Automotive systems offer a rich opportunity for hybrid models, controls, and tools. Beyond the traditional use of hybrid models for representing the behavior of the composition of discrete controller and continuous plants, automotive mechanical systems exhibit hybrid behavior as demonstrated in this chapter. In addition, hybrid systems can be used to capture system specifications at the highest level of abstraction and to model implementation architectures thus enabling a rich design space exploration.

Chapter contents

15.1 Introduction page 440
15.2 Design methodologies for embedded automotive control systems 441
15.2.1 System integration 441
15.2.2 Design scenario and design flow 442
15.2.3 Control system design 445
15.3 Hybrid model-predictive control in automotive applications 452
15.3.1 Motivation 452
15.3.2 Survey of MPC applications in vehicle dynamics control 453
15.3.3 Survey of MPC applications in powertrain control 454
15.3.4 Example: Hybrid MPC of magnetic actuators 455
15.4 Control of a homogeneous charge compression ignition engine 461
15.4.1 HCCI engine concept 461
15.4.2 Control of ignition timing 463
15.4.3 Properties and model of the HCCI engine 464
15.1 Introduction

This chapter presents an application of hybrid systems that is of significant industrial interest: power-train modeling and control for automobiles.

Engine control is a challenging problem that involves many functional and non-functional requirements. The problem is to develop control algorithms and their implementation with guaranteed properties that can substantially reduce emissions and gas consumption with increased performance.

The introduction of hybrid system modeling and control was motivated by the need for verifying closed-loop systems where the plant to be controlled are continuous-time systems and the controller is a digital system. However, hybrid models are general enough to be useful in other areas of design. In particular, engine control offers a rich set of application of hybrid systems:

- The power-train itself can be represented as a hybrid system. In fact, an accurate model of a four-stroke gasoline engine has a “natural” hybrid representation:
  - Each cylinder in the engine has four discrete modes of operation corresponding to the stroke it is in (hence, its behavior is well represented by a finite state machine (FSM)).
  - The power-train delivering the torque produced by the engine can be modeled as a continuous-time process.

The continuous motion of the power-train depends on the torque generated by the engine and at the same time, it determines the timing of transitions between the modes of operation of each cylinder so that the behavior of the plant cannot be decomposed into two independent parts. In traditional design approaches, the modes of operation of the cylinders are smoothed out by using an average continuous model thus converting a mixed discrete-continuous time model into a purely continuous one. This approximation results in control actions that may perform poorly during the transients caused by the mode changes of the cylinder. Hybrid models of course do not suffer from these effects but they do require more complex control algorithm.

- Hybrid models can be extended to address problems at the boundary between control algorithm design and hw/sw implementation, since they allow the designer to capture the effects of limited resources and physical constraints on the performance of the controlled system and check the correctness of the design.

- Hybrid formalisms can be applied to represent system specifications and deploy them in an architecture of control algorithms and requirements.

This chapter is intended to increase the awareness of the automotive industry to the currently available methods, tools, and successful applications in the field.

The chapter is divided into three sections: in the first section we present a hybrid model-based design flow for embedded controllers in automotive control.

In the second section we present the use of hybrid model-predictive control (MPC) in automotive applications. MPC is an optimization approach that takes into account a cost function and constraints. Hybrid MPC provides the further capability
of employing hybrid prediction models. The application of hybrid MPC to vehicle dynamics control and to power-train control is presented.

In the third section, the behavior of the homogeneous charge compression ignition engine is described. This engine combines features of the traditional spark ignited Otto-cycle and compression ignited Diesel-cycle engines. This engine is indeed a new concept and it has great potential for substantial reduction of energy consumption and pollution. In this section, we make the case for hybrid modeling of these engines suitable for control design.

### 15.2 Design methodologies for embedded automotive control systems

#### 15.2.1 System integration

Automobile functions such as engine control, gear-box control, anti-lock braking system (ABS), dashboard controller, and vehicle dynamical control (VDC) are today enabled by a distributed embedded system.

This system consists of a number of ECUs dedicated to each function yielding the so-called federated architecture. The trend is towards an integrated modular approach where the one-to-one correspondence between a function and its implementation does not hold any longer: a function may be supported by more than one ECUs and/or one ECU can support more than one function. This evolution offers clear advantages in terms of cost, extensibility and scalability but does present serious challenges.

Tier-1 suppliers responsible for the subsystems mentioned above, in general, give scant information about the details of their products causing problems for an overall understanding of the interplay of subsystems. The difficulties encountered in integrating complex parts made system integration a nightmare in the automotive industry. To mitigate the integration problems, the AUTOSAR initiative [308], promoted by leading European original equipment manufacturers (OEMs) and Tier-1 suppliers, was formed to establish an open standard for automotive electric/electronic architectures that hopefully allows plug-and-play of embedded controllers. Even in the presence of solidly established standards, the design process for embedded controllers has to be significantly improved to support the movement towards an integrated modular architecture.

Successful approaches to the design of control algorithms using hybrid system methodologies had been presented in the literature, e.g. cut-off control [33], intake throttle valve control [40], actual engaged gear identification [38], and adaptive cruise control [459].

In this section, we analyze a design flow for embedded controllers in the automotive industry with the purpose of identifying challenges and additional opportunities for hybrid system technology. In particular, in Subsection 15.2.2, an overview of the typical design flow for embedded controllers adopted by the automotive industry is presented. The flow is divided into two parts: synthesis and verification/testing. We
focus on the synthesis flow as it does have the highest potential of improving substantially design productivity. In Subsection 15.2.3, we present relevant problems that hybrid system technologies contributed to solve.

### 15.2.2 Design scenario and design flow

The ECUs interact by asynchronous communication over a communication network specifically designed for automotive applications, such as CAN ([340, 561]). Typically, an ECU implements a multirate control system composed of nested control loops, with frequency and phase drifts between fixed sampling-time actions and event-driven actions.

