated and tools for their solutions, i.e., HyTech [13], [14], Kronos [15], Checkmate [16], developed. However, since the class of hybrid control problems is extremely broad (it contains continuous control problems as well as discrete event control problems as special cases), it

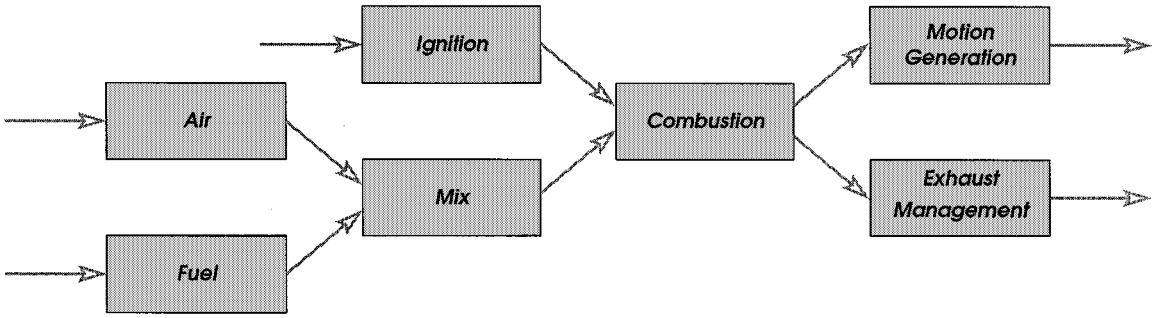


Fig. 1. Functional decomposition of the system.

is very difficult to devise a general yet effective strategy to solve them. In our opinion, it is important to address significant application domains to develop further understanding of the implications of the hybrid model on control algorithms and to evaluate whether using this formalism can be of substantial help in solving complex, real-life control problems.

In this paper, we focus on an application domain for hybrid system theory that is of great industrial interest: automotive engine and power-train control. The engine control problem is very complex (see, e.g., [17] and [18]). Fig. 1 shows the decomposition of the system in a chain of its basic processes [19]–[21]. Air intake and fuel injection can be controlled to yield the desired mix to deliver to the combustion process. The timing of the sparks generated by the spark plugs determines the start of the combustion process that takes place in the cylinders. The torque and the emissions generated by the combustion process depend on the air–fuel mix (quantity and dynamics) and on the spark ignition timing. The torque is then delivered to the power-train and the emissions to the exhaust subsystem. The goals for the control strategy are, in general, given in terms of torque and emissions, but it is often the case that subgoals are given by car manufacturers on all the processes in the chain. Unfortunately, even these goals are quite complex to specify since they depend on the behavior that the driver requests the car to have.[1]

Based on the data from sensors measuring emissions and car dynamics, embedded controllers compute the control actions to apply. The ever increasing computational power of microcontrollers has made it possible to extend the performance and the functionality of these embedded controllers to limits that were unthinkable only a few years ago. This opportunity has exposed the need for control algorithms with guaranteed properties that can reduce substantially emissions and gas consumption while maintaining the performance of the car (e.g., [22]–[24]). To do so, we believe that it is important to use more accurate models than the ones proposed so far. An accurate model of a four-stroke gasoline engine has a "natural" hybrid representation because

- pistons have four modes of operation corresponding to the stroke they are in; hence, their behavior can be represented with a finite-state model;
- power-train and air dynamics are continuous-time processes.

[1]A Ferrari has a different set of requirements on how to react to a fast push on the accelerator pedal than a Honda Civic.

In addition, these processes interact tightly. In fact, the timing of the transitions between two phases of the pistons is determined by the continuous motion of the power-train, which, in turn, depends on the torque produced by each piston. In engine control literature (e.g., [17], [25], and [26]), the discrete part of the engine behavior is converted into a more familiar and easy-to-handle continuous model where only the average values of the appropriate physical quantities are modeled.

In this paper, we present a general framework for power-train control based on hybrid models and demonstrate that it is possible to find effective control laws with guaranteed properties without resorting to average-value models. Hybrid models are used in our paper in two ways:

- to describe a reference model that captures the requirements of car manufacturers in a simple yet rigorous formalism;
- to represent engine and power-train behavior.

The hybrid nature of the problem of engine control does not come only from the use of digital control laws but also it is rooted in the plant to be controlled. This characteristic complicates the modeling problem since several components with different mathematical properties are needed to capture the behavior of the engine and of the power-train. In fact, we use the term "hybrid model" for engine and power-train in a somewhat more general sense than it is common since our model includes interacting finite-state machines, discrete event systems, and continuous time systems. We believe that the interactions among the different parts of a hybrid model of this type have not always been correctly described and characterized. For these reasons, we review the concept of models of computation, which is commonly used in the embedded system design literature, and describe a recently proposed modeling framework called the tagged signal model [27], [28] that addresses explicitly time and partial orders among events and is particularly suited to represent modality of interaction. Our control algorithms applied to the hybrid model for engine and power-train share the same basic idea: relax the hybrid control problem to a simpler one and then modify appropriately the control law determined for the simpler problem to solve the original hybrid problem.

This paper is organized as follows. In Section II, we introduce the model of computations and the tagged signal model framework. In Section III, we introduce a hybrid model that captures the car manufacturers' requirements. In Section IV, we describe the hybrid model of the engine and of the power-

train in detail. In Section V, we examine three different control problems corresponding to different regions of operation identified in our model and show how hybrid system techniques can be used to develop control strategies with guaranteed properties. A final perspective on the use of hybrid systems for automotive control is offered in Section VI.

II. Background on Models of Computation

A design (at all levels of the abstraction hierarchy from functional specification to final implementation) is generally represented as a set of components, which can be considered as distinct monolithic blocks, interacting with each other and with an environment that is not part of the design. The model of computation (MOC) defines the behavior and interaction of these blocks. Compactness of description, fidelity to design style, ability to synthesize to an appropriate implementation and to optimize its behavior are criteria that guide the selection of an MOC. For example, some MOCs are suitable for describing complicated data transfer functions and completely unsuitable for complex control, while others are designed with complex control in mind. Hybrid systems in general terms can be viewed as formalisms for describing a complex system using combinations of MOCs when a single one is not powerful, expressive, or practical enough.

Traditionally, hybrid systems have been defined as finite automata in which each state is associated with a set of differential equations, and transitions are triggered based on state and input values of the differential equations. However, the interactions among the discrete and the continuous dynamics are nontrivial. To capture these interactions rigorously and to prevent confusion that has often arisen by their improper modeling, the concept of hybrid system execution has been recently introduced in [29]. We believe that the concept of hybrid system can and should be extended to contain any combination of different models of computation, not necessarily finite-state automata and ordinary differential equations. In our "hybrid" model of the engine and of the power-train, we use three different models of computation. We use the theory behind the notion of models of computation to identify the role of the interfaces among the different components of the hybrid system. The foundations of a theory of models of computation (for an extended presentation, see [27] and [28]), reviewed in the next section, have been essential in our development.

A. The Tagged-Signal Model

The tagged-signal model (TSM) proposed by Lee and Sangiovanni-Vincentelli [27], [28] is a formalism for describing aspects of models of computation. It defines a semantic framework within which models of computation can be studied and compared. It is abstract—describing a particular model of computation involves imposing further constraints that make it more concrete.

The fundamental entity in the TSM is an event: a value/tag pair (v, t). Tags are often used to denote temporal behavior. A set of events (an abstract aggregation) is a signal. Processes are relations on signals, expressed as sets of tuples of signals. A particular model of computation is distinguished by the order it imposes on tags and the character of processes in the model. More formally, given a set of *values* V and a set of *tags* T, an *event* is an element of $V \times T$. A *signal* s is a set of events, and thus is a subset of $V \times T$. A *functional* (or deterministic) *signal* is a (possibly partial) function from T to V. The set of all signals is denoted by S. A *tuple* of n signals is denoted by $\boldsymbol{s}$, and the set of all such tuples is denoted by S^n. The *empty signal* in S (one with no events) is denoted by λ. For any $s \in S$, $s \cup \lambda = s$. In some models of computation, the set V includes a special value $\perp$, which indicates the *absence of a value*. It should be noticed that $(\perp, t) \notin \lambda$, indeed $(\perp, t)$ does not satisfy $(\perp, t) \cup s = s$ for all $s \in S$.

The issue of time representation has been central to all modeling efforts. While time has a rather well-studied representation in physical processes, this is not always the case in specifications of designs. In fact, we argue that representing specifications using physical time equivalents may result in overspecifications and, as a consequence, less efficient designs. For example, data manipulation operations can often be performed concurrently as long as certain precedence relations are satisfied. The specifications for these systems have to reflect only the precedence relations, thus leaving several options open for embedding the computation in physical processes that will indeed have a global ordering on the computation. If such ordering in time is used for specifying the system, there is limited or no freedom in selecting the embedding of the computation. The tagged-signal model has been primarily developed to clarify the issue of time, concurrency, and communication for embedded systems.

1) Processes: A *process* P is a subset of the set of all n-tuples of signals S^n for some n. A particular $\boldsymbol{s} \in S^n$ is said to *satisfy* the process if $\boldsymbol{s} \in P$. An $\boldsymbol{s}$ that satisfies a process is called a *behavior* of the process. Thus a *process* is a set of possible *behaviors*, or a relation between signals.

Different order relations on the set of tags T partition the space of process representations. In a *timed process* T is totally ordered, i.e., there is a binary relation $<$ on members of T such that if $t_1, t_2 \in T$ and $t_1 \neq t_2$, then either $t_1 < t_2$ or $t_2 < t_1$. In an *untimed process*, T is only partially ordered. For instance, data flows are represented by untimed processes.

Intuitively, a set of processes operate *concurrently*, and constraints imposed on their signal tags define *communication* among them. The environment in which the system operates can be modeled with a process as well.

For many (but not all) applications, it is natural to partition the signals associated with a process into *inputs* and *outputs*. Intuitively, the process does not determine the values of the inputs, and does determine the values of the outputs. A process with i inputs and o outputs is a subset of $S^i \times S^o$, where (S^i, S^o) is a partition of S^n and $n = i + o$. Thus, a process defines a *relation* between input signals and output signals. An $\boldsymbol{s}$ can be written $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2)$, where $\boldsymbol{s}_1 \in S^i$ is an i-tuple of *input signals* for process P and $\boldsymbol{s}_2 \in S^o$ is an o-tuple of *output signals* for process P.

A process F is *functional* (or deterministic) with respect to an input/output partition if it is a single-valued, possibly partial, mapping from S^i to S^o. That is, if $(\boldsymbol{s}_1, \boldsymbol{s}_2) \in F$ and $(\boldsymbol{s}_1, \boldsymbol{s}_3) \in F$, then $\boldsymbol{s}_2 = \boldsymbol{s}_3$. In this case, we can write $\boldsymbol{s}_2 = F(\boldsymbol{s}_1)$, where F: $S^i \rightarrow S^o$ is a (possibly partial) function. A process is *completely specified* if it is a total function, i.e., for all inputs in the input space, there is a unique behavior.

In a *memoryless process*, only inputs with a given tag concur to form outputs with the same tag. The notion of state has traditionally been used for processes with memory to simplify their representation and to provide a powerful analysis and synthesis mechanism. We can formalize the notion of *state* in the TSM following some of the classical notions of system theory (e.g., [30]) by considering a process F that is functional with respect to partition (S^i, S^o). Let us assume for the moment that F belongs to a timed process, in which tags are totally ordered. For any tuple of signals $\boldsymbol{s}$, define $\boldsymbol{s}_{>t}$ to be the tuple of the (possibly empty) subset of the events in $\boldsymbol{s}$ with tags greater than t. Two input signal tuples $\boldsymbol{r}, \boldsymbol{s} \in S^i$ are in relation E_t^F [denoted by $(r^i, s^i) \in E_t^F$] if $\boldsymbol{r}_{>t} = \boldsymbol{s}_{>t}$ implies $F(\boldsymbol{r})_{>t} = F(\boldsymbol{s})_{>t}$. This definition intuitively means that process F cannot distinguish between the *histories* of $\boldsymbol{r}$ and $\boldsymbol{s}$ prior to time t. Thus, if the inputs are identical after time t, then the outputs will also be identical. E_t^F is obviously an equivalence relation, partitioning the set of input signal tuples into equivalence classes for each t. We call these equivalence classes the *states* of F. Under certain conditions, the notion of state can be generalized to untimed models of computation, where events are tagged with partially ordered tags. It is sufficient, for example, that a set A of tags exist that are totally ordered with respect to every event in the input and output signals of a process. Then similar equivalence classes can be defined for each tag in A.

2) Concurrency and Communication: The sequential or combinational behavior just described is related to individual processes, and general systems will typically contain several coordinated concurrent processes. At the very least, such processes interact with an environment that evolves independently, at its own speed. It is also common to partition the overall model into tasks that also evolve more or less independently, occasionally (or frequently) interacting with one another. This interaction implies a need for coordinated communication.

Communication between processes can be *explicit* or *implicit*. Explicit communication implies forcing an order on the events, and this is typically realized by designating a *sender* process, which informs one or more *receiver* processes about some part of its state. Implicit communication implies the sharing of a common time scale (which forces a common partial order of events) and a common notion of state.

a) Basic time: In classical transformational systems, such as personal computers, the correct result is the primary concern—when it arrives is less important (although *whether* it arrives *is* important). By contrast, embedded systems are usually real-time systems, where the time at which a computation takes place is very important. For example, a delay in displaying the result of an Internet search is annoying; a delay in actuating a brake command is fatal.

As mentioned previously, different models of time become different order relations on the set of tags T in the tagged-signal model. Implicit communication generally requires totally ordered tags (*timed processes*), usually identified with physical time.

The tags in a *metric timed process* have the notion of a "distance" between them, much like physical time.[2]

Two events are *synchronous* if they have the same tag (the distance between them is zero). Two signals are synchronous if each event in one signal is synchronous with an event in the other signal and vice versa.

b) Treatment of time in processes: A *synchronous process* is one in which every signal in the process is synchronous with every other signal in the process. An *asynchronous process* is a process in which no two events can have the same tag. Note that the common usage of the term "asynchronous" refers to processes that are *non-synchronous*. Asynchronous processes in our framework are a subset of nonsynchronous processes. We believe that distinguishing between asynchronous and nonsynchronous is important. In fact, asynchronous functional processes always have a unique behavior while all other processes may have problems when feedback connections involving events with the same tag are present. If tags are totally ordered, the process is *asynchronous interleaved*, while if tags are partially ordered, the process is *asynchronous concurrent*. Note, however, that for asynchronous processes, concurrency and interleaving are, to a large extent, interchangeable, since interleaving can be obtained from concurrency by partial order embedding and concurrency can be reconstructed from interleaving by identifying "untimed causality."

c) Implementation of concurrency and communication: Concurrency in physical implementations of processes implies a combination of *parallelism*, which employs physically distinct computational resources, and *interleaving*, which means sharing of a common physical resource. Mechanisms for achieving interleaving, generally called *schedulers*, vary widely, ranging from operating systems that manage context switches to fully static interleaving, in which multiple concurrent processes are converted (compiled) into a single process. We focus here on the mechanisms used to manage communication between concurrent processes.

Parallel physical systems naturally share a common notion of time, according to the laws of physics. The time at which an event in one subsystem occurs has a natural ordering relationship with the time at which an event occurs in another subsystem. Physically interleaved systems also share a natural common notion of time: one event happens before another.

A variety of mechanisms for managing the order of events, and hence for communicating information between

[2]Formally, there exists a function d: $T \times T \rightarrow \mathbb{R}$ mapping pairs of tags to real numbers such that $d(t_1, t_2) \geq 0$ where $d(t_1, t_2) = 0 \Leftrightarrow t_1 = t_2$, $d(t_1, t_2) = d(t_2, t_1)$ and $d(t_1, t_2) + d(t_2, t_3) \geq d(t_1, t_3)$ for any t_1, t_2, t_3.

processes, exists. Using processes to model communication (rather than considering it as "primitives" of the TSM) makes it easier to compare different MOCs, and also allows one to consider *refining* these communication processes when going from specification to implementation [31].

Recall that *communication* in the TSM is embodied in the event, which is a two-component entity whose value is related to function and whose tag is related to time. That is, communication is implemented by two operations:

1) the transfer of values between processes (function; TSM event value);
2) the determination of the relationship in time between two processes (time; TSM event tag).

Unfortunately, often the term "communication" (or data transfer) is used for the former, and the term "synchronization" is used for the latter. We feel, however, that the two are intrinsically connected in embedded systems: both tag and value carry *information* about a *communication*. Thus, communication and synchronization, as mentioned before, are terms that cannot really be distinguished in this sense.

B. Models of Computation Used in Our Model

Car-manufacturer requirements are given in terms of regions of operations of the engine and on driver actions. The region of operations are finite. The transitions from one region to the other is determined by driver actions and by internal dynamics of the engine and of the power-train. Engine and power-train are physical systems, where time plays an essential role. Specifications for the controllers and their implementation are based on physical time. The description of the physical system uses an abstraction of time that involves sequences, i.e., what matters is the sequences of events, not the precise time at which these events occur. Timed MOCs (recall that timed MOCs include both models that are based on physical time and those based on sequences of events) will be used in the description of our system.[3] In particular, we use the following.

- *Finite-state machines (FSMs)*: An FSM is a synchronous TSM process in which the tags take values in the set of integers and the sets of inputs, outputs, and states are finite. The tags represent the ordering of the sequence of events in the signals, not physical time, and are globally ordered. A *finite automaton* (FA) is an FSM with no outputs.
- *Sequential systems (SSs)*: An SS is a synchronous TSM process in which the tags take values in the set of integers and the inputs, outputs and states assume values on infinite sets.
- *Discrete-event systems (DESs)*: In a DES, tags are order-isomorphic with the natural numbers and assume values on the set of real numbers. The tags represent the values of the time at which the events occur.
- *Continuous-time systems (CTSs)*: A CTS is a metric timed TSM process, where T is a connected set.

[3]Note that the specification for the controller may use untimed models. In this paper, we focus on physical models and control laws, and this is the reason for the use of timed models.

The essential issue is how to compose these different models to describe a complex system.