A typical ECU (e.g., the one demanded for engine control) may have more than one hundred I/O signals, implement up to three hundred control algorithms, and share approximately one hundred signals with the other related ECUs. The complexity of the design of automotive ECUs is further increased by very critical constraints on reliability, cost and time-to-market, and constraints on power consumption, weight, and position. As a consequence, a successful design, in which costly and time consuming re-design cycles are avoided, can only be achieved using efficient design methodologies that allow for component reuse and for evaluation of platform requirements at the early stages of the design flow [297].

**Design flow** The standard design flow for automotive ECUs adopted by OEMs and Tier-1 companies (subsystem suppliers) consists of two main parts: the *synthesis flow* and the *integration and testing flow*. In particular, the synthesis flow is articulated in the following steps:

1. **System specification**: Formalization of system level customer requirements; completion of under-specified requirements; abstraction at the system level of customer requirements regarding lower layers (e.g. either a control algorithm or a piece of software to be integrated in the design).

2. **Functional deployment**: Decomposition of the system into a collection of interacting subsystems. The specifications for each subsystem are derived from the overall specifications. For each subsystem, the architecture of control algorithms and their specifications are also defined.

3. **Control system**: Synthesis of each control algorithm, according to the specification defined in the previous step, and its validation.

4. **HW/SW components**: Specifications for the implementation of the control algorithms. Design of the hardware and software architectures.

The synthesis flow terminates with the development of the hardware, the software and possibly some electromechanical components.
The purpose of the integration and testing flow is the complete testing of the realization of the ECU and the verification of the compliance with the requirements. The steps taken in the integration and testing flow are as follows:

1. **HW/SW testing:** The correct realization of the hardware and software architecture is verified. This step includes testing of real-time implementation requirements, electrical power drivers, and communication.
2. **Control validation:** The correct implementation of each control algorithm with respect to the given functional description is assessed by testing either its input/output response or its closed-loop behavior.
3. **Functional integration:** The correct interaction of the implemented control algorithms is tested considering an increasing number of algorithms together, to verify that their composition exhibit the behavior defined during functional deployment.
4. **System testing:** The entire ECU is tested against system specification and the compliance with customer requirements is verified.

**Platform-based design** The platform-based design methodology proposed in [565] is a convenient framework to cast the design problems and the flow presented here. In addition, it provides concepts and techniques to maximize reuse at each design step and early verification with abstracted information from possible implementation platforms [358, 566].

A platform is a layer of abstraction that hides the unnecessary details of the underlying implementation and yet carries enough information about the layers below to prevent design iterations. The choice of the layers of abstraction and of the corresponding parameters are essential in the quality of the final solution of the design problem.

The basic tenets of the platform-based design methodology are:

- regarding design as a “meeting-in-the-middle process’” where successive refinements of specifications meet with abstractions of potential implementation;
- the identification of precisely defined layers where the refinement and abstraction process take place.

The layers then support designs built upon them by isolating from lower-level details but letting enough information transpire about lower levels of abstraction to allow design space exploration with a fairly accurate prediction of the properties of the final implementation. The information should be incorporated in appropriate parameters that annotate design choices at the present layer of abstraction. These layers of abstraction are called *platforms.*

In [21], the application of the platform-based design methodology to the design of powertrain control systems was described.
Derivative design  Since, in the automotive industry, embedded control system design is highly dominated by the need to implement an efficient reuse to meet increasing constraints on cost and time-to-market, a derivative design approach is commonly adopted [440, 441]. According to this approach, every two or three years a new generation of products is conceived. The design of a new generation is intended to accommodate the specifications of customers for the near future, so that for each new customer engagement, the control algorithms as well as the electrical and mechanical components are obtained by (hopefully minor) modifications of the current product generation. When entering the design phase of a new product generation, the architecture of control algorithms as well as of their implementation should be conceived to maximize future re-use, for instance by choosing the correct granularity of partitioning. The resulting ECUs are then variants of a same originating design and ideally share the highest number of parts (e.g. algorithms, software modules, hardware parts, mechanical components).

Model-based design  Another standard approach in automotive industry is the so-called model-based design. In this approach, specifications, functional architectures, algorithms, and implementation architectures are represented formally by models, thus allowing, at least in principle, formal analysis and automatic synthesis. Using block diagram-based modeling tools, control algorithms are designed and initial validation in off-line simulation is performed. Models of control algorithms are the basis for all subsequent development stages.

The advantages of model-based design are obvious:

- sharing models reduces the risk of mistakes and shortens the development cycles;
- design choices can be explored and evaluated much faster and more reliably;
- the result of a model-based development process is an optimized and fully tested system.

While the overall approach is quite powerful, today there is only an incomplete implementation of it in the development cycle. In fact, model-based design is widely used for the formal representation of control algorithms, using tools such as Simulink/Stateflow or ASCET by ETAS, but it is partially and superficially applied to control algorithm validation. The lack of an extensive model-based validation of the control algorithms results in major efforts in experimental validation, which is very expensive, time-consuming, and achieves only a bounded coverage of the system behavior. Due to the high cost of experimental validation, the OEMs will provide less support to Tier-1 companies for it in the future. In the rest of this section, the part of the synthesis flow regarding control design will be analyzed in detail, showing the design steps for which hybrid system techniques contributed or may significantly contribute to provide a more efficient approach.
15.2.3 Control system design

At the control system level, the algorithms to be implemented in the architecture defined at the functional level are designed. The design process for each control algorithm involves:

1. plant modeling;
2. controller synthesis;
3. fast prototyping.

Plant modeling Traditionally, control engineers adopt mean-value models to represent the behavior of automotive subsystems, which abstract from the hybrid nature. However, the need for hybrid system formalisms to model the behavior of subsystems in automotive applications is apparent in many cases. To illustrate the relevance of hybrid modeling in automotive applications, we briefly describe a hybrid model for a spark ignition engine and an automotive driveline.

Example 15.1 Hybrid model of a four-stroke spark ignition engine.

An accurate model of a four-stroke spark ignition engine has a natural hybrid representation because the cylinders have four modes of operation corresponding to the stroke they are in, while driveline and air dynamics are continuous-time processes. In addition, these processes interact tightly. In fact, the timing of the transitions between two phases of the cylinders is determined by the continuous motion of the driveline, which in turn depends on the torque produced by each piston.

In more detail, consider the hybrid model of the torque generation process and the driveline presented in [34]. For a given gear selection and clutch position, the driveline is described by a continuous time system whose state includes the driveline torsion angle, the crankshaft revolution speed, and the wheel revolution speed. The inputs of the model are the torque $T$ produced by the engine and the load wheel torque $T_w$.

The engine torque $T$ is given by $\sum_{i=1}^{N} T_i$, where $T_i$ is the torque generated by each piston at each cycle. The profile of $T_i$ is determined by the phases of the cylinder, the piston position, the mass of air and the mass of fuel loaded in the cylinder during the intake phase, and on the spark ignition timing.

The four-stroke engine cycle can be modeled by means of a finite-state machine (FSM) capturing the sequential nature of the behavior of the cylinders. In fact, each cylinder cycles through the following four phases:

- **Intake ($I$)**: The piston goes down from the top dead center (TDC) to the bottom dead center (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.
However, for spark ignition engines, the torque generated by each piston is related not only to the phase of the cylinder and the air and fuel charge, but also to the spark generation process. Intuitively, spark ignition should occur exactly when the piston reaches the TDC of the compression stroke. Since the combustion process takes nonzero time to complete, then the pressure in the cylinder reaches its maximum some time after spark ignition. As a consequence, in order to achieve maximum fuel efficiency, it is convenient to produce the spark before the piston completes the compression stroke (positive spark advance). On the other hand, 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 (and much faster than using only the throttle valve) the value of the torque generated during the expansion run. 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. 15.1. The FSM state 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 FSM changes its state either when a spark is given or when a dead center (DC) is reached. The latter event depends on the continuous motion of the driveline and more precisely on the crankshaft angle, which defines the position of the piston. In turn, the crankshaft revolution speed depends on the torque $T$ produced by the engine.

Fig. 15.1 Finite-state machine describing the behavior of the $i$-th cylinder.
Finally, the torque produced by the cylinder depends on the air–fuel mixture loaded during the intake stroke. Since the air–fuel mixture is loaded in the cylinder during the intake stroke while the torque generation starts after the spark is ignited, then there is a delay between the time at which the mixture is loaded and the time at which the corresponding active torque is generated. This delay can be modeled by means of another discrete-event model that is synchronized with the FSM transitions. The overall model of the torque generation process for a single cylinder consists then of four communicating submodels:

- an FSM, modeling the four-stroke engine cycle and the spark generation process;
- a discrete model describing the discrete delay on the active torque generation; and
- two continuous-time systems, modeling the air intake process and the profile of the generated torque.

Example 15.2 Driveline

A second very interesting automotive subsystem rich of discrete-continuous interactions is the driveline [37]. An accurate model of the driveline has a natural hybrid representation because of the discontinuities due to clutch and the gear on the continuous motion of the driveline.

The clutch can be modeled as a hybrid system with three discrete states: Locked, Slipping, and Unlocked. In Fig. 15.2 an FSM representing the clutch is depicted. When the clutch is Locked the clutch plate and the flywheel are rigidly connected by static friction, so that their inertias are collected in a single first-order dynamics.

The highest coupling torque \( T_{\text{max}} \) before incurring in clutch slipping corresponds to the maximum static friction torque, which is a function of the pressure \( P_c \) between the clutch plate and the flywheel, i.e. \( T_{\text{max}} = \mu_s P_c \). When the transmitted torque \( T \) exceeds \( T_{\text{max}} \), the system enters the state Slipping, where the clutch plate and the flywheel are no longer strictly connected but they slip one on the other. In this case the coupling torque is due to dynamical friction and it is a function of the sliding speed. If the transmitted torque decreases, then the clutch returns in the state Locked, while if \( P_c = 0 \) the clutch enters the state Unlocked. In the state Unlocked the crankshaft is completely decoupled from the rest of the driveline and the two systems follow independent dynamics.

For some applications, e.g. actual engaged gear identification [38], it is useful to collect in a single state the clutch Unlocked and Slipping states. In such cases, the overall system can be described by a hybrid system with seven discrete states and four continuous-state variables.
The FSM describing the discrete dynamics of the model is depicted in Fig. 15.3. The discrete state $q_i$, for $i = 1, \ldots, 5$, correspond to $i$-th gear engaged and clutch locked.

The location $q_{RG}$ models reverse gear engaged and the location $q_N$ corresponds to the case of open driveline (idle gear and/or clutch open) or slipping clutch. The continuous state variables are the driveline torsion angle $\alpha$, the crankshaft revolution speed $\omega_c$, the clutch plate revolution speed $\omega_e$, and the wheel revolution speed $\omega_w$. When the clutch is locked, then $\omega_c = \omega_e$, so that the continuous behavior of the driveline can be described by a third-order linear system with parameters depending on the selected gear. The hybrid model has as inputs the position of the gear lever

$$\text{lever} \in \{1, 2, 3, 4, 5, RG, N\}$$

and the torque generated by the engine while the connection pressure of the clutch plates $P_c$ and the load wheel torque $T_w$ are considered as disturbances. A more detailed driveline hybrid model, with 6048 discrete state combinations and 12 continuous state variables, is presented in [37]. In addition to the clutch and the gear, the proposed hybrid model describes the discontinuities in the driveline due to engine suspension, elastic torsional characteristic, tires, frictions and backlashes. This very detailed driveline hybrid model exhibits a behavior very close to the physical driveline and has been developed to be used for control algorithm validation.