III. System Behavior Specifications

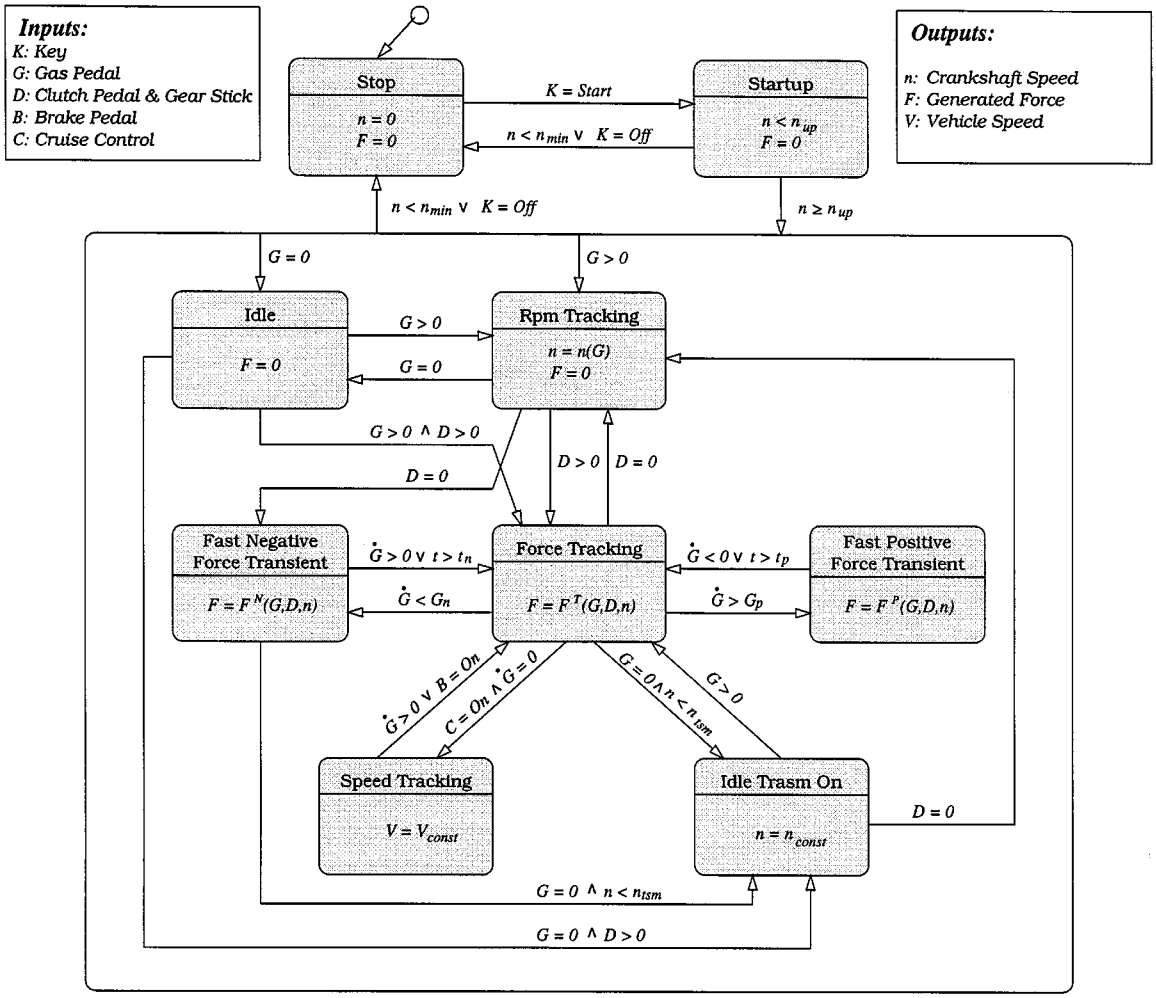
A car manufacturer gives system specifications that describe how the vehicle should react to the inputs supplied by the driver. In this paper, we restrict our attention to specifications related to force requests in the longitudinal motion of the car. We assume that the driver requests force by using the gas, brake, and clutch pedals and a manual gear shift. Usually, the specifications are not provided in a well-structured manner; they are often obscured by considerations involving the implementation of the electronic subsystem. Because of the increased complexity of the engines, of the power-train, of the functionality requested of the control system, and the pressure for reduced time-to-market, there has been an increased interest in methods that could reuse parts of existing designs and that substantially reduce design iterations, eliminating errors. Being able to enucleate the requirements from other considerations related to the model of the engine and to the control strategy is an important step toward better reuse and fewer errors.

In our approach, the specifications are captured using a hybrid model as shown in Fig. 2. The discrete modes of the hybrid system correspond to different regions of operations of the engine.[4] The transitions are determined by the action of the driver or by engine conditions. Each region of operation is characterized by an optimization problem that includes a set of constraints related to driving performance, such as comfort and safety, or gas and noise emission, and a cost function that identifies the desired behavior of the controlled system. In each region of operation, the controlled system is represented by a model that includes ordinary differential equations as well as discrete components.

The goal of the controller is to act on the inputs to the plant (the throttle plate motor, the injection pulse duration, and the spark advance angle) so that it behaves according to the specifications summarized in the hybrid reference model of Fig. 2.

The initial state is the *Stop* state that corresponds to the engine being off. From this state, the driver causes the transition to the *Startup* state, turning the ignition key to the start value ($K = Start$). From the *Startup* state, the *Idle* state is entered if the gas pedal is released ($G = 0$); otherwise, ($G > 0$), the *RPM Tracking* state is entered. The desired behavior of the controlled system in the *Idle* state is the following: the force transmitted to the vehicle has to be zero ($F = 0$) and the crankshaft revolution speed must be regulated to a reference value. For instance, under a load torque smaller than 12 Nm, the excursion of the crankshaft revolution speed with respect to the reference value of 800 rpm must be less than 30 rpm while minimizing fuel consumption. Moreover, the transient response with respect to sudden loads up to 25 Nm must have a settling time of 2 s with an excursion less than 40 rpm. Last, the air–fuel ratio must remain within 1% of the

[4]The use of a finite automaton to capture specifications was, to the best of our knowledge, introduced in [32].



(a)

Fig. 2. FSM of the regions of operation describing system specifications.

stoichiometric desired level. In the *RPM Tracking* state, the crankshaft speed is requested to track the gas pedal signal $(n = n(G))$. When the transmission is engaged $(D > 0)$, the *Force Tracking* state is entered. In this state, a particular torque profile, which depends on the gas pedal position, the gear, and the crankshaft speed $(F = F^T(G, D, n))$, must be achieved with constraints on the settling time and with cost function related to fuel consumption and drive comfort. The *Fast Negative Force Transient* state is entered if the gas pedal is suddenly released ($\dot{G} < G_n$, where $\dot{G}$ represents the speed with which the pedal is released, and G_n is a reference value). In this state, a step reference for the requested force is considered and a different control law is used. This control law should minimize the transient time subject to very tight drive comfort constraints. However, if no electronic-throttle control system[5] is available, a feasible solution may not exist. In this case, the optimization problem is reformulated so that the driver comfort constraint becomes the cost function and the transient time is required to be bounded. In this case, we refer to this region of operation as the *Cutoff* state. When the gas pedal is pushed again or the transient is elapsed $(t > t_n)$, a transition to the *Force Tracking* state occurs. If the gas pedal is completely released and the minimum crankshaft speed is approached, the transition to the *Idle with transmission on* state is enabled. In this state, the crankshaft speed is kept constant at a prescribed value $(n = n_{const})$ until the gas pedal position or the crankshaft speed $(n > n_{tsm})$ triggers the transition to the *Force tracking* state. If the gas pedal is pushed quickly, then the transition to the *Fast Positive Force Transient* state is enabled. Here, the torque profile to be followed is a positive step, and the control law minimizes the transient time subject to fuel consumption and comfort constraints. Finally, when a cruise control is available on the car, a *Speed Tracking* state is present that is entered from the *Force tracking* state with the speed tracking control on $(C = On)$ and with no variation on the position of the accelerator pedal $(\dot{G} = 0)$. The state is exited if either the brake pedal has been touched $(B = On)$ or the accelerator pedal position is changed to indicate more speed wanted $(\dot{G} > 0)$.

[5]In traditional cars, the gas pedal is directly connected to the throttle valve, so that the amount of air loaded by the cylinders is fixed by the driver. In modern cars, the gas pedal command is processed by the engine control system that actuates the throttle valve by means of an electric motor. This allows full air flow regulation. Such equipment is usually referred to as *electronic-throttle control system*.

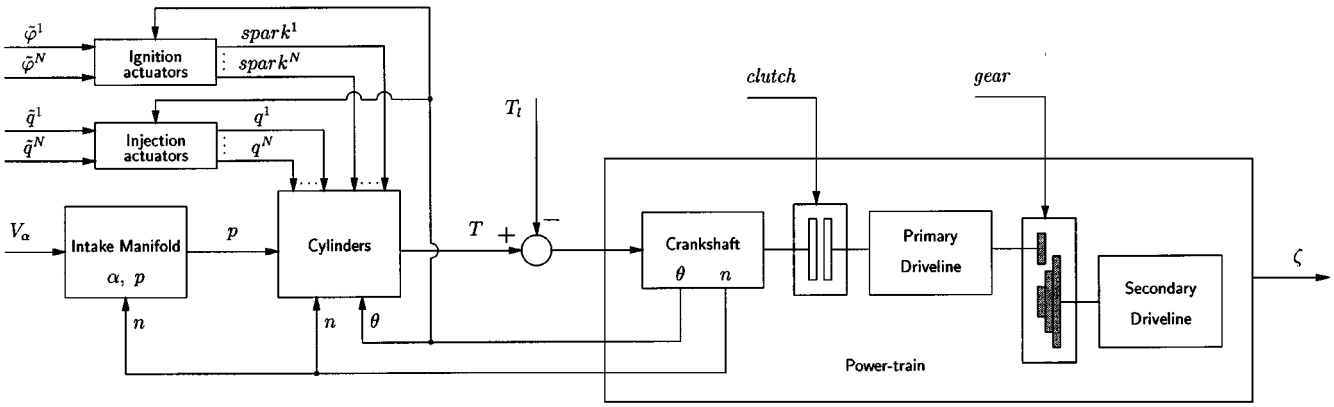


Fig. 3. Engine blocks and their communication topology.

If the driver turns the key to the stop position $(K = Off)$ or if the engine turns off $(n < n_{\min})$, then the system must return to the *Stop* state.

Although this scheme is simple, it has quite an impact in clarifying the objectives of the design and how to decompose the problem into well-defined subproblems. The requirements captured by the diagram can be considered as the specifications for the control problem. The structure of the hybrid reference model is the same independently of the car manufacturer or of the car class: only the parameters of the cost and constraint functions change.

IV. Hybrid Model for Engine and Power-Train

In this section, we propose a new model of a power-train with a N-cylinder four-stroke engine.[6] The overall system is composed of four main interacting blocks, namely, the *intake manifold*, the *cylinders*, the *power-train*, and the *actuators* (Fig. 3). Our goal is to develop a model that is general enough to serve as a "platform" for the development of control algorithms for the entire range of behaviors of the engine–power-train subsystem.[7]

The tagged signal model formalism allows us to formally describe systems represented as interacting processes of heterogeneous MOCs. Their interaction has to be clearly defined and analyzed since the general properties of the overall system depend critically on such interaction. Our formalism has the definite advantage of orthogonalizing concerns such as behavior and communication, time and ordering of events so that description and analysis is made simpler and clearer. This is the avenue we have followed to represent the physical processes taking place in an automotive engine and their interaction. In particular, we used a combination of FSMs, DESs, and CTSs to form a hybrid system that is the basis for our design.

A. General Structure of the Model

The manifold pressure p is controlled by the throttle valve, which, in the electronic-throttle case, is powered by an electrical motor. We denote by V_α and α the motor input voltage and the throttle-valve position, respectively. The mass of air loaded in the cylinders depends on the pressure p and on the crankshaft revolution speed n.

The torque T produced by the engine is given by $\sum_{i=1}^{N} T^i$, where T^i is the torque generated by the ith cylinder, which is determined by the mass of loaded air m^i, the mass of fuel q^i injected in the cylinder, and the $spark^i$ ignition command.[8] The timing sequence of the four strokes of each cylinder is determined by the continuous motion of the crankshaft. We denote by θ the crankshaft angular position, which is obtained by the integration of the crankshaft velocity n.

The ignition subsystem generates, for each cylinder, the spark signal $spark^i$ at the instant of time specified by the desired spark advance $\tilde{\varphi}^i$. The injection subsystem[9] injects an amount of fuel q^i according to the desired value $\tilde{q}^i$. By measuring the crankshaft angle θ, both actuation systems synchronize with the evolution of the cylinders.

Finally, the power-train dynamics, controlled by the generated torque T, are subject to the sum of a number of load torques T_l and depend on the position of the *clutch* and the selection of the *gear*.

B. The Model

In this section, we give a detailed description of the behavior of the four blocks that compose the engine and power-train system—namely, the *intake manifold*, the *cylinders*, the *power-train*, and the *actuators*—and of their composition (see Fig. 3).

1) The Intake Manifold: Manifold pressure dynamics is a continuous-time process controlled by the throttle-valve position α that changes the effective section of the intake rail of

[6]For a survey on internal combustion engines, see [20], [21], and [33]–[38].

[7]For brevity, we report in this paper only the parts of the model which are of interest in the control problems presented here.

[8]From this point on, we use the superscript i to indicate variables related to the ith cylinder.

[9]In non-GDI (gasoline direct injection) systems, the injectors deliver fuel in the intake manifold. The fuel adheres in part to the walls of the manifold forming a liquid film [39], which will evaporate and enter the cylinder during the next intake. To accurately account for the liquid film, a new dynamical system is required. This refined modeling is very important for finely controlling the air–fuel mix composition. We assume that this phenomenon is compensated by the fuel injection algorithm, so we neglect the fuel dynamics in the proposed model and directly think in terms of fuel entering the cylinder.

the manifold. Denoting by p the mean-value pressure, manifold dynamics is modeled as (see [40] and [26])

$$\dot{p}(t) = a_p[n(t), p(t)]n(t)p(t) + b_p[p(t)]s[\alpha(t)] \quad (1)$$

where $s(\alpha)$ is the *equivalent throttle area*, given in terms of throttle angle α. Parameters a_p and b_p depend in a strongly nonlinear fashion on the geometric characteristics of the manifold, on the physical characteristics of the gas and atmosphere, and on the current value of the pressure p and engine speed n. While in traditional engines the throttle valve is directly connected to the gas pedal, in electronic-throttle systems, it is controlled by the engine control system to achieve better performance. The dynamics of actuation of the throttle valve (usually a dc motor) is modeled by a linear first-order dynamical system.

2) The Cylinder: The cylinder model is the most complex. It is "responsible" for torque generation. The torque T^i generated by each piston at each cycle depends on the thermodynamics of the air–fuel mixture combustion process. The profile of T^i depends on the phases of the cylinder, the piston position ϕ^i, the mass m^i of air, the mass q^i of fuel both loaded in the cylinder during the intake phase, and the spark ignition timing.

In a four-stroke combustion engine, a piston reaches the *top dead center* (TDC) [*bottom dead center* (BDC)] when it is at its uppermost (lowermost) position. Each cylinder cycles through the following four phases.

- *Intake* (I): The piston goes down from the TDC to the BDC loading the air–fuel mix present in the intake manifold.
- *Compression* (C): The trapped mix is compressed by the piston during its upward movement from the BDC to the TDC.
- *Expansion* (E): The combustion takes place pushing down the piston from the TDC to the BDC.
- *Exhaust* (H): During its upward movement, from the BDC to the TDC, the piston expels combustion exhaust gases.

Let ϕ^i be the position of the ith piston, expressed in terms of the position of the corresponding crank angle, with respect to the last *dead center* (DC), that is

$$\phi^i(t) = [\theta(t) - \phi_0^i] \bmod 180^\circ \quad (2)$$

where ϕ_0^i is the value of θ for which the ith cylinder is at a dead center. This corresponds to reset ϕ^i at the beginning of each phase. Note that since the pistons are connected to the crankshaft, their positions ϕ^i are related to each other.[10]

The quantity m^i of air loaded into each cylinder at the end of the intake run depends, in a nonlinear fashion, on the evolution of the intake manifold pressure and the crankshaft speed. The amount of air loaded up to time t, denoted by $m^i(t)$, is sampled at the intake BDC time t_ℓ^i to obtain the loaded air for the current engine cycle.

The amount of fuel q^i that can be injected is subject to constraints to limit emissions and to increase efficiency. These constraints are usually expressed in terms of the air-to-fuel ratio $A/F = m^i/q^i$ of the mixture. When $A/F = 14.64$, the mix is said to be at stoichiometry, which is a desirable operating point for emissions. Rich mixtures $A/F < 14.64$ produce excess of CO and HC, while lean mixtures $A/F > 14.64$ have excess of NO_x. The efficiency of three-way catalytic converters, commonly used to reduce emissions, is satisfactory only in a narrow range around the stoichiometry ($\pm 5\%$). This results in an interval $[q_{\min}, q_{\max}] = [15.4^{-1}, 13.9^{-1}]m^i$ of admissible values for the injected fuel.

Spark ignition must occur at every cycle. Intuitively, it should occur exactly when the piston reaches the TDC of the compression stroke. Since the combustion process takes nonzero time to complete, the pressure in the cylinder reaches its maximum some time after spark ignition. It is then convenient to produce a spark before the piston completes the compression stroke (*positive spark advance*), to achieve maximum fuel efficiency. Producing a spark after the piston has completed the compression phase and is in the expansion stroke (*negative spark advance*) may be used to reduce drastically the value of the torque generated during the expansion run. Hence, the spark control input has a very short delay and can be used to reduce torque much faster than using only the throttle valve.[11] The spark ignition time is commonly defined in terms of the spark advance φ^i, which denotes the difference between the angle of the crank at the TDC between compression and expansion and the one at the time of ignition t_j^i. In terms of the piston position ϕ^i, we have

$$\varphi^i = \begin{cases} 180^\circ - \phi^i(t_j^i), & \text{for a positive spark advance} \\ -\phi^i(t_j^i), & \text{for a negative spark advance.} \end{cases} \quad (3)$$

It is positive for sparks ignited in the compression stroke, negative otherwise. Note that the spark advance has to be bounded both from above and from below to prevent the mix from not burning uniformly, thus causing undesired knocking [41], [42] (upper bound) and from misfiring [43], [44] (lower bound), which causes undesired pollutants. These bounds depend on the revolution speed n. The spark advance and the amount of injected fuel is set at each cycle to control the generated torque (see [45]).