![Fig. 15.3 Hybrid model of the driveline.](image)

**Controller synthesis** Control algorithms are often characterized by many operation modes, that are conceived to cover the entire life-time of the product: starting from in-factory operations before car installation, configuration, first power-on, power-on, functioning, power-off, connection to diagnostic tools, and so on. During normal functioning, control strategies can be either in one of the nominal operation modes or in some recovery mode. A significant number of algorithms are dedicated to the computation of switching conditions between modes and controller initializations.

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

Diagnostic algorithms represent a large part of the strategies implemented in automotive ECUs. For engine control, the implementation of diagnosis algorithms is enforced by legislation: OBDII (on-board diagnosis II) in USA and EOBD (European On Board Diagnosis) in the EU. In general, these requirements specify that every fault, malfunction or simple component degradation that leads to pollutant emissions over given thresholds should be diagnosed and signaled to the driver. This requirement has a significant impact on ECU design, since it implies the development of many on-line diagnostic algorithms [168].

Both specifications and accurate models of the plant are often hybrid in automotive applications but the methodology currently adopted for algorithm development is rather crude and can be summarized as follows: the continuous functionalities to be implemented in the controller are designed based on mean-value models of the plant, with some ad hoc solutions to manage hybrid system issues (such as synchronization with event-based behaviors); if the resulting behavior is not satisfactory under some specific conditions, then the controller is modified to detect critical behaviors and operate consequently (introducing further control switching). Nevertheless hybrid system techniques can significantly contribute to the improvement of control algorithm design as illustrated by the two following algorithms regarding cut-off control and identification of the actual engaged gear.

**Cut-off control**  A quite critical driveability objective is the control of longitudinal oscillations of the car when fast engine torque variations are requested by the driver (tip-in and tip-out). Roughly speaking, the control consists of active damping of powertrain oscillations.

The problem is particularly challenging when the engine is not equipped with electronic throttle valve, since in this case only fuel injection and spark ignition controls can be used for engine torque modulation to achieve the desired damping of the oscillations. Most of the proposed approaches are based on mean-value continuous-time models of the torque generation. As a consequence, since the torque generation process has a discrete behavior, the implementation of such control strategies on a real engine may result in very poor closed-loop performance and may give rise to unpredictable unpleasant behaviors. On the contrary, a design based on a hybrid model of the engine allows to develop control laws for which closed-loop performances are guaranteed.

A hybrid approach to the design of a longitudinal oscillation damping control during tip-out was presented in [33]. The control problem arises when the driver, by releasing the fuel pedal, requests no torque to the engine. In this case, an obvious strategy to minimize fuel consumption and emissions is to shut fuel injection, an operation called cut-off. However, cutting off fuel injection as soon as the gas pedal is
released, causes a sudden torque reduction that may result in unpleasant oscillations compromising driving comfort. A more complex control action involves modulating the engine torque from the present value to the value corresponding to cut-off in an attempt to prevent oscillations. This control policy is implemented by slowing down air flow decay and, when air quantity is below a threshold, reducing fuel injection gradually to zero.

As it is often the case, heuristic rule-based controls need extensive tuning, yield satisfactory solutions only in a limited range of operations, and are hardly optimal with respect to the emissions and fuel consumption. In particular, if air reduction is too slow, when the driver releases the gas pedal and presses the clutch pedal to change gear, engine speed raises for a while, thus causing a definite reduction in passengers’ comfort. Moreover, if air and/or fuel reduction is too fast, oscillations take place anyway.

The hybrid control algorithm presented in [33] is able to steer the evolution of the system to the fuel cut-off condition, minimizing the amplitude of the undesired oscillations. Since a hybrid model of the engine has been considered during the design, the algorithm acts on fuel injection and spark ignition once per engine cycle.

---

Fig. 15.4 Evolutions of the oscillating modes (a) and engine speed, accelerations, and injection signal profiles (b), in an uncontrolled cut-off (top) and with the proposed hybrid control strategy (bottom).
for each cylinder taking into account synchronization and actuators’ delay. The hybrid approach adopted for synthesis of the cut-off control algorithm guarantees the correctness of the behavior when applied to the real plant.

The proposed cut-off control strategy was tested at Magneti-Marelli Engine Control Division on a commercial car, a 16-valve 1400 cc engine car. The experiment was carried out driving the car in the test ring and measuring the important parameters and variables that determine the performance of the control strategy. In Fig. 15.4 the performance achieved by the proposed hybrid cut-off strategy is compared with an instantaneous uncontrolled cut-off operation. On the left the evolution of the oscillating modes \( \hat{x} \) are reported. The effectiveness of the proposed controller is apparent from the evolution of the vehicle acceleration and engine speed reported on the right.

**Actual engaged gear identification**  As a second example, consider the problem of on-line identification of the actual engaged gear. Engine control strategies achieving high performance and efficient emissions control depend critically on such identification algorithm. In fact, the knowledge of the actual engaged gear is necessary in engine torque control to compensate the equivalent inertia of the vehicle on the crankshaft and, for diesel engines, it is very important to improve emissions control.

In cars equipped with manual gear shift, this information is not directly available and, at present, it is deduced by comparing the revolution speed of the wheels with the revolution speed of the crankshaft. However, since both of them are affected by oscillations due to the elasticity of the driveline and the tires, then this approach implies large time delays in the identification and may produce significant identification errors.

The identification algorithm presented in \([38]\) is based on a hybrid model of the driveline (see the driveline model previously presented) where the engaged gear and connection clutch state are represented as discrete states. The design problem is then formulated as the identification of the discrete state of the driveline hybrid model and an on-line identification algorithm is obtained by applying the methodology for hybrid observer design proposed in \([35]\). The algorithm is able to detect a change in the continuous time dynamics and identify the discrete state of the driveline hybrid model by processing the crankshaft and wheel revolution speed measurements and an estimate of the engine torque mean-value.

On-line identification is achieved by the generation of appropriate residual signals, one for each gear, which vanish when the corresponding gear is engaged and the clutch is locked.

**Figure 15.5** shows the results on actual engaged gear identification obtained in Magneti Marelli Powertrain using an Opel Astra equipped with a diesel engine and a robotized gearbox SeleSpeed. For the validation of the identification algorithm, the estimated engaged gear is compared to the signal on actual engaged gear provided by the control unit of the robotized gearbox. The algorithm was tested on several maneuvers for a total of 250 gear engagements and it was able to identify correctly the actual engaged gear within 250 msec, as requested by specification. It proved to be remarkably robust with respect to parameter uncertainties (e.g. vehicle inertia) and time-varying unknown disturbances (e.g. wheel torque and road slope).
Fig. 15.5 Gear identification with experimental data. The maneuver starts with car at rest, clutch open and first gear engaged \((q = q_N = 0)\). After a clutch slipping phase \((q = q_N = 0)\), the clutch is locked \((q = q_1 = 1)\); later, second gear \((q = q_2 = 2)\) and then third gear \((q = q_3 = 3)\) are engaged, passing through idle and slipping \((q = q_N = 0)\).

### 15.3 Hybrid model-predictive control in automotive applications

#### 15.3.1 Motivation

Automotive control problems are characterized by fast dynamics, constraints on manipulated and control variables, performance criteria, and the need for low-complexity control software. Model-predictive control (MPC) accounts for a cost function to be optimized, for constraints on system inputs and outputs. Furthermore, explicit MPC synthesis strategies convert the controller into an equivalent piecewise affine feedback law, which can be quickly evaluated by a simple control code, hence allowing the use of MPC for fast systems, and the implementation in simple electronic control units.

Hybrid model-predictive control provides the further capability of employing hybrid prediction models in MPC. Hybrid behaviors appear in automotive systems in the form of impulsive effects such as impacts [72], different operating modes [106], and discrete elements such as quantized inputs [268]. Furthermore, hybrid models with mode-dependent linear continuous dynamics provide a better approximation of nonlinear dynamics compared to linear models [137]. In the automotive domain,
hybrid model-predictive control has been successfully applied to vehicle dynamics control and to powertrain control.

15.3.2 Survey of MPC applications in vehicle dynamics control

The main contributions involve traction control, adaptive cruise control, and semi-active suspensions. In [106] the problem of traction control has been considered. Two modes for traction torque on the tire $\tau_t$ have been identified from experimental data:

$$\tau_t(\Delta \omega, \mu) = k_{i1} \Delta \omega + k_{i2} \mu + k_{i3}, \quad \text{if } h_i \Delta \omega + k_i \mu \leq \ell_i, \quad i = 1, 2,$$

where $\Delta \omega$ is the wheel slip (the difference between normalized engine and vehicle speed) and $\mu$ is the road adhesion coefficient. As a result, the vehicle dynamics is formulated as a hybrid system with two modes, that is used for prediction in a hybrid MPC controller. Experiments on a test vehicle were performed by executing an explicit MPC controller consisting of 504 regions on a 266-MHz Pentium laptop connected to the car, showing very good performance and robustness.

In the cruise control problem, the vehicle velocity is regulated to track a user-defined reference. In adaptive cruise control, the controller accounts also for the interaction with other on-road vehicles, ensuring a safety distance. In [459] the adaptive cruise control is formulated as a multi-objective problem, in which one objective is to track the reference velocity, the other is to maintain a safety distance. The second objective becomes active only when another vehicle is close enough,

$$J_d(d(k)) = \begin{cases} 0 & \text{if } d(k) \geq d_{\text{safe}}, \\ w_d(d(k) - d_{\text{safe}})^2 & \text{otherwise}, \end{cases}$$

where $J_d$ is the cost related to the distance $d(t)$ to the nearest vehicle, $d_{\text{safe}}$ is the safety distance above where the distance cost is null and $w_d$ is a positive weight. The hybrid character of this control problem appears in the performance criterion, and hence hybrid model-predictive control techniques are employed.

Another approach for adaptive cruise control was presented in [187] where a leader-follower problem was considered. In this case an MPC controller regulates the follower so that it keeps a reference separation distance from the leading vehicle. The vehicle dynamics have been represented by a piecewise affine model, modeling nonlinear aerodynamical friction and gear-dependent traction force. The minimum safety distance is enforced as a hard constraint:

$$x_{\text{leader}} - x_{\text{follower}} \geq d_{\text{safe}},$$

where $x_{\text{leader}}$ is the leading car position and $x_{\text{follower}}$ is the following car position. This application has been also used to compare the performance and the complexity of different hybrid model-predictive control strategies.

In [267] a hybrid MPC approach has been proposed for control of semi-active suspensions. As opposed to standard (passive) suspensions, active suspensions are
devices in which an active controller is used to improve the dynamical response of the suspension. In semi-active suspensions a controller is still present, but is passive, that is, it can only regulate the damping coefficient. Even though a semi-active suspension is significantly simpler and less expensive than an active suspension, the capability of modifying the parameters results in an increased performance in terms of driver’s feeling and vehicle handling with respect to passive suspensions. In [267] the employed dynamical model is linear (quarter car model), and the hybrid systems framework is necessary to model the passivity constraint

\[ F_{\text{cmp}}v \geq 0 \iff \begin{cases} F_{\text{cmp}} \leq 0 & \text{if } v \leq 0, \\ F_{\text{cmp}} \geq 0 & \text{if } v > 0, \end{cases} \]

where \( F_{\text{cmp}} \) is the compression force applied by the controller to the suspension, and \( v \) is the velocity of extension of the suspension. The passivity constraint limits the resulting controller to only be able to dissipate energy, and hence the whole suspension system is semi-active.