The air–fuel mixture is loaded in the cylinder during the intake stroke while the torque generation starts after the spark is ignited. Hence, to complete the description of the torque generation process, we need to model the delay between the time at which the mixture is loaded and the time at which the corresponding active torque is generated [46].

The overall model of the torque generation process for a single cylinder consists of four communicating processes of different MOCs:

[10]Most cars have four cylinders, but there are engines that have a different number of cylinders. For example, Formula 1 racing cars can have eight, ten, or 12 cylinders. The Fiat Coupé 2000 Turbo has five cylinders. In a four-cylinder in-line engine $\phi^1(t) = \phi^2(t) = \phi^3(t) = \phi^4(t)$.

[11]Because of this property, negative spark advance is very useful to regulate the engine speed in the $Idle$ region of operation.

- an FSM, modeling the four-stroke engine cycle;
- a DES, modeling the discrete delay on the active torque generation;
- two memoryless CTSs, modeling the air intake process and the profile of the generated torque.

a) Modeling the four-stroke engine cycle with an FSM: This part of the cylinder model is used to capture the sequential nature of the behavior of the cylinders. Based on the events generated by the spark ignition signal and by the reaching of dead centers, the FSM takes a transition and outputs the appropriate information to coordinate the other parts.

The four phases of the piston are associated to the states of an FSM that represents the behavior of the cylinder. A state transition would then occur when the piston reaches a dead center. However, the torque generated by the piston is related not only to the four phases of the piston but also to the spark generation process. Since spark ignition may occur either during the compression stroke or during the expansion stroke, a six-state FSM is needed to model the possible behaviors of the cylinder. The cylinder FSM is shown in Fig. 4. The FSM state s_k^i takes one of the following values.

- I, denoting *Intake*.
- BS, denoting *Before Spark*. The piston is in the compression stroke and no spark has been ignited yet.
- PA, denoting *Positive Advance*. The piston is in the compression stroke and the spark has been ignited.
- NA, denoting *Negative Advance*. The piston is in the expansion stroke and the spark has not been ignited yet.
- AS, denoting *After Spark*. The piston is in the expansion stroke and the spark has been ignited.
- H, denoting *Exhaust*.

The cylinder changes phase either when a spark is given (FSM input event $u_k^i = spark^i$ or $u_k^i = spark\&DC^i$ if the spark is given exactly at the dead center), or when a dead center is reached (FSM input event $u_k^i = DC^i$). The evolution of the torque produced by the cylinder depends on the transitions of the FSM, provided by the output o_k^i of the FSM that takes the following values: $BS2AS$, $BS2PA$, $BS2NA$, $PA2AS$, $NA2AS$, $AS2H$, $H2I$, and $I2BS$. The next-state and output functions of the cylinder FSM

$$s_{k+1}^i = \delta(s_k^i, u_k^i), \qquad o_k^i = \lambda(s_k^i, u_k^i) \tag{4}$$

are shown in Fig. 4. Note that, for the sake of notational simplicity, we dropped the superscript i, indicating the correspondence of the variable with cylinder i, from the index k.

b) Modeling the air-intake process with a CTS: Assuming small variations of the crankshaft speed n during intake and recalling that p represents the pressure mean-value over the engine cycle, air intake can be described by the following memoryless CTS:

$$m^i(t) = w[p(t), n(t)]. \tag{5}$$

This abstraction of the air-intake process is sufficient for most control purposes. However, this model can be further refined by describing the opening and closing of intake valves, which are synchronized with the piston position ϕ^i. In this paper, we will use $w[p(t), n(t)]$ proportional to the product $a[p(t), n(t)]p(t)$, with $a(\cdot)$ as in (1).

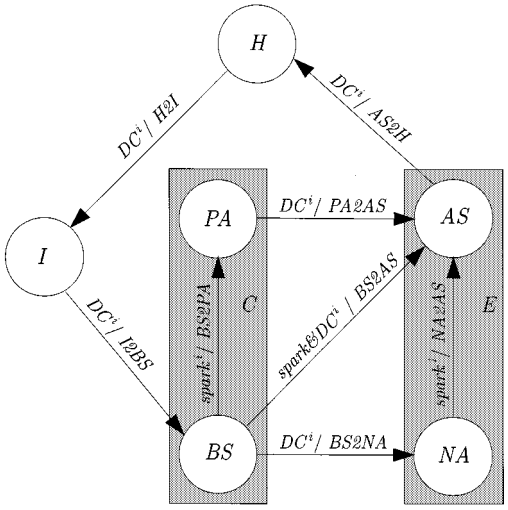


Fig. 4. FSM describing the behavior of the ith cylinder.

c) Modeling the torque profile with a CTS: The profile of the torque T^i produced by the ith piston as ϕ^i evolves is modeled by a memoryless CTS

$$T^i(t) = g_{o_k^i}[y_k^i, \phi^i(t)] \tag{6}$$

where o_k^i is the current FSM output and y_k^i collects the values (m^i, q^i, φ^i) and changes only at the FSM transitions. An example of torque profiles is given by Shim *et al.* in [47], where the shape of the torque is approximated with harmonics of a sinusoidal wave up to the third order.

This representation is general enough to allow the accurate description of complex torque profiles. However, in this paper, we restrict ourselves to a simpler model obtained by abstracting away the details of the combustion process as well as those related to gas pumping in and out of the cylinder. We set to zero the torque T^i during the passive phases of the cylinder, but we take into account the loss of energy due to these phases by reducing the amount of torque generated during the active phase. As a consequence of this simplification, the profile T^i is described by a piecewise constant function that is assumed to be zero everywhere except in the expansion phase when the spark ignition command has already been given, i.e.,

$$\begin{aligned} &g_{BS2PA} = g_{BS2NA} = g_{AS2H} = g_{H2I} = g_{I2BS} \equiv 0 \\ &g_{PA2AS} = g_{BS2AS} = g_{NA2AS} = G_f q^i \eta(\varphi^i) \end{aligned} \tag{7}$$

where the gain G_f represents the potential value of the torque that can be achieved by the given mix and the ignition efficiency function $\eta(\varphi)$ has in general the profile shown in Fig. 5. Note that in (7), we do not make explicit use of the mass of air m^i; this is a good approximation around stoichiometry.

d) Modeling the discrete delay on active torque generation with a DES: The delay on active torque generation,

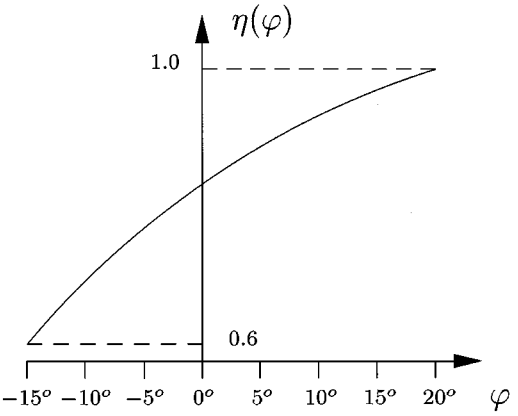


Fig. 5. Ignition efficiency function (at low engine speed).

which is characteristic of four-stroke engine cycles, is modeled by means of a DES synchronized with the FSM transitions and whose dynamics depends on the FSM transitions

$$\begin{aligned} z^i_{k+1} &= f_{o^i_k}(z^i_k, v^i_k) \\ y^i_k &= h_{o^i_k}(z^i_k, v^i_k) \end{aligned} \tag{8}$$

where o^i_k denotes the kth FSM transition. The components of the DES input vector v^i_k are

- the mass of air m^i loaded during the intake phase;
- the mass of injected fuel q^i during the intake phase;
- the piston position ϕ^i, used to compute the spark advance φ^i according to (3).

The DES state z^i_k is used to model the delay between the mixture intake and the active torque generation, while the DES output y^i_k provides the values (m^i, q^i, φ^i) to the CTS describing the profile of the engine torque (6).

The functions $f_{o^i_k}$ and $h_{o^i_k}$ describing the dynamics and the output of the DES are the following:

$$\begin{aligned} f_{I2BS} &= h_{I2BS} = (m^i_k, q^i_k, 0) \\ f_{BS2PA} &= h_{BS2PA} = z^i_k + (0, 0, 180^\circ - \phi^i_k) \\ f_{BS2NA} &= h_{BS2NA} = z^i_k \\ f_{BS2AS} &= h_{BS2AS} = z^i_k \\ f_{PA2AS} &= h_{PA2AS} = z^i_k \\ f_{NA2AS} &= h_{NA2AS} = z^i_k + (0, 0, -\phi^i_k) \\ f_{AS2H} &= h_{AS2H} = z^i_k \\ f_{H2I} &= h_{H2I} = z^i_k. \end{aligned} \tag{9}$$

Consider, for example, the torque produced in the state AS when a positive spark advance has been applied, i.e., $T^i(t) = g_{PA2AS}[y^i_k, \phi^i(t)]$. According to the DES dynamics, this torque depends on the value of the DES output y^i_k at the transition $PA \rightarrow AS$, which in turn depends on values m^i_{k-2}, q^i_{k-2} at the transition $I \rightarrow BS$, i.e.,

$$\begin{aligned} y^i_k &= z^i_k \\ &= z^i_{k-1} + (0, 0, 180^\circ - \phi^i_{k-1}) \\ &= (m^i_{k-2}, q^i_{k-2}, 180^\circ - \phi^i_{k-1}). \end{aligned}$$

This shows how the DES model captures the delays in the torque generation process: a one-step delay associated to the spark ignition (due to the fact that the spark is given during the compression stroke while the torque is generated during the expansion stroke), and a two-step delay associated to the mix mass (due to the fact that the mix is loaded during the intake phase).

e) Composing the MOCs describing the cylinder: Fig. 6 shows the (nontrivial) interactions among the MOCs modeling the different phenomena in the cylinder behavior. Using the TSM framework, we can underline the importance of sequencing and timing of events in the torque generation process and, hence, be more effective in the synthesis of control algorithms.

A transition in the FSM is caused by two possible events, $spark^i$ and DC^i, which may occur at the same time. The first task of the composition of the MOCs is to collect all the events that cause a transition in the FSM (producing a single input to the FSM) and the times at which these events occur. Note that the FSM uses only the information about the sequencing of events, not their exact timing. However, torque generation does need time information since it feeds a continuous time system (the model of the power-train) and the inputs to that CTS have to be correctly placed in time. Hence, the block that takes care of collecting the events is also responsible for generating the right coupling of sequence indexes and actual times.

The input signals $(spark^i, t^i_j)$, (DC^i, t^i_h), are received by the *signal_merge_1* process at times t^i_j (spark ignition times), t^i_h (DC times), respectively. This process merges the input signals to yield the FSM input signal (u^i_k, k), where k is the index of the totally ordered sequence $\{t^i_k\} = \{t^i_j\} \cup \{t^i_h\}$, so that $\{t^i_k\}$ is the sequence of times t^i_k at which the FSM associated to the ith cylinder takes the kth transition. Note that u^i_k takes the value $(spark\&DC^i, t^i_j)$ when $t^i_j = t^i_h$. The signals (k, t^i_k) and (o^i_k, t^i_k) are also produced as outputs to coordinate the other parts of the model. Signal (o^i_k, k) is received by the SS to select the current dynamics $f_{o^i_k}$ and output $h_{o^i_k}$. Signal (o^i_k, t^i_k) is used by the torque profile CTS and by the two samplers. The sequence of sampling times $\{t^i_j\}$ (corresponding to the spark ignition times) is extracted from the signal (o^i_k, t^i_k), by taking the tags t^i_k for which o^i_k assumes value in $\{BS2PA, BS2AS, NA2AS\}$. Similarly, sampling times $\{t^i_\ell\}$ (corresponding to intake BDC times) are extracted from the signal (o^i_k, t^i_k) by taking the tags t^i_k for which o^i_k assumes the value $I2BS$.

The CTS that represents torque generation needs the piston position, a continuous time variable, the phase of the cylinder (provided directly by the FSM output), and the amount of air and fuel that was loaded at the end of the intake phase as well as the spark advance. The DES delivers the information about the amount of air and fuel at the end of the intake phase as well as the spark advance, all appropriately placed in physical time by virtue of the signal merge taking place at its output. The inputs to the SS are generated by merging the appropriate signals so that the sequencing is synchronized with the input of the FSM. In doing so, we are "synchronizing" nonsynchronous signals by assigning the special value $\perp$ (absence of value) to signals that do not have a value for a particular tag that is of interest for another signal.

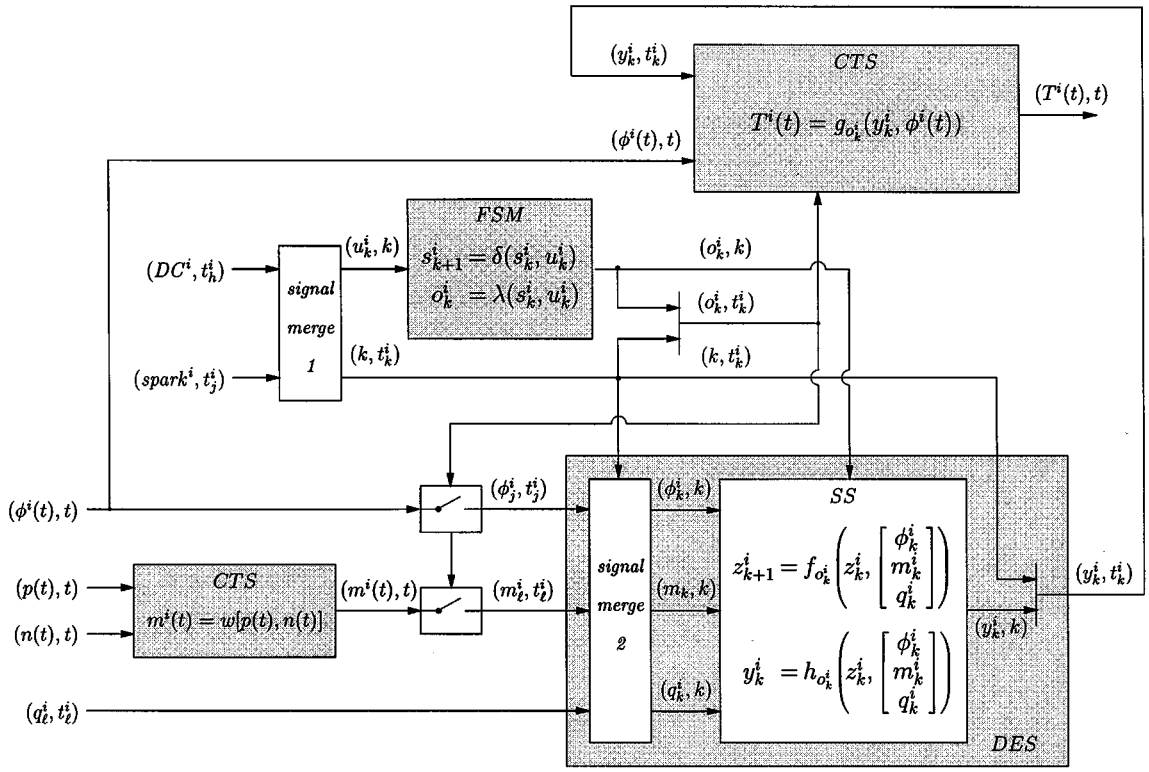


Fig. 6. TSM hybrid model of the ith cylinder.

In particular, at intake BDC times t_ℓ^i, the *signal_merge_2* process receives the signals (q_ℓ^i, t_ℓ^i) and (m_ℓ^i, t_ℓ^i)—where the value m_ℓ^i is equal to $m^i(t_\ell^i)$. At spark ignition times t_j^i, it further receives the signal (ϕ_j^i, t_j^i) (piston position at the time of the spark) to compute the spark advance according to (3). For each of the previous signals—say, (a_m, t_m^i)—on the basis of the signal (k, t_k^i), the *signal_merge_2* process yields the SS input signals (ϕ_k^i, k), (q_k^i, k), (m_k^i, k) applying the following rule:

$$(a_{\overline{k}}, \overline{k}) = \begin{cases} (a_{\overline{m}}, \overline{k}), & \text{if } (a_{\overline{m}}, t_{\overline{k}}^i) = (a_{\overline{m}}, t_{\overline{m}}^i) \in (a_m, t_m^i) \\ (\perp, \overline{k}), & \text{otherwise.} \end{cases} \tag{10}$$

This makes the output signals of the *signal_merge_2* process and of the FSM synchronous. Note that setting $a_{\overline{k}} = \perp$ when $t_{\overline{m}}^i \neq t_{\overline{k}}^i$ does not change the behavior of the system since the SS does not read this input when the tag k is equal to $\overline{k}$. Since $\{t_m^i\} \subset \{t_k^i\}$, then, by the previous rule, the sequence $\{a_m\}$ is appropriately augmented and defined over all times t_k^i.

From the SS output (y_k^i, k), the DES output signal (y_k^i, t_k) is generated by replacing the signal tag k with the corresponding time t_k. Then, this value is given as a parameter to the CTS to generate the output torque $T^i(t)$.

3) The Power-Train: For a given *gear* selection and *clutch* position, the power-train is described by the continuous time system

$$\dot{\zeta}(t) = A\zeta(t) + b(T(t) - T_l(t)) - b_0 \tag{11}$$

$$\dot{\theta}(t) = (0,\ 6,\ 0)\zeta(t) \tag{12}$$

$$a(t) = c\zeta(t) - d_0 \tag{13}$$

$$j(t) = (cA)\zeta(t) + (cb)(T(t) - T_l(t)) - (cb_0) \tag{14}$$

where $\zeta = (\alpha_e, n, \omega_p)^T$ includes the drive-line torsion angle, the crankshaft revolution speed, and the wheel revolution speed and θ is the crankshaft angle position. Input T is the torque produced by the engine, while T_l represents the load torque acting on the crankshaft. Vector b_0 models the resistant actions on the power-train, due to internal friction and external forces at the equilibrium point. Dynamics (11) is exponentially stable and is characterized by a real dominant pole λ_1, and a pair of conjugate complex poles $\lambda \pm j\mu$. Vehicle acceleration (13) and jerk (14) $(j = da/dt)$ are the power-train outputs of interest for drivability and comfort. Model parameters depend on the selected gear and the clutch position.