15.3.3 Survey of MPC applications in powertrain control

In powertrain control, hybrid model-predictive control has been applied for airpath control in diesel engines, control of electronic throttles, control of magnetic actuators, and control of direct injection stratified charge engines.

In [494] the airpath of turbocharged diesel engines has been controlled by means of a switched model-predictive controller. The airpath dynamics is controlled by means of exhaust gas recirculation valves (EGR) and variable geometry turbines (VGT) and it can impressively reduce the emissions of nitrogen oxides (NO\(_x\)) and particulate matter. The approach presented in [494] is based on partitioning the engine speed and the injected fuel into intervals \( I_s^i, i = 1, \ldots, n_s, I_f^j, j = 1, \ldots, n_f \), and to compute a linear approximation of the airpath dynamics in each of the \( n_s \times n_f \) resulting regions. A linear MPC controller is designed for each one of these regions, and during the control cycle, depending on the current engine conditions, the appropriate controller is activated according to the scheme

\[ u(k) = u_{\text{MPC}}^{(ij)}(x(k), r(k), u(k-1)), \quad \text{if speed} \in I_s^i, \text{fuel} \in I_f^j. \]

In [644] hybrid model-predictive control is applied to an electronic throttle. The electronic throttle is one of the fundamental components in drive-by-wire vehicles, and it must be controlled to guarantee high tracking performance, which implies fuel consumption and emissions reduction, while ensuring safety constraints on the system components. In [644] the authors formulate a piecewise affine model of the nonlinearities in the throttle dynamics, such as friction, and the spring characteristics

\[ F_{\text{sp}} = -\theta_h x, \quad \text{if } x \in I_h, \]

where \( x \) is the spring displacement from the neutral position, \( F_{\text{sp}} \) is the spring force, and \( I_h \) and \( \theta_h, h = 1, \ldots, n_x \) are intervals that partition the operating range of \( x \). The
obtained piecewise affine prediction model is used for time-optimal model-predictive control to achieve high tracking performance and safety constraints satisfaction.

In [137] the authors consider the control of magnetic actuators that are in automotive applications, used for example in semi-active suspensions and in fuel delivery systems. Magnetic actuators are devices in which a moving cursor is displaced using the magnetic force generated by a voltage-controlled coil. The control objective is to maximize the tracking performance of a reference signal and to reduce the excitation of manipulated variables. Meanwhile, several constraints have to be enforced that avoid impacts of the cursor with the coil, and that bounds the manipulated variables within safe ranges. The magnetic actuator dynamics are nonlinear, and in [137] the authors have proposed a hybridized model exploited for hybrid model-predictive control. The hybrid MPC is shown to perform better with respect to a linear MPC implementation. This example application will be discussed in detail in Section 15.3.4. The approach was extended in [72] to handle actuators composed of multiple interacting masses, whose model represents a good example of hybrid dynamics.

Hybrid MPC has been applied to direct injection stratified charge (DISI) engines in [268]. DISI engines have two operation modes: homogeneous charge and stratified charge. Homogenous charge is the standard vehicle operation mode. The stratified charge mode is activated by late fuel injection. Stratified charge mode reduces fuel consumption, but it can only be enabled at low to medium engine speeds and loads and it must be periodically disabled to purge a lean NO\textsubscript{x} trap. In [268] hybrid MPC is used to select the discrete operation mode concurrently with the continuous inputs. The resulting vector of manipulated variables is

\[ \mathbf{u}^T = [W_{th}, W_f, \delta, \rho], \]

where \( W_{th} \) is the airflow through the throttle, \( W_f \) is the fuel flow, \( \delta \) is the spark timing, and \( \rho \) is the mode, where \( \rho = 1 \) is homogenous charge and \( \rho = 0 \) is stratified charge. The model-predictive controller achieves fuel consumption reduction, while ensuring torque, air-to-fuel ratio, and spark timing tracking performance and safety constraints on airflow rate, air-to-fuel ratio, and spark timing.

### 15.3.4 Example: Hybrid MPC of magnetic actuators

As an illustrative example of hybrid model-predictive control in automotive application we discuss the control of a magnetic actuator which can be potentially used in fuel delivery systems. The magnetically actuated mass-spring-damper system, whose schematics are shown in Fig. 15.6, is a heterogenous system composed by a mechanical subsystem and an electromagnetic subsystem that influence each other. A mass \( m \) [kg] moves linearly within a bounded region under the effect of a controlled magnetic force \( F \) [N] generated by a coil placed at one of the boundary of the region. Additional forces acting on the mass are generated by a spring and a damper. The overall equations defining the system are
Fig. 15.6 Schematics of a magnetically actuated mass spring damper system.

\[ m\ddot{x} = F - c\dot{x} - kx, \quad (15.1a) \]
\[ \dot{\lambda} = V - R\dot{i}, \quad (15.1b) \]
\[ \lambda = \frac{2k_ai}{k_b + z}, \quad (15.1c) \]
\[ F = \frac{k_ai^2}{(z + k_b)^2} = \frac{\lambda^2}{4k_a}, \quad (15.1d) \]
\[ z = d - x. \quad (15.1e) \]

Equation (15.1a) represents the dynamics of the mass position \( x \) [m] under the effect of the external force \( F \), of a spring with stiffness \( k \) [N/m] and of a damper with coefficient \( c \) [N·s/m]. Equation (15.1b) is Faraday’s law for a resistive circuit with resistance \( R \) [Ω], subject to magnetic flux variations, where the applied voltage \( V \) [V] is the control input. The relation between the magnetic flux \( \lambda \) [V·s] and current \( i \) [A] is defined by (15.1c), where \( k_a, k_b \) are constants, while (15.1d) defines the magnetic force either as a function of the current or as a function of the magnetic flux. Finally, (15.1e) defines the relation between position coordinates in the mechanical (\( x \)) and in the electromagnetic (\( z \)) subsystem.