4) The Engine and Power-Train Model: Fig. 7 shows the hybrid model for vehicles with four-stroke N-cylinder gasoline engine. Such a model is obtained by combining N cylinder hybrid models and it is composed of the following parts:

1) N subsystems as in Fig. 6 describing the behavior of the N cylinders of the engine;
2) two CTSs modeling, respectively, the power-train and intake manifold dynamics;
3) the block $SYNC$ that synchronizes the cylinders models by generating the piston position angles ϕ^i from the crankshaft angle θ, according to (2), and the events DC^i.

Input $T(t)$ to the CT power-train dynamics is obtained by summing all the contributions $T^i(t)$ of the cylinders.

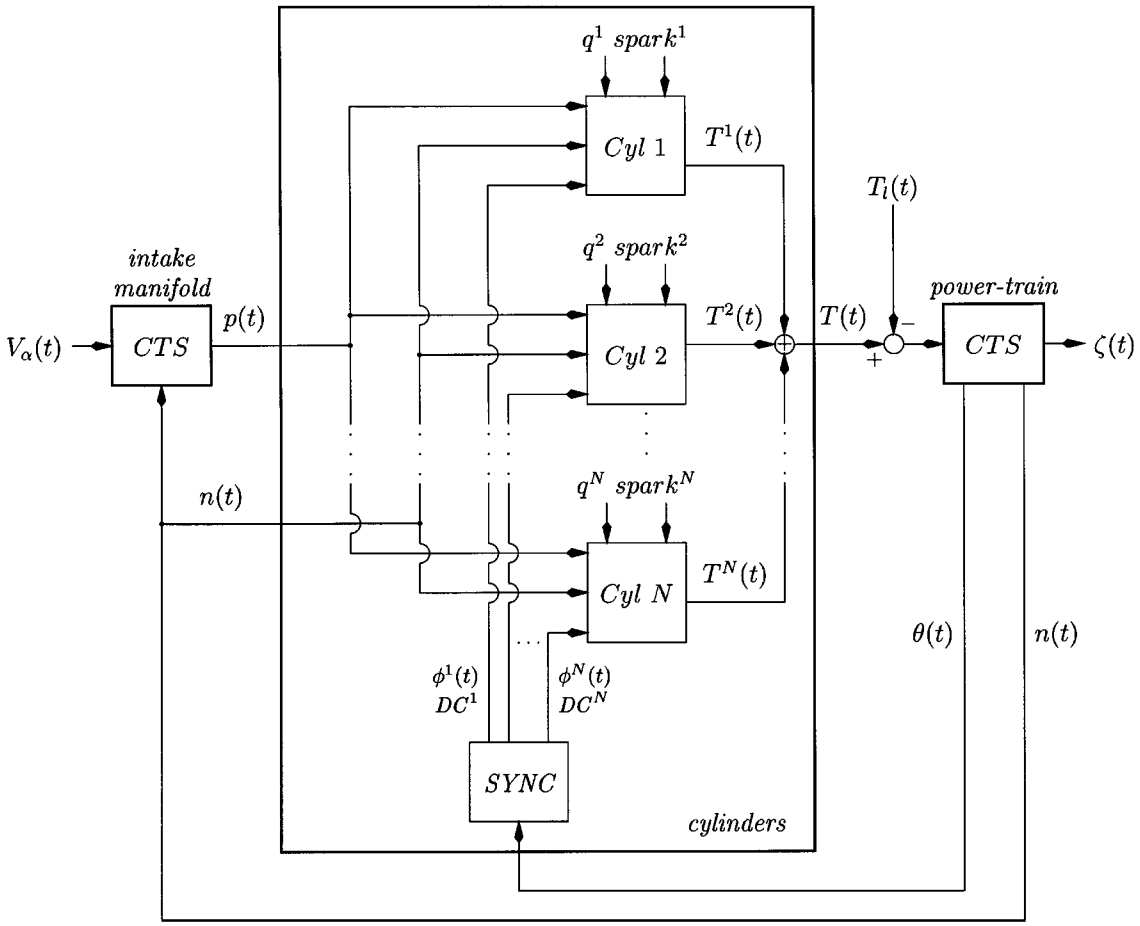


Fig. 7. Composition of cylinders models. For the sake of simplicity, in this figure tags are not reported.

5) The Actuators: Control algorithms for fuel injection and spark ignition must be executed at every cycle, synchronously with the crankshaft position θ. Both controls take time to actuate. Hence, we must decide the control actions sufficiently in advance to make sure that they are correctly delivered to the plant. In traditional non-GDI engines, injection typically is decided at the $H \rightarrow I$ transition to guarantee its completion by the end of the intake stroke. The spark is in general ignited with a different spark advance at every cycle. The value of the spark advance must then be computed at the $I \rightarrow BS$ transition, so that an appropriate actuator subsystem can be programmed to ignite the spark at the proper time.

To reflect these constraints on the control algorithms, the cylinder model is modified to include the effect of actuators delays. The FSM component of the hybrid model of the ith cylinder with actuator constraints is shown in Fig. 8. Assuming that the actuators apply exactly the desired spark advance and fuel injection, that is, $\varphi^i = \tilde{\varphi}^i$ and $q^i = \tilde{q}^i$, and since desired spark advance $\tilde{\varphi}^i$ is set at the $I \rightarrow BS$ transition, then the torque profile $T^i(t)$ during compression and expansion is completely determined at the $I \rightarrow BS$ transition. The torque profile is zero during both states BS and PA, and we can safely merge them into a single state C because we know the spark advance at the $I \rightarrow BS$ transition. Similarly, we can merge states NA and AS into the E state.

Using the same notation as for the model without actuators, at the transitions $I \rightarrow C$ and $C \rightarrow E$ the output is set to $o_k^i = I2C$ and $o_k^i = C2E$, respectively. The CTS function g_{C2E} represents the torque profile during expansion, assuming perfect actuation of the desired spark advance, that is

$$g_{C2E} = \begin{cases} G_f \tilde{q}^i \eta(\tilde{\varphi}^i), & \text{for } \phi^i \in [-\tilde{\varphi}^i, 180) \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The torque is zero during the other states, i.e., $g_{E2I} = g_{H2I} = g_{I2C} \equiv 0$. The DES describing the delay in the torque generation process has input signals $(\tilde{q}_r^i, t_r)$ at times $\{t_r\}$ at which the exhaust TDC is reached and signals (m_ℓ^i, t_ℓ), $(\tilde{\varphi}_\ell^i, t_\ell)$ at times $\{t_\ell\}$ at which the intake BDC is reached. The DES dynamics is as follows:

$$\begin{aligned} f_{H2I} &= h_{H2I} = (0, \tilde{q}_k^i, 0) \\ f_{I2C} &= h_{I2C} = z_k^i + (m_k^i, 0, \tilde{\varphi}_k^i) \\ f_{C2E} &= h_{C2E} = z_k^i \\ f_{E2H} &= h_{E2H} = z_k^i. \end{aligned} \tag{16}$$

V. Control Algorithm Design

Our (ambitious) goal has been since the beginning of this research project to develop control algorithms and their implementations for all the regions of operation described in Section III using the hybrid model of the engine described in

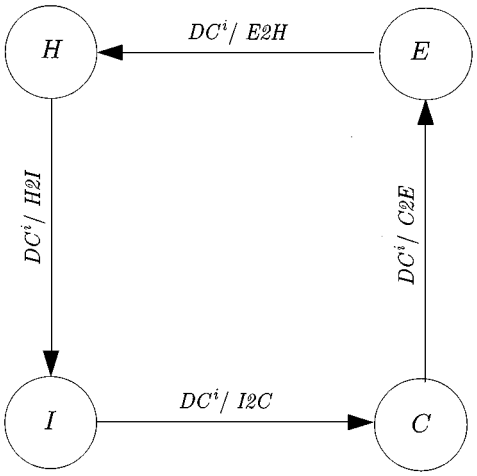


Fig. 8. FSM describing the behavior of the ith cylinder with actuators.

Section IV-B to yield a complete engine control subsystem. In this section, we describe our approach for three important subproblems that are considered the most difficult to solve: the Fast (Positive and Negative) Force Transient problem (FPFT and FNFT), the Cutoff problem, and the Idle Control problem.

To cope with the difficulties arising in the design of a controller in the hybrid domain, our approach consists of:

- "relaxing" each of the control problems of Section III by simplifying some components of the control problem (the cost function, the constraints, or the model of the plant) to obtain a solvable problem;
- using the solution of the relaxed problem as a "guideline" to devise a solution to the original problem, reintroducing the simplified components.

In particular, for the FPFT, FNFT, and Cutoff control problems, the hybrid plant model is relaxed to a continuous time one and the corresponding continuous time optimal control problem is solved by applying classical methods. The solution to the relaxed problem provides lower bounds for the minimum attainable cost for any hybrid control policy and suggests suboptimal control laws for the original problem. Solutions to the original problem are devised by mapping the solutions of the relaxed problems back to the hybrid domain. The solution in the hybrid domain is demonstrated to yield a behavior that is close (within a precisely specified bound) to the behavior of the control in the continuous case and, hence, even closer to the optimal solution in the hybrid domain.

In the idle speed control problem, we first find the set of all controllers that guarantee that the obtained closed-loop hybrid system satisfy the constraints (relaxed problem). Then, among those controllers, the one that minimizes a given cost function can be extracted. In both cases, we exploit the peculiarities of the control problem by reducing appropriately the complexity of the plant model.

A. Fast Force Transients

In the design of modern engine control, drivability requirements play an important role. In particular, longitudinal car oscillations represent one of the most critical aspects especially when fast torque changes are requested by the driver (tip-in and tip-out). To tackle this problem, active damping of power-train oscillations has been recently proposed in [48]–[51]. Damping of the oscillations can be achieved by modulation of the generated torque via throttle, fuel injection, and spark ignition control. The control strategies proposed in the literature are based on continuous-time models of torque generation and power-train dynamics. Oscillations are quantified in terms of either vehicle acceleration or vehicle jerk [52]–[54]. However, the closed-loop performance of such control strategies have not been deeply investigated when implemented on a real engine. Indeed, the discrete process of torque generation in a real engine may give rise to unpredicted behaviors that may lead to system instability. In our approach, modeling discrete phenomena in the engine behavior allows one to take into account the hybrid nature of the control problem, thus developing control laws for which closed-loop convergence is guaranteed.

1) Specialized Plant Hybrid Model: In this section, the hybrid model introduced in Section IV-B is specialized for control applications that require the control decision to take place at discrete points in time, namely, the TDC and BDC, consistently with the model in Fig. 8 presented in Section IV-B5 where actuator delays are taken into account. The combination of the FSM, the DES, and the CTS represents the torque generation mechanism for a single cylinder. In order to compensate for the actuation delays, the closed-loop control laws need to estimate the future state of the power-train corresponding to the time when the currently computed control values will effect the value of the produced torque. From (7), the active part of the torque is produced during the expansion phase after the combustion begins. It is then interesting to estimate the state of the power-train at the TDC between compression and expansion.

To simplify these predictions, the torque T^i is modeled as a constant value during the expansion phase and zero otherwise

$$g_{C2E} = G_f \tilde{q}^i \eta'(\tilde{\varphi}^i) \tag{17}$$

where

$$\eta'(\tilde{\varphi}^i) = \eta(\tilde{\varphi}^i) \min(180,\, 180 + \tilde{\varphi}^i)/180. \tag{18}$$

This is a slight difference with respect to the plant model shown in Fig. 8, where the torque T^i is produced during the expansion but only after the spark ignition. Indeed, the two torque generation models are equivalent for positive spark advance, while they differ for negative spark advance. However, in the latter case, the average torque is the same.

Since we are mainly interested in torque control, it is very convenient to reason directly in terms of desired ignition efficiency $\tilde{\eta}$ instead of using $\tilde{\varphi}^i$, the conversion between the two being expressed by (18).

The hybrid model for the case of a four-cylinder in-line engine is labeled $\mathcal{M}_{\text{fxft}}$.

2) Fast Force Transient Control: In the Fast (both Positive and Negative) Force Transient (FxFT) regions of operation, the objective is to steer the system from a given point, characterized by torque delivered to the crankshaft $T(0) = T_0$ to a new point with torque value T_R in minimum time. Since we consider drive-by-wire subsystems, available controls are: air mass that depends on the throttle angle, fuel injection, and spark advance. The control action must also satisfy comfort constraints. "Peak-to-peak" acceleration and jerk have been experimentally identified as the most important factors in passenger comfort. These quantities are the outputs of interest of system (11).

We formulate the FPFT (FNFT) control problem as: steer in minimum time the power-train state, keeping the jerk bounded $j(t) \in [0, j_{\max}]$ ($-[j_{\min}, 0]$, respectively) to a point such that the application of the new requested value of transmitted torque produces an oscillating acceleration evolution $\tilde{a}(t)$ such that $|\tilde{a}(t)| \leq \tilde{a}_{th}$ where $\tilde{a}_{th} > 0$ is a threshold of perception.

The dynamics of the oscillating acceleration $\tilde{a}$ can be isolated by applying natural mode decomposition to dynamics (11)

$$\dot{x} = A_x x + b_x(T - T_R) \tag{19}$$

$$\tilde{a} = c_x x \tag{20}$$

where x is two-dimensional and $A_x = \begin{bmatrix} \lambda & -\mu \\ \mu & \lambda \end{bmatrix}$. The third component (orthogonal) represents the dominant dynamics associated to the pole λ_1, $\dot{x_3} = \lambda_1 x_3 + b_{x_3}(T - T_R)$; its contribution to the oscillating acceleration may be neglected here. System trajectories under constant input $T(t) = T_R$ are hyper spirals in the x–x_3 space and spirals in the x space. Then, under constant torque $T = T_R$, constraint $|\tilde{a}(t)| < \tilde{a}_{th}$ is satisfied along system trajectories when the state is inside a target set given by a disk $B_{\tilde{\rho}}$ in the x state subspace and a cylinder $\mathcal{C}_x$ when considering also the third component of the state. The same cylinder is referred to as $\mathcal{C}_\zeta$, in the ζ coordinates state space.

Given a value T_0 of torque produced by the engine, let $\mathcal{I}(T_0) \subset Q \times Z \times W$ denote the invariant set in the hybrid state space (state of the FSM, DES, and CTS dynamics) given by points of the trajectory described by the hybrid model $\mathcal{M}_{\text{fxft}}$ during its evolution in the steady state corresponding to the torque T_0. Elements of the set W are triples (S, θ, P). In the sequel, only the design of a FPFT hybrid controller is illustrated. The FNFT controller design is analogous.

Problem 5.1.1: Given the engine hybrid model $\mathcal{M}_{\text{fxft}}$, find feedback control laws $\hat{\alpha}(\cdot)$, $\tilde{q}^i(\cdot)$ and $\tilde{\eta}^i(\cdot)$, for throttle plate position, quantities of fuel, and ignition efficiencies, which steer in minimum time $\mathcal{T}$ an initial state in $\mathcal{I}(T_0)$, with ζ-component $\zeta_0 \notin \mathcal{C}_\zeta$, to the boundary of the set

$$Q \times Z \times (\mathcal{C}_\zeta \times \mathbb{R} \times \{p_R\}) \tag{21}$$

satisfying the constraint

$$0 \leq j(t) \leq j_{\max} \qquad \text{for all } t \in [0, \mathcal{T}]. \tag{22}$$

a) FPFT problem relaxation to the continuous domain: In this section, the hybrid model $\mathcal{M}_{\text{fxft}}$ is relaxed to the continuous time domain. Consequently, the desired behavior is expressed by an optimal control problem which represents a relaxation of the original hybrid Problem 5.1.1. The relaxed problem assumes no constraint on torque signal $T(\cdot)$; it is only concerned with comfort requirements for minimum time optimal trajectories of the power-train dynamics to the set $\mathcal{C}_\zeta$.

Problem 5.1.2: Given the power-train dynamics (11), find a feedback control $\hat{T}(\cdot)$ which steers in minimum time an initial state $\zeta_0 \notin \mathcal{C}_\zeta$ to $\mathcal{C}_\zeta$ satisfying constraint (22) on the jerk $j(\cdot)$.

The minimum time control $\hat{T}$ to the manifold $\mathcal{C}_\zeta$ is obtained applying Pontryagin's maximum principle. The optimal solutions are made up of arcs of trajectory along which constraint (22) on the jerk is active. The set of points from which there exists a feasible trajectory to $\mathcal{C}_\zeta$ in the ζ-state space is partitioned into the sets S_ζ^0, $S_\zeta^{j_{\max}}$ where, respectively, $j = 0$ and $j = j_{\max}$. Hence, from (14), the minimum time torque is expressed as

$$\hat{T}(\zeta) = \begin{cases} (cb)^{-1}(-cA\zeta + cb_0), & \text{if } \zeta \in S_\zeta^0 \\ (cb)^{-1}(-cA\zeta + cb_0 + j_{\max}), & \text{if } \zeta \in S_\zeta^{j_{\max}}. \end{cases} \tag{23}$$

The boundaries of the sets $S_\zeta^{j_{\max}}$, S_ζ^0 are determined integrating trajectories backward in time using classical Pontryagin's methods; they are depicted in Fig. 9. See [55] and [40] for more details.