The magnetically actuated mass spring damper is subject to several constraints related to physical limits and performance. The constraint

\[ -d \leq x \leq d \quad [\text{m}], \quad (15.2) \]

where \( d = 4 \times 10^{-3} \), prevents the moving mass from penetrating the coil or the symmetric stop at the other end and avoids undesirable bouncing with consequent wear of the parts.

The soft-landing constraint

\[ -\varepsilon - \beta(d - x) \leq \dot{x} \leq \varepsilon + \beta(d - x) \quad [\text{m/s}], \quad (15.3) \]

where \( \varepsilon \) and \( \beta \) are constants avoids high velocities when the mass is positioned against the coil with \( (x = d) \). This reduces collision wear and prevents excessive
disturbances to the electrical current. When the mass is at the contact position \((x = d)\), the velocity is constrained in \([-\varepsilon, \varepsilon]\), \(\varepsilon = 0.1\text{m/s}\), a range which is progressively relaxed with rate \(\beta\) as the mass moves away from the coil.

The current in the circuit cannot be negative and, as a consequence of (15.1d), the magnetic force is able to only attract the mass

\[
\begin{align*}
    i & \geq 0 \quad \text{(A)}, \\
    F & \geq 0 \quad \text{(N)}.
\end{align*}
\]  

In addition, the constraint on the voltage \(0 \leq V \leq V_{\text{max}}\) [V] is enforced to take into account the physical limits and the safety of operation of the electrical circuit.

**Decoupled hybrid MPC architecture**  Because of the nonlinearity in (15.1) and of the constraints, nontrivial control techniques are needed to meet the requested specifications. As the dynamics of the electrical subsystem are much faster than the mechanical ones, the control problem can be decoupled by designing an inner-loop controller acting only on the electrical subsystem, and an MPC controller based on the reduced system model

\[
\ddot{x} = -\frac{c}{m} \dot{x} - \frac{k}{m} x + \frac{F}{m},
\]  

where the position \((x)\) and the velocity \((\dot{x})\) of the mass are the state components and the magnetic force \(F\) is the controlled input, subject to constraints (15.2), (15.3), (15.4b), and

\[
F \leq k_a \frac{i_{\text{max}}^2}{(d + k_b - x)^2}.
\]  

Constraint (15.6) defines an upper bound on the available force, related to the maximum available current \(i_{\text{max}}\). The value \(i_{\text{max}}\) is computed from (15.1b) in static conditions, given the maximum voltage \(V_{\text{max}}\). The set (15.6) in the \((x, F)\)-space is the nonconvex hypograph of a convex function. The force command generated by the MPC is used as reference by the inner-loop controller.

Thus, the control system architecture operates as follows:

- the MPC controller computes the force profile \(r_F\), based (15.5) and on the mass reference position \(r_x\);
- the current reference profile \(r_i\) computed from \(r_F\) is used as the reference by the inner-loop controller to regulate the electromagnetic subsystem;
- the inner-loop controller actuates the voltage \(V\) to make the current \(i\) in the electromagnetic subsystem track \(r_i\);
- the current \(i\) generates the actual force \(F\).

A simple way to make the current \(i\) track the desired reference \(r_i\) given the nonlinear dynamics (15.1b), (15.1c), (15.1d) is to design a controller so that the
closed-loop current dynamics become linear first-order dynamics with a stable pole 
\( p_i = -\beta \) and steady-state gain \( \gamma/\beta \). For the considered actuator dynamics (15.1), an approach to obtain this is by feedback linearization. The feedback linearization controller gives good performance in the nominal case, but it is not very robust. Alternative designs may be used to increase robustness.

The overall idea behind the proposed control architecture is to exploit a nonlinear controller to obtain a linearization of the dynamics, and to use a model-predictive controller to meet specifications and performance. In the MPC prediction model we assume that the dynamics of the electrical subsystem in closed-loop with the inner-loop controller are infinitely fast, hence \( F = r F \). Thus, we will refer to \( F \) as the output of the MPC controller.

A hybrid MPC controller is needed to account for the nonconvex constraint (15.6). The nonlinear constraint is approximated by a piecewise affine function so that the prediction model is formulated as a piecewise affine system used in the hybrid MPC algorithm.

**Actuator model** We approximate the upper bound of (15.6) as a piecewise affine function of the position \( \bar{F}(x) = r_i x + q_i \), if \( x \in [\bar{x}_i, \bar{x}_{i+1}) \), \( i = 0 \ldots \ell - 1 \), where \( \bar{x}_i < \bar{x}_{i+1} \). The points \( \{\bar{x}_i\}_{i=1}^{\ell - 1} \) are the function breakpoints and define the borders between regions in which \( \bar{F}(x) \) has different affine terms. Next, we introduce \( \ell - 1 \) binary variables \( \delta_1, \ldots, \delta_{\ell-1} \in \{0, 1\} \) defined by the logical conditions

\[
[\delta_i = 1] \leftrightarrow [x \leq \bar{x}_i], \quad i = 1, \ldots, \ell - 1 \quad (15.7)
\]

and \( \ell - 1 \) continuous variables \( z_1, \ldots, z_{\ell-1} \in \mathbb{R} \) defined by

\[
z_i = \begin{cases} (r_{i-1} - r_i) x + (q_{i-1} - q_i) & \text{if } \delta_i = 1 \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, \ldots, \ell - 2, \quad (15.8a)
\]

\[
z_{\ell-1} = \begin{cases} r_{\ell-2} x + q_{\ell-2} & \text{if } \delta_{\ell-1} = 1 \\ r_{\ell-1} x + q_{\ell-1} & \text{otherwise} \end{cases}. \quad (15.8b)
\]