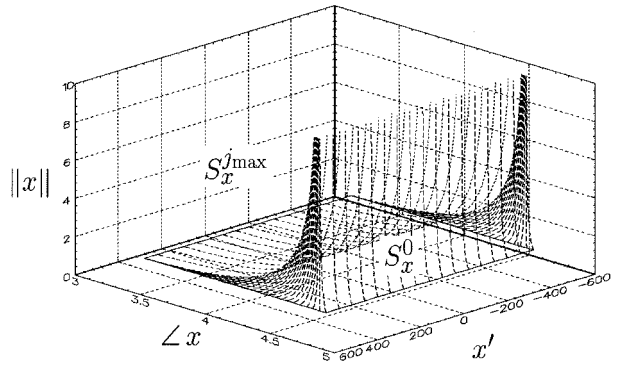
a) FPFT hybrid domain solution: The minimum time torque signal (23) is clearly not feasible for the hybrid model $\mathcal{M}_{\text{fxft}}$ in Section V-A1 [torque signals $T(\cdot)$ produced by $\mathcal{M}_{\text{fxft}}$ are piecewise-constant and synchronized with the FSM]. We derive in this section an implementable approximation of torque (23), consistent with the hybrid model $\mathcal{M}_{\text{fxft}}$, in terms of: throttle angle α, fuel quantities $\tilde{q}^i$, and ignition efficiencies $\tilde{\eta}^i$. The main difficulties are as follows.

a) The available torque is limited to the set of values $\{0\} \cup [G_f \eta_{\min} m_{\min}, G_f m_{\max}]$, where $m_{\min}$ and $m_{\max}$ depend on the air mass m, subject to manifold pressure dynamics.
b) Torque generation has to be synchronized with the power-train dynamics.
c) There is a delay between the time at which control signals $\tilde{q}^i$ and $\tilde{\eta}^i$ are set and the time at which the corresponding torque is generated.

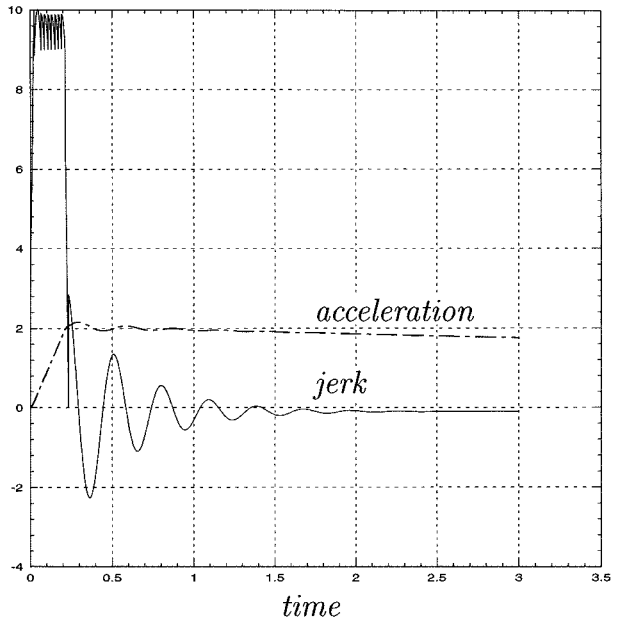
In order to properly set feedback signals $\tilde{q}^i$ and $\tilde{\eta}^i$, respectively, at times t_{k-2} and t_{k-1}, a prediction of ζ at time t_k is needed at both instants. Assuming that the power-train state is available to measurement, one- and two-step predictions are obtained by a forward integration of (11).

As illustrated in Fig. 1, the torque generation process can be viewed as a composition of interconnected systems. The impact of the driveline on air dynamics is weak enough to allow for a decentralized control, i.e., the feedback law for α and those for $\tilde{q}^i$ and $\tilde{\eta}^i$ can be devised independently (see [40] for details).

b) Air feedback design: A reference evolution $\hat{m}(t)$ for the air mass is obtained from the minimum time torque profile (23). A variable structure control (VSC) technique



(a)



(b)

Fig. 9. (a) Switching surfaces in the minimum time control to $\mathcal{C}_x$ in cylindrical coordinates. (b) Acceleration and jerk profile in the hybrid closed-loop system.

(see [56] and [57]) is used to achieve robust tracking of the reference $\hat{m}(t)$.

c) Fuel and ignition feedback design: A feedback control law in terms of fuel injection signals $\tilde{q}^i(k)$ and ignition efficiencies $\tilde{\eta}^i(k)$ is presented next. Torque law (23) is reformulated in the discrete time axis of the TDCs and BDCs as follows:

$$\hat{T}_d(k) = \begin{cases} (c\hat{b})^{-1}c(\hat{A}-I)\zeta(k), & \text{if } \zeta \in S_\zeta^0 \\ (c\hat{b})^{-1}c(\hat{A}-I)\zeta(k) & \\ \quad +(c\hat{b})^{-1}\tau_k j_{\max}, & \text{if } \zeta \in S_\zeta^{j_{\max}} \end{cases} \tag{24}$$

where $\tau_k = 30/n(t_k)$, $\hat{A} = e^{A\tau_k}$, and $\hat{b} = (\hat{A}-I)A^{-1}b$. Torque (24) produces a jerk with mean value equal to either 0 or $j_{\max}$, by construction. The actual jerk profile in the evolution of the hybrid model $\mathcal{M}_{\text{fxft}}$ exhibits a ripple on the average value. This is because $T(t)$ is piecewise-constant, then between two samples the natural modes of dynamics (11) evolve causing a jerk profile with values both below and above the desired $j = 0$ or $j_{\max}$. To avoid violating constraints (22), feedback laws for signals $\alpha(\cdot)$, $\tilde{q}^i(\cdot)$, and $\tilde{\eta}^i(\cdot)$ are modified to yield a torque $T^*(k)$ more conservative than (24). Fig. 9 shows an evolution of jerk and acceleration in the hybrid system $\mathcal{M}_{\text{fxft}}$ under the proposed feedback control law.

d) Convergence analysis and performances: The switching surface defined by (23) (Fig. 9) guarantees convergence in minimum time for the relaxed model (CTS). The synchronization constraints present in the hybrid model may affect the convergence as well as the performance of the closed loop system at low engine speeds. In Fig. 10, a continuous time optimal trajectory is depicted in the x-subspace; its approximation in the hybrid domain is depicted in Fig. 11 at low engine speed for different initial conditions. Comparing the two figures, it is clear how the hybrid nature of torque generation can produce instability in the closed-loop system even when the continuous time control law is perfectly stable. Having modeled the system in the hybrid domain, it is possible to analyze closed loop stability and also to synthesize the hybrid control law so as to guarantee convergence. The continuous time optimal solution is used as a guideline to achieve good performance.

It is possible to prove the following.

Proposition 5.1.1: There exists engine speed $n_{\min}$ such that if $n(t) > n_{\min}$, then the hybrid control $T^*(k)$ steers the hybrid system $\mathcal{M}_{\text{fxft}}$ to a hybrid state with ζ-component in $\mathcal{C}_\zeta$ in finite time.

The proof of the theorem is based on conservative approximations of the reachable sets.

In particular, for the parameters identified in our model and for the approximation used to obtain $T^*(k)$, the bound is $n_{\min} = 1205$ rpm. This bound is the result of a tradeoff between distance from optimality (in terms of transient time $\mathcal{T}$) and domain of convergence (range of engine speed for which convergence to $\mathcal{C}_\zeta$ is achieved). At high revolution, speed convergence is never a problem, because the continuous time solution can be tracked accurately. At low revolution speeds, it may happen that the torque signal fed into the driveline cannot be changed at the time required by the optimal continuous solution. Then the choice is whether to switch from $j = j_{\max}$ to $j = 0$ at the transition before that time or at the one after it. Switching before guarantees reaching $\mathcal{C}_\zeta$ for lower engine speeds, hence it increases the stability domain of the closed-loop system. Similarly, by anticipating the switching surface, it is possible to increase the domain of convergence. However, anticipating the switching surface increases the transient time $\mathcal{T}$, thus reducing performance of the closed-loop system.

The proposed solution is to modify the switching surface, anticipating it by an amount that depends on the engine speed n (one of the components of the ζ state). This allows an increase of the stability domain (to lower engine speeds) without giving up performance at high engine speed.

The optimal solution to Problem 5.1.1 is unknown; however, the optimal solution to Problem 5.1.2 attains a value for $\mathcal{T}$, which is a lower bound on the achievable transient time, by definition of relaxed problem. Similarly, the transient time

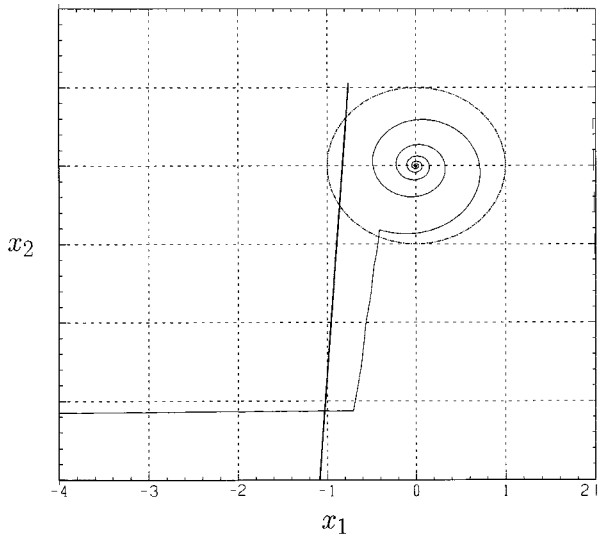


Fig. 10. Continuous time optimal trajectory in the x-subspace.

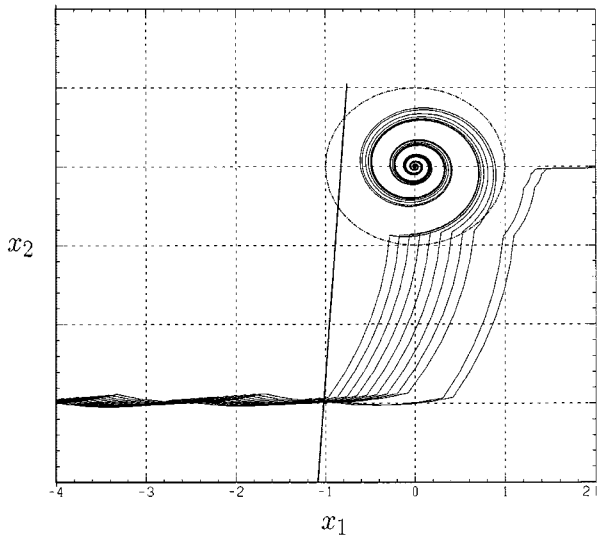


Fig. 11. Hybrid system trajectories at low engine speed.

of the proposed suboptimal solution to Problem 5.1.1 is an upper bound for the $\mathcal{T}$ attained by the optimal solution.

B. Cutoff Control

In this section, we consider a particular case of FNFT control design that corresponds to the complete release of the gas pedal by the driver. For cars equipped with electronic-throttle systems, this control problem can be formulated as a FNFT problem with $T_R = 0$, whose solution has been presented in Section V-A. However, for traditional cars, where the throttle valve is directly connected to the gas pedal and the engine control unit cannot control the mass of air loaded by the cylinders, a different approach has to be used. In the FNFT control problem, the jerk was constrained to be nonpositive, thus achieving zero oscillation (monotone acceleration) during the controlled operation. In the cutoff control problem, the available control is limited with respect to the electronic-throttle case and lower performance levels may be obtained. Indeed, in this case, the goal is to *cutoff* fuel injection *minimizing* passengers' discomfort, but we cannot in general prevent, like we did in the FNFT case, power-train oscillations even during controlled operation.

1) Problem Formulation: In a cutoff operation, after gas pedal release, the throttle valve reaches the maximum closure ($\alpha = \alpha_0 > 0$) and the sequence $m(k)$ of air intakes converges to the value m^o given by manifold pressure dynamics at the corresponding steady-state $p = p_0$. This results in an upper bound for the torque the engine can deliver. In the sequel, we denote by $\mathcal{M}_{\rm cut}$ the model obtained from $\mathcal{M}_{\rm fxft}$ with $\alpha = \alpha_0$ and $p = p_0$. The objective of cutoff control is to minimize the peak of the acceleration $\tilde{a}(t)$, given by (20), until it is below the threshold of acceleration perception $\tilde{a}_{\rm th}$. Recall that once the power-train state ζ is inside the region $\mathcal{C}_\zeta$ (with $T_R = 0$), introduced in Section V-A, fuel injection can be shut off with vehicle oscillations below threshold. Hence, the cutoff control is formulated as follows.

Problem 5.2.1: Given the engine hybrid model $\mathcal{M}_{\rm cut}$, find feedback control laws $\hat{q}^i(\cdot)$ and $\hat{\eta}^i(\cdot)$, for the fuel injection and spark modulation factors, which steer in an unspecified finite time $\mathcal{T}$ an initial state in $\mathcal{I}(T_0)$, with ζ-component $\zeta_0 \notin \mathcal{C}_\zeta$, to the boundary of the set $Q \times Z \times (\mathcal{C}_\zeta \times \mathbb{R} \times \{p_0\})$, minimizing $\sup_{0 \le t \le \mathcal{T}} |\tilde{a}(t)|$.

Note that the target set is the same as in (21), because now the target pressure p_R and the initial pressure p_0 coincide with the steady-state value. Moreover, the transient time $\mathcal{T}$ is a design variable and is constrained to be finite; its value is given by the solution to Problem 5.2.1, i.e., it is a value that minimizes the peak of acceleration.

2) Control Synthesis:

a) Problem relaxation: The control problem can be simplified considerably by relaxing the plant hybrid model $\mathcal{M}_{\rm cut}$ to the continuous-time domain, thus removing all synchronization constraints, as we did in the FxFT case. System dynamics are limited to the power-train oscillating dynamics (19), with torque T bounded to belong to $[0,\ M]$ with $M = Gq_{\max}$ and $q_{\max} = 13.9^{-1}m^o$. Denoting by $B_{\hat{\rho}}$ the projection of $\mathcal{C}_x$ on the oscillating mode subspace (i.e., $\|x\| \le \hat{\rho} = \tilde{a}_{\rm th}\|c_x^T\|^{-1}$), the following relaxed problem is considered.

Problem 5.2.2: Given the powertrain dynamics (19), find a feedback control $\hat{T}(\cdot) \in [0,\ M]$ that steers in an unspecified finite time $\mathcal{T}$ an initial state $x_0 \notin B_{\hat{\rho}}$ to $B_{\hat{\rho}}$, minimizing $\sup_{0 \le t \le \mathcal{T}} |\tilde{a}(t)|$.

In [58] and [24], the authors derived an optimal feedback law solution to Problem 5.2.2. Since the objective function is nondifferentiable, standard tools, such as Pontryagin's principle, cannot be applied and optimality is proven by comparing the closed-loop trajectories with the reachable sets. Fig. 12 shows a typical closed-loop phase space under the proposed control. The optimal torque $\hat{T}$ is bang-bang outside the sector $D_{\mathcal{M}}$ of the circle $C_{\mathcal{M}}$, delimited by the line $v^T x = 0$ passing through $x_M = -A_x^{-1}b_x M$ [with $C_{\mathcal{M}}$ containing (0, 0) and x_M]. Indeed, outside sector $D_{\mathcal{M}}$, the available torque T is not strong enough to counteract the elasticity of the drive-line, and the torque switches between the bounds 0 and M, when x crosses the line $v^T x = 0$.

b) Hybrid system solution: The relaxed-problem optimal torque $\hat{T}$, obtained with the continuous time model, is clearly not feasible for the engine hybrid model $\mathcal{M}_{\text{cut}}$, whose inputs are fuel quantities q^i and spark modulations η^i. Indeed, torque generation mechanism constraints described in Section V-A2, i.e., a) bounds, b) synchronization, and c) delay, have to be taken into account. In the hybrid feedback proposed in [24], only the two extreme torque values 0 and M are used, obtained by setting $\eta^i = 1$ and either $q^i = 0$ or $q^i = q_{\max}$. Synchronization of the generated torque with the power-train dynamics is the main difficulty. To compensate the torque generation two-step delay, a prediction $\tilde{x}_2(k)$ of $x(t_{k+2})$ is obtained from $x(t_k)$ by forward integration of (19). The injection control is defined according to a switching curve $\sigma(x) = 0$, with σ: $\mathbb{R}^2 \to \mathbb{R}$, as follows:

$$\hat{q}^i(k) = \begin{cases} 0, & \text{if } \tilde{x}_2(k) \in B_{\hat{\rho}} \\ \begin{cases} 0, & \text{if } \sigma(\tilde{x}_2(k)) \geq 0, \\ q_{\max}, & \text{if } \sigma(\tilde{x}_2(k)) < 0, \end{cases} & \text{if } \tilde{x}_2(k) \notin B_{\hat{\rho}}. \end{cases} \tag{25}$$

Since the power-train evolution $x(t)$ is known only at times t_k, the hybrid system may fail to converge if $x(t_k)$ jumps around $B_{\hat{\rho}}$. The bang-bang switching line of optimal feedback $\hat{T}$ is $v^T x = 0$. If $\|x_M\| \leq \hat{\rho}$, control law (25) with $\sigma(x) = v^T x$, has been proven to yield closed-loop hybrid trajectories that converge to $B_{\hat{\rho}}$, when engine speed is larger than an appropriate bound. However, since for commercial cars $\|x_M\| > \hat{\rho}$, $\sigma(x)$ has to be modified to guarantee convergence.

c) Convergence and performance analysis: The switching curve $\sigma(x) = 0$ that guarantees convergence in the closed-loop hybrid model is shown in Fig. 12. This curve consists of the following four pieces (for increasing x_1): a half-line lying on $v^T x = 0$, an arc of the circle on the boundary of $B_{\hat{\rho}}$, the arc $\psi_c^{(2)}$, and a half-line lying on $v^T x = 0$. The arc $\psi_c^{(2)}$ is a piece of a trajectory of the system (19), under torque $T = M$ and passing through a point $x_c^{(2)}$ belonging to the boundary of $B_{\hat{\rho}}$. Details are reported in [24]. Convergence of the hybrid system is guaranteed by the following proposition.

Proposition 5.2.1: Let $\|x_M\| > \hat{\rho}$. There exists engine speed $n_{\min}$ and a point $x_c^{(2)}$ such that if $n(t) > n_{\min}$, then the hybrid control (25) steers the hybrid system $\mathcal{M}_{\text{cut}}$ to a hybrid state with x-component in $B_{\hat{\rho}}$ in finite time.

The proof reported in [24] is constructive and gives $n_{\min}$ and $x_c^{(2)}$, which defines $\sigma(x)$ in (25).

An optimal solution to Problem 5.2.1 is unknown. However, bounds on the performance can be obtained by comparing the relaxed problem optimal solution to the proposed hybrid problem solution. The performance is measured by the value of the cost function, namely

$$\tilde{A} = \sup_{0 \leq t \leq \mathcal{T}} |\tilde{a}(t)|.$$

The optimal cost $\tilde{A}_{hyb}(x_0)$ corresponding to the unknown optimal solution to Problem V-B-2 can be bounded by comparing the relaxed problem optimal solution cost $\tilde{A}_{rel}(x_0)$ to the cost $\tilde{A}_{ctrl}(x_0)$ attainable by the proposed hybrid control,

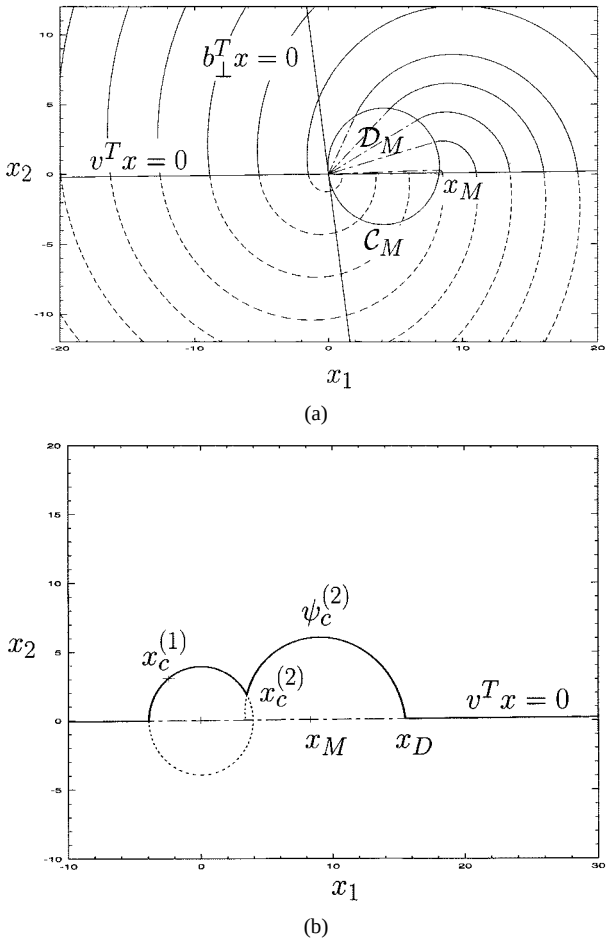


Fig. 12. (a) Optimal trajectories for the relaxed problem. If $x \notin \mathcal{D}_M$, either $T = M$ (solid line) or $T = 0$ (dashed line); otherwise, $0 < T < M$ (dash-dot line). (b) The switching line $\sigma(x) = 0$, which defines the hybrid control, when $\|x_M\| > \rho$.

for the same initial condition x_0, thus assessing the quality of the approach. In [24], a function $\Omega(\cdot)$ was derived for which we have the following.

Proposition 5.2.2: Let $\|x_M\| > \hat{\rho}$, and assume $n(t) > n_{\min}$, with $n_{\min}$ given by Proposition 5.2.1 If $(c_x x_M)(c_x b_x) > 0$, there exists $\rho_\Omega > 0$ such that, for any x_0, with $\|x_0\| > \rho_\Omega$, we have

$$\begin{aligned} \tilde{A}_{rel}(x_0) \leq \tilde{A}_{hyb}(x_0) \leq \tilde{A}_{ctrl}(x_0) \\ \leq \tilde{A}_{rel}(x_0) + \Omega(\rho_\Omega, \omega_{\min}) \end{aligned}$$

Further, if x_0 satisfies either

$$(v^T x_0 < 0) \wedge ((c_x x_M)(c_x A_x)^T (x_0 - x_M) < 0)$$

or

$$(v^T x_0 > 0) \wedge ((c_x x_M)(c_x A_x)^T x_0 > 0)$$

then the hybrid control (25) is optimal for Problem 5.2.1.

Proposition 5.2.2 describes the performance of the closed-loop system. In particular, it identifies sets of initial conditions such that the closed-loop system reaches $B_{\hat{\rho}}$ along an optimal trajectory. It also provides an upper bound for the degradation in terms of acceleration peak, which applies to trajectories originating at points outside the region of optimal initial conditions.

3) Experimental Results: The proposed cutoff control strategy has been implemented and tested at Magneti–Marelli Engine Control Division on a commercial car, a 16-valve 1400-cc-engine car. The engine control electronics is a 4-LV Magneti Marelli on-board computer based on a 25-MHz 32-bit Althair Motorola microprocessor with fixed-point arithmetic unit. By Proposition V-B-1, convergence is guaranteed for engine speeds greater than $n_{\min} \simeq 396$ rpm, less than minimum engine speed to prevent engine stalls. A discrete-time Luenberger observer has been shown to perform satisfactorily for power-train state estimation. In Fig. 13, the performance achieved by the proposed cutoff strategy is compared with the performance of a currently implemented open-loop strategy. On the left, the evolutions in the $\hat{x}$ subspace of the observer are reported, along with the switching curve $\sigma(x) = 0$ and the target set $B_{\hat{\rho}}$. On the right, the resulting evolutions of the oscillating component $\tilde{a}(t)$ of the acceleration are shown. With the proposed control strategy, the state is steered to $B_{\hat{\rho}}$ with no encirclement and $\tilde{a}(t)$ monotonically decreases to a_{th}. As expected from the theoretical results, once injection is set to zero permanently, $\tilde{a}(t)$ remains bounded within the perception threshold. In the open-loop strategy, an encirclement of $B_{\hat{\rho}}$ produces a peak of the acceleration.

C. Idle Speed Control

In the *Idle* operation mode [59]–[62], the gear is set to neutral and the engine speed should be maintained as close as possible to a reference value $n = n_0$. The goal is to design a feedback controller that keeps the speed of the crankshaft in a specified range, $n_0 \pm \Delta n$, robustly with respect to two sources of disturbances: 1) load torque on the crankshaft due to subsystems such as air conditioning and servo-mechanisms for steering and braking and 2) the inertial load increase that occurs when the driver releases the clutch pedal. The cost function to minimize is the energy of the variations of the engine speed with respect to the reference value.

A survey on different engine models and control design methodologies for idle control is given in [59]. Both time-domain (e.g., [60]) and crank-angle domain (e.g., [61]) models have been proposed in the literature to solve the idle management problem. These models deal with the average value of engine speed. More recently, cycle-accurate models, which describe periodic speed variations due to torque fluctuations, have been investigated [47].

Several design techniques have been applied to the idle control problem, such as multivariable control [63], ℓ_1 control [60], H_∞ control [64], μ-synthesis [65], sliding mode control [66], and LQ-based optimization [67].

In our approach, the adoption of a hybrid formalism allows us to represent the cyclic behavior of the engine, thus capturing the effect of each spark command on the generated torque, the interaction between the discrete torque generation and the continuous power-train and air dynamics, and the discrete changes in the parameters of the power-train model. In [61] and [47], crank-angle domain models are used. In these models, the spark command is represented at the previous TDC to take into account the actuators delay. Here instead we concentrate on the model of Fig. 4, where only the primary inputs to the engine are considered and actuator delays are ignored: the spark command is supposed to coincide with actual spark ignition.

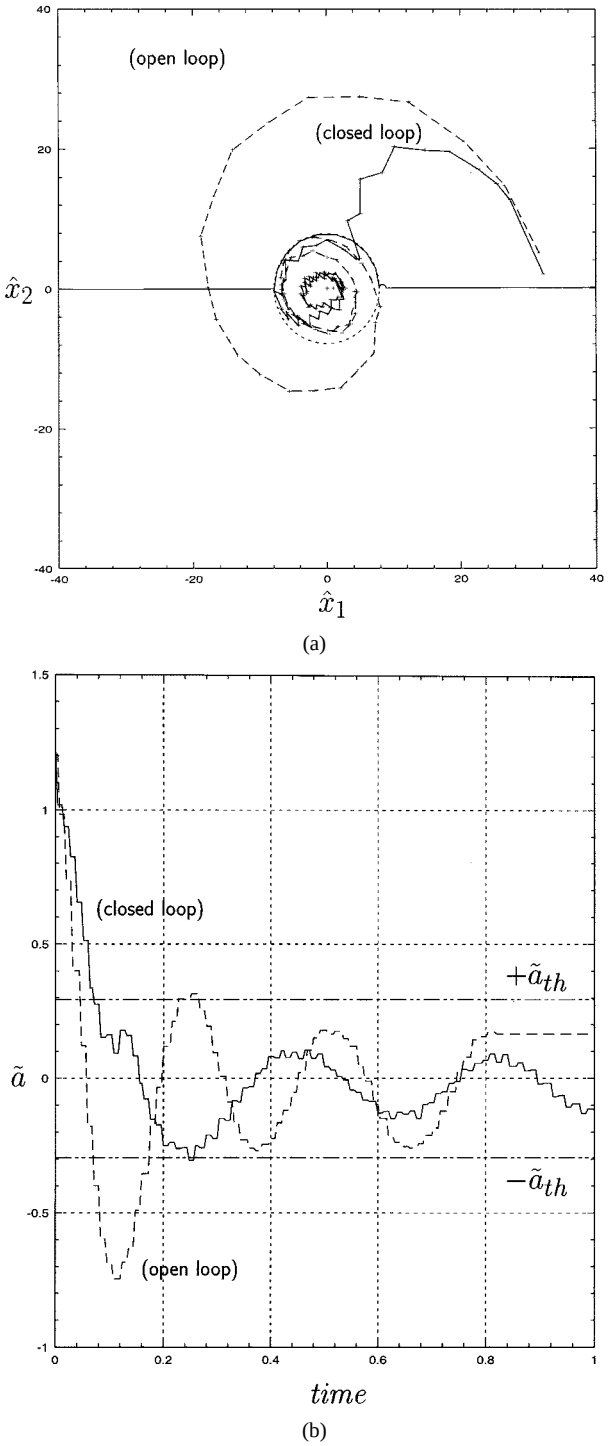


Fig. 13. Observer subspace $\hat{x}$ evolution with target set $B_{\hat{\rho}}$ and switching line $\sigma(x) = 0$ and (b) oscillating acceleration (in m/s^2) $\tilde{a}(t)$ signal profiles in a controlled cutoff (solid line) and an uncontrolled cutoff (dashed line).

1) Problem Formulation: The plant hybrid model presented in Section IV-B has been simplified by:

- ignoring details that are unimportant for the *Idle* region of operation (see Section III);

- considering only four cylinder engines;
- imposing constraints on the fuel mix.

In particular, the following assumptions have been made.

- Fuel injection is regulated by an inner control loop to maintain the air-to-fuel ratio to stoichiometry. This control injects in each cylinder the quantity $q^i = m^i/14.64$ of fuel, so that q^i is no longer an available input. In (7), the functions g_{PA2AS}, g_{BS2AS}, and g_{NA2AS} modeling the generated torque can be rewritten as

$$Gm^i\eta(\varphi^i) \tag{26}$$

  where $G = G_f/14.64$.
- Manifold pressure dynamics (1) are linearized about the idle equilibrium point and α is considered as the input. This simplification is justified by the fact that, by specification, the crankshaft speed excursion is bounded to belong to a close interval about the idle value, while the pressure evolution and the throttle position remain close to their idle values. Then, by linearizing (5), we obtain

$$m^i(t) = k_m p(t).$$

- As discussed in Section IV-B, the spark advance φ^i is bounded to avoid both knocking and misfiring. Hence, at the idle crankshaft speed, we consider as feasible ignition times the ones *from* 20^0 *before the top dead center to* 15^0 *after*. Moreover, we abstract away the delay of the spark ignition subsystem (actuator) and we act on the spark directly.
- The gear position is fixed in neutral. Consequently, the gear does not generate any discrete transitions in the power-train dynamics and the secondary drive-line is disconnected from the engine. Hence, the subdominant oscillatory modes of the power-train dynamics (11) can be neglected and the power-train evolution is described only in terms of the crankshaft speed n. Due to the actions of the driver on the clutch pedal, the first part of the drive-line is either connected or disconnected from the engine. This changes, in a discrete way, the dynamic parameters of the system

$$\begin{cases} \dot{n}(t) = a_n n(t) + b_n(T(t) - T_l(t)), \\ \qquad \text{if the } clutch \text{ is open} \\ \dot{n}(t) = a_n^L n(t) + b_n^L(T(t) - T_l(t)), \\ \qquad \text{if the } clutch \text{ is closed} \end{cases} \tag{27}$$

  where $a_n = -B/J, b_n = 1/J, a_n^L = -(B+B')/(J+J')$, and $b_n^L = 1/(J+J')$, with $J, B\,(J',\ B')$ denoting, respectively, the inertial momentum and the viscous friction coefficient of the crankshaft primary drive-line. If the crankshaft and the primary driveline rotate at different velocities, when the clutch pedal is released, a torque impulse is generated to even the two velocities, the final engine speed being an intermediate value of the two. This effect, however, has been neglected in this model to simplify the computation.

The cylinders behavior for a four-cylinder in-line engine can be represented using only three discrete states. In fact, the engine kinematics are such that, at any time, each cylinder is in a different stroke of the cycle and $\phi(t) = \phi^1(t) = \phi^2(t) = \phi^3(t) = \phi^4(t)$ for every t. Since the cylinders have the same behavior when they are in the same stroke, we can reorder the cylinders labeling at each cycle to obtain the configuration (I, C, E, H). The behavior of the cylinders can then be described by the three states $S = (I, BS, AS, H)$, $S_+ = (I, PA, AS, H)$, $S_- = (I, BS, NA, H)$, denoting all the possible cylinders configurations up to a reordering. Note that (I, PA, NA, H) is not feasible with the ignition constraints on φ^i since it would correspond to the case of the cylinder in the compression stroke receiving the spark before the cylinder in the expansion stroke.

In Fig. 14, the transitions in the compacted model of the cylinders are compared to those of the single cylinder FSM in Fig. 4. In state S, the cylinder in the expansion stroke has received the spark and is generating torque (AS), and the cylinder in the compression stroke has not yet received the spark command (BS). If spark ignition occurs before the next DC, then the cylinder that is in the compression stroke enters state $PA(BS \to PA)$, while the others remain in the same state. This corresponds to the transition from S to S_+. Otherwise, if the DC is reached before the spark is ignited, all the cylinders change phase.

- The one that was in the intake stroke enters the compression stroke $(I \to BS)$.
- The one that was in the compression stroke enters the expansion stroke and keeps waiting for the spark $(BS \to NA)$.
- The one that was in the expansion stroke enters the exhaust stroke $(AS \to H)$.
- The one that was in the exhaust stroke starts the intake process $(H \to I)$.

This corresponds to the transition from S to S_-. When the spark is ignited at the DC, all the cylinders change phase and a self-looping transition from S to S takes place.

In state S_+, the spark command has been given for both cylinders in the compression stroke and in the expansion stroke, and at the DC all the cylinders change phase so that the transition from S_+ to S takes place. In state S_-, both cylinders in the expansion and compression strokes are waiting for the spark command. No torque is generated in this case. When the spark ignition is given (necessarily before the next DC), the cylinder that is in the expansion stroke changes from NA to AS, and the transition from S_- to S takes place.

The overall engine and power-train FSM in the idle operation mode is shown in Fig. 15, where the discrete changes of the power-train due to clutch motion are taken into account.

The discrete state s takes value in the finite set $\{S_-, S, S_+, S_-^L, S^L, S_+^L\}$, where in S_-, S, S_+ the clutch is open, while in S_-^L, S^L, S_+^L it is closed. The transitions between the six modes are readily obtained

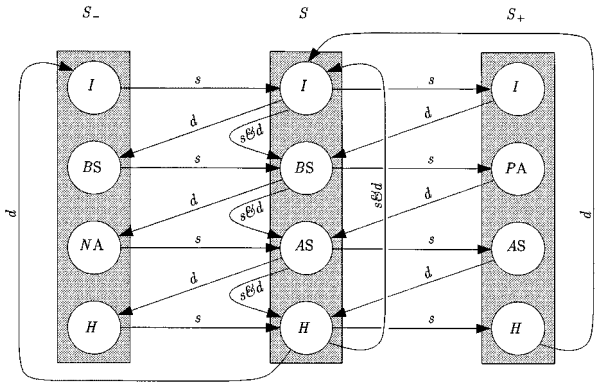


Fig. 14. Four-stroke engine phases versus cylinders phases: s denotes transitions due to spark ignition, d those due to the reaching of a DC, and $s\&d$ those due to spark ignition occurring at a DC.

extending those between states S_-, S, S_+ with clutch position changes. The input u takes value *spark*, when a spark is given to any cylinder, DC when a dead center is reached, and either *on* or *off* when the clutch is open or closed, respectively. The values $DC\&spark$, $on\&DC$, ... represent simultaneous actions.

The four DESs modeling the discrete delays in the torque generation can also be compacted. Indeed, for four-cylinder engines, the dimension of the DES state z can be reduced to three. Since the four cylinders evolve synchronously, then to describe the torque delay it is sufficient to store the spark advance φ corresponding to the last spark ignition, the mass of air m^i trapped at the end of the intake run in the ith cylinder starting the compression run (denoted by m_C), and the mass of air m^j trapped in the jth cylinder starting the expansion run (denoted by m_E). Hence, $z = (\varphi, m_C, m_E)$. The DES dynamics and output functions f_{o_k} and h_{o_k} are readily obtained by appropriately modifying those in (9). Since the torque profile is piecewise constant, then the DES produces as an output the torque value T_k generated by the engine in the next phase according to (26), where m^i is either m_C, for positive spark advance, or m_E, for negative spark advance. Then, the CTS that generates the continuous profile of the torque is simply a zero-order holder, with state T.

The overall engine and power-train hybrid model $\mathcal{M}_{\text{idle}}$ in the idle operation mode is shown in Fig. 16 using the TSM formalism. The hybrid state of model $\mathcal{M}_{\text{idle}}$ is $(s, z, x) = (s, (\varphi, m_C, m_E), (n, \theta, T))$.

The informal control problem specification, "keep the crankshaft speed within a specified range while minimizing the energy of its variation with respect to the reference speed," is formalized as follows.

Problem 5.3.1: Given the engine hybrid model $\mathcal{M}_{\text{idle}}$, find the feedback control laws for the spark timing spark$(\cdot)$ and throttle valve $\alpha(\cdot)$ that minimize

$$\max_{\substack{clutch\in(on,off)\\ T_l\in[0,\,T_l^{\max}]}} \int (n(t)-n_0)^2\,dt$$

subject to $n(t) \in [n_0 - \Delta n, n_0 + \Delta n]$.

2) Control Synthesis: The relaxation of the control problem is obtained by removing from Problem 5.3.1 the cost function and considering only the constraint imposed on the crankshaft speed. Thus, we convert the problem from selecting an optimum control law to one of finding the control laws that satisfy the constraints.

Problem 5.3.2: Given the engine hybrid model $\mathcal{M}_{\text{idle}}$, find all the initial hybrid states (s, z, x) such that there exist control strategies for the spark timing $spark(\cdot)$ and throttle valve $\alpha(\cdot)$, which keep the crankshaft speed n in the range $[n_0 - \Delta n, n_0 + \Delta n]$, independently of the two disturbances: clutch and $T_l \in [0, T_l^{\max}]$.

The resulting problem belongs to the general class of *safety specification problems*, described by giving a set of *good* hybrid states within which the closed-loop system must stay. The set of all initial states satisfying this invariance property is called the *maximal safe set* and corresponds to the maximal controlled invariant set contained in the set of good hybrid states. The set of all control strategies that make this set invariant is called *maximal controller*.

A systematic procedure for the solution of safety specification problems for a hybrid system has been proposed in [68] and [29]. The interaction between the controller and a nondeterministic hybrid plant is viewed as a two-person zero-sum game. Each player moves by setting both discrete and continuous control inputs. The controller wins if it keeps the state of the system within the specified set of good states; its adversary environment tries to force the system outside this set. The maximal safe set is obtained by first over approximating it with the set of all good states. The synthesis procedure iteratively prunes away the set of states from which the environment wins via either one additional discrete step, or one additional continuous flow. If the procedure terminates, the maximal safe set remains determined.

For the idle regime control problem, the set of good states is specified by $n \in [800 \pm 30]$, the control actions are *spark* and α, and the environment actions are *clutch* and T_l. In the hybrid engine model, there is a discrete step every time a spark command is given, a dead point occurs, or the clutch position is changed. The continuous evolutions between two discrete steps is determined by the air and crankshaft dynamics, which depend on the controller input α, the generated torque T, and the load torques T_l. The continuous dynamics of the crankshaft position θ locates the dead center occurrences.

The maximal safe set is shown in Fig. 17, limited to the modes S, S_+, S^L, S_+^L. The figure presents some projections of the maximal safe set to three dimensional subspaces with fixed specific values of the remaining components.

This result for the idle control is the first of its kind and allows us to determine tightly the maximum range of allowed torque disturbances, given the maximum interval of angular speed possible values. Indeed, considering $T_l^{\max}$ as a parameter, we can determine the maximum value of $T_l^{\max}$ for which a nonempty maximal safe set exists. Butts *et al.* [60] obtained a sort of dual result: by synthesizing a robust ℓ_1 controller for a discrete-time model of the engine, they achieved the minimum excursion of crankshaft speed under the action of bounded torque load, for the system initially at rest. The use of a hybrid framework, where discrete and continuous

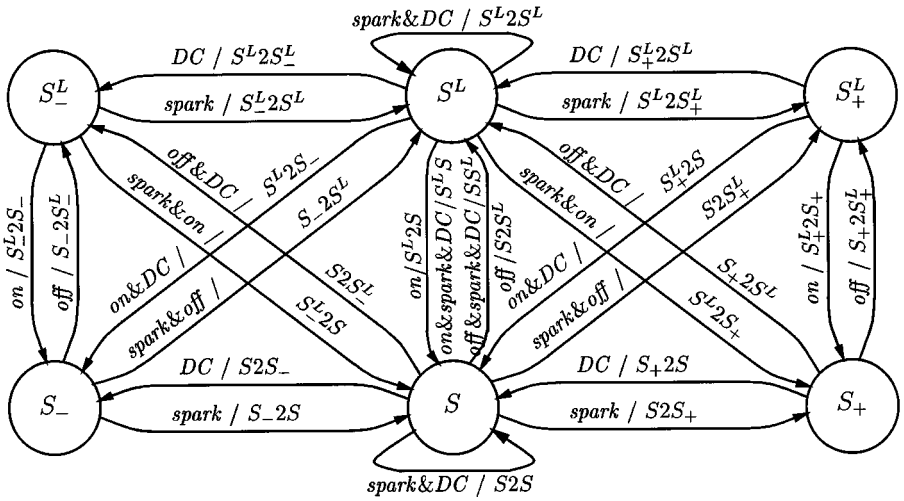


Fig. 15. FSM for the engine running at idle.

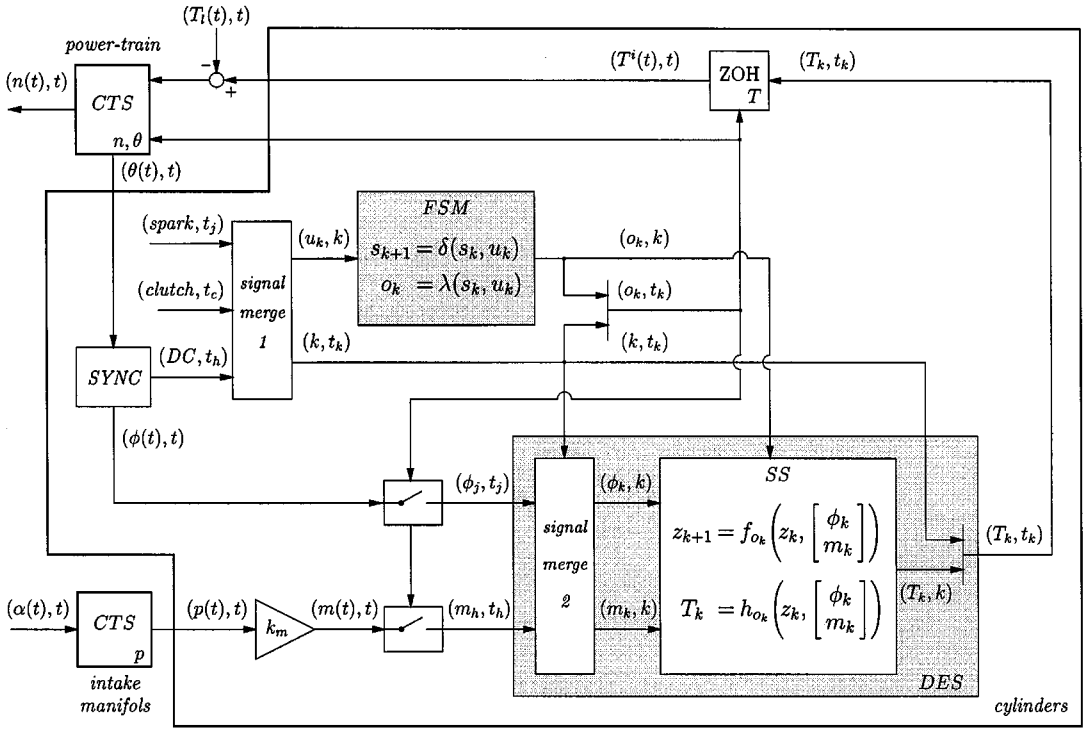


Fig. 16. TSM hybrid model $\mathcal{M}_{\text{idle}}$ for the engine and the power-train running at idle.

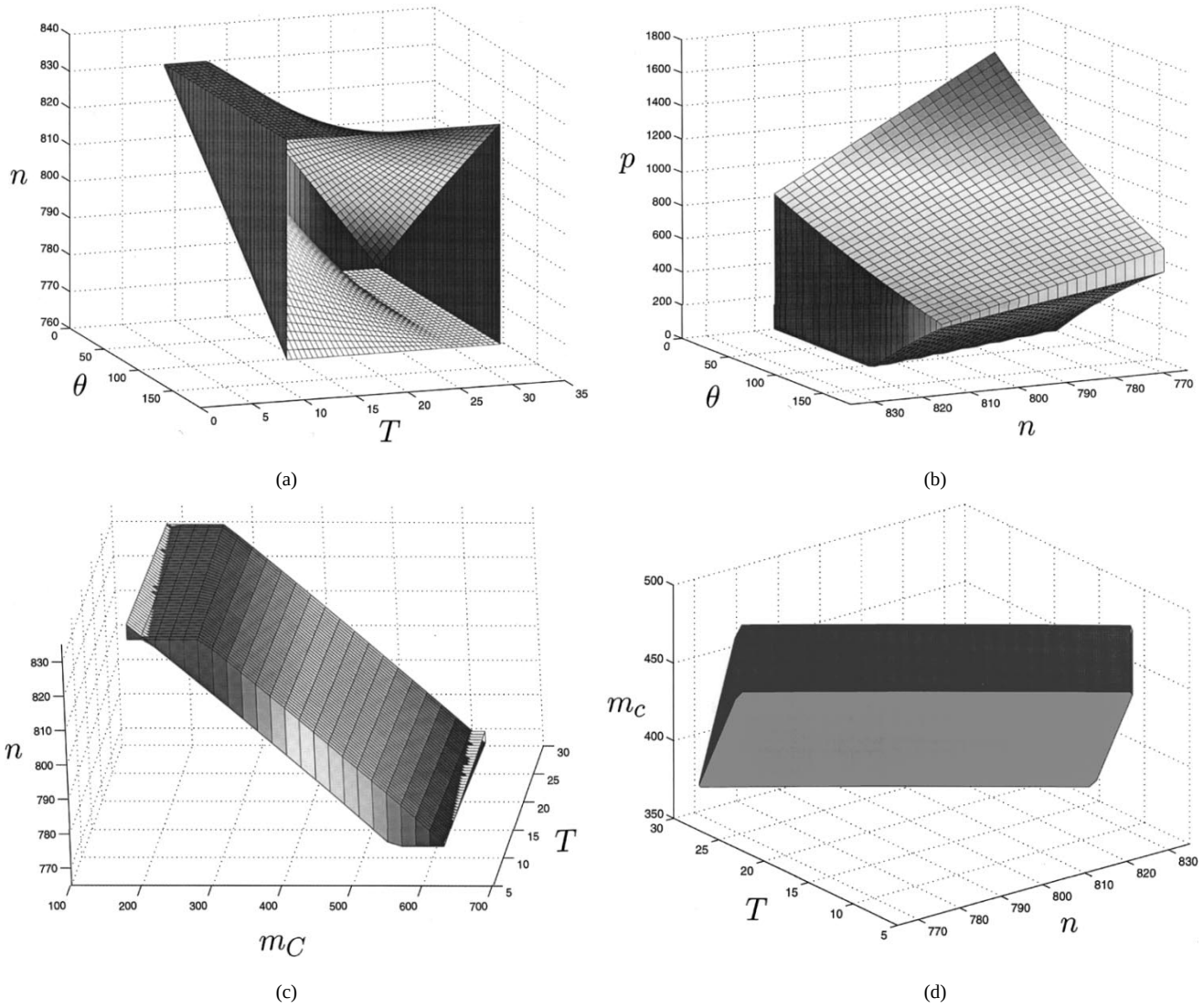
signals are modeled in a separate but integrated manner, is a definite advantage over other approaches, which approximate the system by converting it to continuous [47], or discrete sampled [61] representations, thus obtaining solutions whose properties are not guaranteed.

VI. Conclusion

We presented the application of hybrid system techniques to an important industrial domain: automotive engine and power-train control. We argued that while, in the past, average models were successfully used to describe the behavior of the engine, the ever increasing demands on drive comfort, safety, emissions, and fuel consumption imposed by car manufacturers require cycle-accurate models, which can only be described using a hybrid formalism.

We reviewed our work on hybrid system and engine control by presenting

- a hybrid reference model that unifies the requirements, the constraints, and the behavior of the engine and the power-train under different operating conditions;
- a hybrid model of the engine and of the power-train based on three components: a component describing the mechanical behavior of the power-train itself with ODEs, one describing the intake manifold dynamics with ODEs, and one describing the behavior of the



(a) (b) (c) (d)

Fig. 17. Projections of the maximal safe set for the relaxed idle control problem.

engine and the torque generation mechanism with interacting finite-state machines, discrete event systems, and continuous time systems;
- the derivation of hybrid control laws for fast transients with their stability and optimality properties;
- the determination of the maximal controller for idle regime management.

We demonstrated that this approach can be used to implement a control system by showing experimental results on an instrumented car for the cutoff problem. It is our intention to complete the implementation of all control modes expressed in our reference model.

As part of the design methodology we provide guidelines for the use of the specification models introduced in Section III, for the derivation and validation of control laws, as well as for their implementation as software components running on industrial platform architectures [19]. Our industrial partners are fully engaged in this project given its potential benefits. The entire design methodology and the various roles played by our industrial partners will be fully disclosed in a special session of the upcoming Convergence 2000 Conference in Detroit.

Even if much has been accomplished, we are still far from exhausting the mine of interesting problems. We are now engaging, among others, in the following aspects of power-train control.

- Diagnosis and recovery are now a substantial part of the tasks an electronic control subsystem has to carry out. They both require a hybrid formalism. In fact, sensors, actuators, and subsystem failures typically produce discrete changes in the system parameters or configuration. If these changes are not properly handled in a hybrid formalism, the diagnosis algorithms are not guaranteed to converge. Further, the design of a recovery procedure, under which the plant evolves in a degraded mode, may need to use a hybrid formalism to handle a change of functionality of either a sensor or an actuator due to its failure or the failure of others.
- Hybrid engines, where a standard combustion engine is coupled with an electric motor to provide optimal

power consumption and minimal emissions in city driving conditions.
- Electronic valves, where the valve opening and closing are de-linked from the piston phases and controlled independently.

ACKNOWLEDGMENT

The support and continuous encouragement of Dr. Pecchini, President and General Manager of Magneti Marelli Powertrain Division, and his staff is gratefully acknowledged. Dr. C. Rossi, Dr. G. Gaviani, R. Flora, G. Serra, and their teams at Magneti Marelli have introduced the problem and discussed its solution with the authors. Dr. P. Popp and T. Demmler at the BMW Research Center in Palo Alto, Dr. Ziebart, member of the Board of Management of BMW AG, Research, Development and Purchase, Dr. Thoma and his staff, and, in particular, J. Ehret of BMW, Munich, provided insight and discussed actively automotive electronics problems. The authors also acknowledge the contributions of A. Ferrari to the design methodology, L. Lavagno to the use of MOCs and to the methodology, E. Lee to the TSM framework, S. Sastry for his support in the theory of hybrid systems, R. Newton for his support as director of the GSRC, and M. Broucke for her careful reading of the manuscript.

REFERENCES

[1] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds., *Hybrid Systems*. Berlin, Germany: Springer-Verlag, 1993, no. 736, Lecture Notes in Computer Science.

[2] P. Antsaklis, A. Nerode, W. Kohn, and S. Sastry, Eds., *Hybrid Systems II: Proceedings of the 1994 Workshop on Hybrid Systems and Autonomous Control*. Berlin, Germany: Springer-Verlag, 1995, no. 999, Lecture Notes in Computer Science.

[3] R. Alur and T. A. Henzinger, Eds., *Hybrid Systems III*. Berlin, Germany: Springer-Verlag, 1996, no. 1066, Lecture Notes in Computer Science.

[4] O. Maler, Ed., *Hybrid and Real-Time Systems*. Berlin, Germany: Springer-Verlag, 1997, no. 1201, Lecture Notes in Computer Science.

[5] T. A. Henzinger and S. Sastry, Eds., *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer-Verlag, 1998, no. 1386, Lecture Notes in Computer Science. see also *Proc. 1st Int. Workshop (HSCC'98)*.

[6] P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, Eds., *Hybrid Systems V*. Berlin, Germany: Springer-Verlag, 1999, no. 1567, Lecture Notes in Computer Science.

[7] T. A. Henzinger and S. Sastry, Eds., *Hybrid Systems: Computation and Control*. Berlin, Germany: Springer-Verlag, 1999, no. 1569, Lecture Notes in Computer Science. Proc. 2nd Int. Workshop (HSCC'99).

[8] A. Nerode and W. Kohn, "Models for hybrid systems: Automata, topologies, controllability, observability," in *Hybrid Systems*. Berlin, Germany: Springer-Verlag, 1993, no. 736, Lecture Notes in Computer Science, pp. 317–356.

[9] T. Dang and O. Maler, "Reachability analysis via face lifting," in *HSCC 98: Hybrid Systems—Computation and Control*, T. A. Henzinger and S. Sastry, Eds. Berlin, Germany: Springer-Verlag, 1998, no. 1386, Lecture Notes in Computer Science, pp. 96–109.

[10] M. Kourjanski and P. Varaiya, "Stability of hybrid systems," in *Hybrid Systems III*, R. Alur, T. A. Henzinger, and E. D. Sontag, Eds. Berlin, Germany: Springer-Verlag, 1996, Lecture Notes in Computer Science, pp. 413–423.

[11] M. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 4, pp. 475–482, 1998.

[12] J. Lygeros, C. Tomlin, and S. Sastry, "On controller synthesis for nonlinear hybrid systems," in *Proc. 37th IEEE Conf. Decision and Control*, 1998, pp. 2101–2106.

[13] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "HYTECH: A model checker for hybrid systems," *Software Tools Technol. Transfer*, vol. 1, no. 1, pp. 110–122, 1997.

[14] T. A. Henzinger and P.-H. Ho, "HYTECH: The Cornell hybrid technology tool," in *Hybrid Systems II*, P. Antsaklis, A. Nerode, W. Kohn, and S. Sastry, Eds: Springer-Verlag, 1995, no. 999, Lecture Notes in Computer Science, pp. 265–293.

[15] O. Maler and S. Yovine, "Hardware timing verification using KRONOS," in *Proc. IEEE 7th Israeli Conf. Computer Systems and Software Engineering (ICCBSSE'96)*, Herzliya, Israel, June 1996.

[16] A. Chutinan and B. H. Krogh, "Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations," in *Hybrid Systems: Computation and Control, LNCS*. London, U.K.: Springer-Verlag, 1999.

[17] J. A. Cook and B. K. Powel, "Modeling of an internal combustion engine for control analysis," *IEEE Contr. Syst.*, pp. 20–25, 1988.

[18] E. Hendricks, A. Chevalier, M. Jensen, and T. Vesterholm, "Event based engine control: Practical problems and solutions," SAE, Tech. Rep. 950 008, 1995.

[19] M. Antoniotti, A. Balluchi, L. Benvenuti, A. Ferrari, C. Pinello, A. L. Sangiovanni-Vincentelli, R. Flora, W. Nesci, C. Rossi, G. Serra, and M. Tabaro, "A top-down constraints-driven design methodology for powertrain control system," in *Proc. Global Powertrain Congress (GPC98)* Detroit, MI, Oct. 1998, vol. Emissions, Testing and Controls, pp. 74–84.

[20] J. B. Heywood, *Internal Combustion Engine Fundamental*. New York: McGraw-Hill, 1988.

[21] C. R. Stone, *Introduction to Internal Combustion Engines*. London, U.K.: Macmillan, 1992.

[22] L. Y. Wang, A. Beydoun, J. Cook, J. Sun, and I. Kolmanovsky, "Optimal hybrid control with applications to automotive powertrain systems," in *Control Using Logic-Based Switching*, A. S. Morse, Ed. London, U.K.: Springer-Verlag, 1997, no. 222, Lecture Notes in Control and Information Sciences, pp. 190–200.

[23] K. Butts, I. Kolmanovsky, N. Sivashankar, and J. Sun, "Hybrid systems in automotive control applications," in *Control Using Logic-Based Switching*, A. S. Morse, Ed. London, U.K.: Springer-Verlag, 1997, no. 222, Lecture Notes in Control and Information Sciences, pp. 173–189.

[24] A. Balluchi, M. D. Di Benedetto, C. Pinello, C. Rossi, and A. L. Sangiovanni-Vincentelli, "Hybrid control in automotive applications: The cut-off control," *Automatica*, vol. 35, pp. 519–535, Mar. 1999.

[25] E. Hendricks and T. Vesterholm, "The analysis of mean value SI engine models," SAE, Tech. Rep. 920 682, 1992.

[26] E. Hendricks and S. C. Sorenson, "Mean value modelling of spark ignition engines," SAE, Tech. Rep. 900 616, 1990.

[27] E. Lee and A. Sangiovanni-Vincentelli, "A denotational framework for comparing models of computation," in *Proc. Int. Conf. CAD*, Santa Clara, CA, 1996.

[28] ——, "A framework for comparing models of computation," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 1217–1229, Dec. 1998.

[29] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, Mar. 1999.

[30] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory: A Behavioral Approach*. New York: Springer-Verlag, 1998.

[31] J. Rowson and A. Sangiovanni-Vincentelli, "Interface-based design," in *Proc. Design Automation Conf.*, 1997, pp. 178–183.

[32] Z. Manna and A. Pnueli, "Specification and verification of concurrent programs by ∀-automata," in *Proc. 14th Annu. ACM Symp. Principles of Programming Languages*, 1987.

[33] A. Chow and M. L. Wysynski, "Thermodynamic modelling of complete engine systems—A review," *IMechE J. Auto. Eng.*, vol. 213, no. D, pp. 403–415, 1999.

[34] M. B. Young, "Cyclic dispersion in the homogeneous-charge spark-ignition engine: A literature survey," SAE, Tech. Rep. 810 020, 1981.

[35] N. Ozdor, M. Dulger, and E. Sher, "Cyclic variability in spark ignition engines: A literature survey," SAE, Tech. Rep. 940 987, 1981.

[36] J. K. Ball, R. R. Raine, and C. R. Stone, "Combustion analysis and cycle-by-cycle variations in spark ignition engine comustion, Part 1: An evaluation of comustion analysis routines by reference to model data," *IMechE J. Auto. Eng.*, vol. 212, no. D, pp. 381–399, 1998.

[37] ——, "Combustion analysis and cycle-by-cycle variations in spark ignition engine comustion, Part 2: A new parameter for completeness of combustion and its use in modelling cycle-by-cycle variations in combustion," *IMechE J. Auto. Eng.*, vol. 212, no. D, pp. 507–523, 1998.
[38] F.-Q. Zhao, M.-C. Lai, and D. L. Harrington, "A review of mixture preparation and combustion control strategies for spark-ignited direct-injection gasoline engines," *SAE J. Engines*, vol. 106, no. 970 627, pp. 861–904, 1997.
[39] L. Benvenuti, M. D. Di Benedetto, C. Rossi, and A. Sangiovanni-Vincentelli, "Injector characteristics estimation for spark ignition engines," in *Proc. 37th IEEE Conf. Decision and Control*, Tampa, FL, Dec. 1998, pp. 1546–1551.
[40] A. Balluchi, M. D. Di Benedetto, C. Pinello, and A. L. Sangiovanni-Vincentelli, "A hybrid approach to the fast positive force transient tracking problem in automotive engine control," in *Proc. 37th IEEE Conf. Decision Contr.*, Tampa, FL, Dec. 1998, pp. 3226–3231.
[41] M. L. Franklin and T. E. Murphy, "A study of knock and power loss in the automotive spark ignition engine," SAE, Tech. Rep. 890 161, 1989.
[42] G. König, R. R. Maly, D. Bradley, A. K. C. Lau, and C. G. W. Sheppard, "Role of exothermic centres on knock initiation and knock damage," SAE, Tech. Rep. 890 161, 1989.
[43] J. Hacohen *et al.*, "Experimental and theoretical analysis of flame development and misfire phenomena in a spark-ignition engine," SAE, Tech. Rep. 920 415, 1992.
[44] D.-L. Tang, M.-F. Chang, and M. C. Sultan, "Predective engine spark timing control," SAE, Tech. Rep. 940 973, 1994.
[45] A. Balluchi, M. D. Di Benedetto, C. Pinello, C. Rossi, and A. L. Sangiovanni-Vincentelli, "Cut-off in engine control: A hybrid system approach," in *Proc. 36th IEEE Conf. Decision Contr.*, San Diego, CA, Dec. 1997, pp. 4720–4725.
[46] D. Hrovat, D. Colvin, and B. K. Powell, "Comments on 'Applications of some new tools to robust stability analysis of spark ignition engine: A case study'," *IEEE Trans. Contr. Syst. Technol.*, vol. 6, pp. 435–436, May 1998.
[47] D. Shim, J. Park, P. P. Khargonekar, and W. B. Ribbens, "Reducing automotive engine speed fluctuation at idle," *IEEE Trans. Contr. Syst. Technol.*, vol. 4, pp. 404–410, July 1996.
[48] C. Y. Mo, A. J. Beaumont, and N. N. Powell, "Active control of driveability," SAE, Tech. Rep. 960 046, 1996.
[49] M. Pettersson, L. Nielsen, and L. G. Hedström, "Transmission-torque control for gear shifting with engine control," SAE, Tech. Rep. 970 864, 1997.
[50] A. Hoess, W. Hops, R. Doerfler, and H. Rauner, "Longitudinal autonomous vehicle control utilizing access to electronic throttle control, automatic transmission and brakes," SAE, Tech. Rep. 961 009, 1996.
[51] S. Richard, P. Chevrel, P. de Larminat, and B. Marguerie, "Polynomial pole placement revisited: Application to active control of a car longitudinal oscillations," in *Proc. 6th IEEE Eur. Contr. Conf.*, Karlsruhe, Germany, 1999.
[52] S. Hang and W. Ren, "Safety, comfort, and optimal tracking control in AHS applications," *CSM*, vol. 18, no. 4, pp. 50–64, Aug. 1998.
[53] R. Schernewski, K. Dadhe, and U. Kiencke, "Different approaches to integrated drive unit control," in *Proc. 6th IEEE Eur. Contr. Conf.*, Karlsruhe, Germany, 1999.
[54] H. List and P. Schoeggl, "Objective evaluation of vehicle driveability," SAE, Tech. Rep. 980 204, 1998.
[55] L. S. Pontryagin, V. G. Boltyansky, R. V. Gamkrelidze, and E. F. Mischenko, *The Mathematical Theory of Optimal Processes*. New York: Wiley, 1962.
[56] V. I. Utkin, "Variable structure systems with sliding modes: A survey," *IEEE Trans. Automat. Contr.*, vol. 22, pp. 212–222, 1977.
[57] S. B. Choi and J. K. Hedrick, "An observer-based controller design method for improving air/fuel characteristics of spark ignition engines," *IEEE Trans. Contr. Syst. Technol.*, vol. 6, pp. 325–334, May 1998.
[58] A. Balluchi, M. Di Benedetto, C. Pinello, C. Rossi, and A. Sangiovanni-Vincentelli, "Hybrid control for automotive engine management: The cut-off case," in *Hybrid Systems: Computation and Control*. London, U.K.: Springer-Verlag, 1998, vol. 1386, LNCS.
[59] D. Hrovat and J. Sun, "Models and control methodologies for IC engine idle speed control design," *Contr. Eng. Practice*, vol. 5, no. 8, Aug. 1997.
[60] K. R. Butts, N. Sivashankar, and J. Sun, "Application of ℓ_1 optimal control to the engine idle speed control problem," *IEEE Trans. Contr. Syst. Technol.*, vol. 7, no. 2, pp. 258–270, Mar. 1999.
[61] S. Yurkovich and M. Simpson, "Crank-angle domain modeling and control for idle speed," *SAE J. Engines*, vol. 106, no. 970 027, pp. 34–41, 1997.
[62] J. L. Chen and G. Chen, "Throttle body at engine idle-tolerance effect on flow rate," SAE, Tech. Rep. 951 057, 1995.
[63] C. H. Onder and H. P. Geering, "Model-based multivariable speed and air-to-fuel ratio control of a SI engine," SAE, Tech. Rep. 930 859, 1993.
[64] C. Carnevale and A. Moschetti, "Idle speed control with H_∞ technique," SAE, Tech. Rep. 930 770, 1993.
[65] D. Hrovat and B. Bodenheimer, "Robust automotive idle speed control design based on μ-synthesis," in *Proc. IEEE Amer. Contr. Conf.*, San Francisco, CA, 1993, pp. 1778–1783.
[66] L. Kjergaard, S. Nielsen, T. Vesterholm, and E. Hendricks, "Advanced nonlinear engine idle speed control systems," SAE, Tech. Rep. 940 974, 1994.
[67] M. Abate and V. Di Nunzio, "Idle speed control using optimal regulation," SAE, Tech. Rep. 905 008, 1990.
[68] C. Tomlin, J. Lygeros, and S. Sastry, "Synthesizing controllers for nonlinear hybrid systems," in *1st Int. Workshop, HSCC'98, Hybrid Systems: Computation and Control*, T. Henzinger and S. Sastry, Eds., 1998, no. 1386, Lecture Notes in Computer Science, pp. 360–373.

Andrea Balluchi received the Laurea degree (*summa cum laude*) in electrical engineering and the Ph.D. degree in automation and industrial robotics from the University of Pisa, Italy, in 1993 and 1997, respectively.

He was a Visiting Scholar at the Department of Computer Science, University of Sheffield, in 1992; at the Laboratoire d'Analyse et d'Architecture des Systemes CNRS, Toulouse, in 1995; and at the Department of Electrical Engineering and Computer Science, University of California, Berkeley in 1996. In 1993, he was a junior Researcher at the Department of Electrical Systems and Automation, University of Pisa. In 1997, he joined PARADES, Rome, Italy, a research laboratory in Rome supported by Cadence Design Systems, Magneti Marelli, and ST-Microelectronics. His research interests are in the areas of hybrid systems, automotive engine control, man–machine interface for man-in-the-loop systems, variable structure, and decentralized control.

Luca Benvenuti was born in Rome, Italy, on February 8, 1966. He received the Laurea degree (*summa cum laude*) in electrical engineering and the Ph.D. degree in systems engineering from the University of Rome La Sapienza, Rome, in 1992 and 1995, respectively.

He was a Visiting Graduate Student at the University of California, Berkeley, in 1995. In 1997, he was a Scientific Consultant for Magneti Marelli. From 1997 to 1999, he had a postdoctoral position at the Department of Electrical Engineering, University of L'Aquila, L'Aquila, Italy. He is currently a Scientific Consultant at PARADES, Rome, Italy, a research laboratory supported by Cadence Design Systems, Magneti–Marelli, and ST-Microelectronics. His research interests are in the areas of control of nonlinear systems, analysis and control of hybrid systems, and positive linear systems.

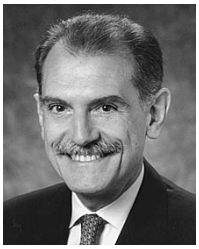
Maria Domenica Di Benedetto (Senior Member, IEEE) received the Dr.Ing. degree (*summa cum laude*) in electrical engineering and computer science from the University of Roma, La Sapienza, in 1976. She received the D.-Ing. degree and the Doctorat d'Etat Sciences degree from the Universite' de Paris-Sud, Orsay, France, in 1981 and 1987, respectively.

From 1979 to 1983, she was Research Engineer at the scientific centers of IBM in Paris and Rome. From 1983 to 1987, she was "Ricercatore" at the University of Roma, La Sapienza. From 1987 to 1990, she was an Associate Professor at the Istituto Universitario Navale of Naples. From 1990 to 1993, she was an Associate Professor at the University of Roma, La Sapienza. Since 1994, she has been Professor of control theory at the University of L'Aquila. Since 1995, she has been Adjunct Professor at the Department of Electrical Engineering and Computer Science, University of California, Berkeley (UCB). In 1987, she was Visiting Scientist at the Massachusetts Institute of Technology, Cambridge; in 1988, 1989, and 1992 Visiting Professor at the University of Michigan, Ann Arbor; in 1992, Chercheur Associe', C.N.R.S., Poste Rouge, Ecole Nationale Superieure de Mecanique, Nantes, France; in 1990, 1992, 1994, and in 1995, McKay Professor at UCB. Her research interests revolve around nonlinear control and hybrid systems. In March 2000, she organized and chaired the Trilateral Conference Universita' di L'Aquila, Universita' di Roma, La Sapienza, and UCB.

Prof. Di Benedetto received the IBM Italy Prize for outstanding contributions to the "Progetto Voce" in 1982 and the Mose' Ascoli Award for the best graduate in electrical engineering and computer science, Università di Roma, La Sapienza, Rome, in 1976. She was Associate Editor of the IEEE TRANSACTIONS OF AUTOMATIC CONTROL and is Subject Editor of the *International Journal of Robust and Nonlinear Control*.

Claudio Pinello received the Laurea degree (*summa cum laude*) in electrical engineering from the Università di Roma, La Sapienza, Rome, in June 1997, and is pursuing the Ph.D. degree from the Department of Electrical Engineering and Computer Science, University of California, Berkeley.

He worked at PARADES research laboratory, Rome, from 1997 to 1998, and the BMW Technology Office, Palo Alto, CA, during summer 1999. His interests are in control theory and applications, with particular emphasis on hybrid systems and automotive engine control, embedded systems design, and fault-tolerant distributed systems.



Alberto Luigi Sangiovanni-Vincentelli (Fellow, IEEE) received the Dottore in Ingegneria degree (*summa cum laude*) in electrical engineering and computer science from the Politecnico di Milano, Milan, Italy, in 1971.

He holds the Edgar L. and Harold H. Buttner Chair of Electrical Engineering and Computer Sciences, University of California, Berkeley, where he has been on the Faculty since 1976. From 1980 to 1981, he was a Visiting Scientist at the Mathematical Sciences Department, IBM T.J. Watson Research Center. In 1987, he was Visiting Professor at the Massachusetts Institute of Technology, Cambridge. He has held a number of Visiting Professor positions at the universities of Torino, Bologna, Pavia, Pisa, and Rome. He was a Co-founder of Cadence and Synopsys, the two leading companies in the area of electronic design automation. He was a Director of ViewLogic and Pie Design System and Chair of the Technical Advisory Board of Synopsys. He is Chief Technology Advisor of Cadence Design System. He is a Member of the Board of Directors of Cadence, where he is Chairman of the Nominating Committee, Sonics Inc., and Accent, an ST-Cadence joint venture. He is Founder of the Cadence Berkeley Laboratories and of the Cadence European laboratories. He was Founder of the Kawasaki Berkeley Concept Research Center, where he holds the title of Chairman of the Board. He has consulted for a number of U.S. companies including IBM, Intel, ATT, GTE, GE, Harris, Nynex, Teknekron, DEC, HP; Japanese companies including Kawasaki Steel, Fujitsu, Sony, and Hitachi; and European companies including SGS-Thomson Microelectronics, Alcatel, Daimler–Benz, Magneti–Marelli, BMW, and Bull. He is Scientific Director of PARADES, Rome, a European Group of Economic Interest supported by Cadence, Magneti–Marelli and ST Microelectronics. He is on the Advisory Board of the Lester Center of the Haas School of Business and of the Center for Western European Studies, and is a member of the Berkeley Roundtable of the International Economy (BRIE). He is author of more than 480 papers and ten books in the area of design methodologies, large-scale systems, embedded controllers, hybrid systems, and tools.

In 1981, Dr. Sangiovanni-Vincentelli received the Distinguished Teaching Award of the University of California. He received the worldwide 1995 Graduate Teaching Award of the IEEE, the Guillemin–Cauer Award (1982–1983), and the Darlington Award (1987–1988). In 1999, he was awarded the CASS Golden Jubilee Medals to a set of the IEEE Circuits and Systems Society members in order to express its appreciation for their exceptional contributions toward advancing in various forms the society's goals during the first 50 years of its existence. He is a Member of the National Academy of Engineering. He was Technical Program Chairperson of the International Conference on CAD and General Chair. He was Executive Vice-President of the IEEE Circuits and Systems Society.