Then, the piecewise affine approximation is

\[
\bar{F}(x) = \sum_{i=1}^{\ell-1} z_i. \quad (15.9)
\]

Relations (15.7), (15.8c), (15.9) can be embedded into an MLD system, using for instance the modeling language HYSDEL [632], along with the logical constraints

\[
[\delta_i = 1] \to [\delta_{i+1} = 1], \quad i = 1, \ldots, \ell - 1 \quad (15.10)
\]

which are included to largely simplify the complexity of the MPC optimization problem associated with the MLD model.

We consider here \( \ell = 3 \), and, as a consequence, two \( \delta \) and two \( z \) auxiliary variables have been introduced and (15.6) is approximated as

\[
u \leq z_1 + z_2. \quad (15.11)
\]
Model (15.5), (15.7), (15.8c), (15.11), and the mechanical subsystem model (15.5) sampled with $T_s = 0.5 \text{ms}$ can be modelled in HYSDEL to get the equivalent MLD model

$$\xi(k + 1) = A\xi(k) + B_1u(k) + B_2\delta(k) + B_3z(k), \quad (15.12a)$$
$$y(k) = C\xi(k) + D_1u(k) + D_2\delta(k) + D_3z(k), \quad (15.12b)$$
$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4\xi(k) + E_5, \quad (15.12c)$$

with $\xi^T = [x \; \dot{x}], y^T = [x \; \beta x + \dot{x} - \beta x + \dot{x}]$.

Control design The second and the third output are used to enforce (15.3) as output constraints. The hybrid MPC optimization problem is formulated as

$$\min_{U(k)} \|\xi(N|k) - r_\xi(k)\|_2^{Q_N} + \sum_{i=0}^{N-1} \|\xi(i|k) - r_\xi(k)\|_2^{Q_\xi} + \|u(i|k)\|_2^{Q_u} \quad (15.13a)$$

s.t.:

MLD dynamics (15.2),
$$y_{\min} \leq y(i|k) \leq y_{\max}, \quad i = 1, \ldots, N, \quad (15.13b)$$
$$u_{\min} \leq u(i|k) \leq u_{\max}, \quad i = 0, \ldots, N - 1, \quad (15.13c)$$
$$\xi(0|k) = \xi(k), \quad (15.13d)$$

where $\|a\|_2^Q = a^TQa$, $U(k) = \{u(i|k)\}_{i=0}^{N-1}$, $r_\xi^T = [r_x \; 0]$ and $r_x$ is an externally generated reference, (15.13c) models (15.2) and (15.3), and (15.13d) models (15.4b). Here, we choose

$$Q_\xi = Q_N = \begin{pmatrix} 2 \cdot 10^6 & 0 \\ 0 & 10^{-7} \end{pmatrix}, \quad Q_u = 10^{-7}, \quad N = 3,$$
$$y_{\min} = \begin{pmatrix} -4 \cdot 10^{-3} \\ -10.2 \end{pmatrix}, \quad y_{\max} = \begin{pmatrix} 4 \cdot 10^{-3} \\ 10.2 \end{pmatrix}, \quad u_{\min} = 0, \quad u_{\max} = +\infty,$$

and the approximation of (15.6), which is embedded into the MLD model, are always enforced as hard constraints.

Results In Fig. 15.7 the behavior of the system in closed-loop with the hybrid MPC controller (15.13) is reported. The dashed line shows the behavior of the system in closed loop with a linear controller that enforces (15.6) by a cascaded saturator instead of in the MPC. The performance improvements are evident in terms of reduced overshoot and settling time.

Due to the fast sampling time and to the implementation in a simple microcontroller, the MPC controller cannot be executed in its on-line form, where an optimization problem has to be solved at every control cycle. Instead, the explicit hybrid feedback law [10] is computed. The resulting controller is automatically generated as a C-code through the Hybrid Toolbox [57] and has an average execution time of 0.025 ms and a worst case of 0.3 ms on a 2-GHz Pentium-M, with 1 Gb RAM. The worst case number of operations to be executed is of about 12 000 multiplications, 9000 sums, and 4000 comparisons per control cycle. By an architecture simulator

...
Evaluation of the results The automotive control problems considered in this section are relatively simple (few states and control variables), yet they exhibit hybrid features and require small sampling times. Therefore, despite their apparent simplicity, they are quite challenging from a control viewpoint. Hybrid modeling based on HYSDEL or piecewise affine models is a quite convenient approach to come up with a compact dynamical model of the open-loop process and its operating and dynamical constraints, capturing the most relevant aspects of the problem.

Regarding the implementation, the hybrid MPC controllers are more computationally demanding than standard techniques used in automotive applications, such as feedforward (open-loop) control or PID controllers. The MPC controllers can certainly take advantage of the increased amount of computational power embedded in the electronic control units, and ad-hoc computation reduction techniques can be applied to the MPC controller. Also, the controller complexity can often be tuned by appropriate selection of the prediction model and of the prediction horizon until a desired trade-off between performance and hardware cost is found. For instance in [137] it has been shown that for different MPC controllers for magnetic actuators, the number of computations is feasible for different classes of automotive hardware.
15.4 Control of a homogeneous charge compression ignition engine

The homogeneous charge compression ignition (HCCI) engine with its excellent potential for combining low exhaust emissions with high efficiency gained substantial interest towards the end of the twentieth century [171]. The HCCI engine combines features of the traditional spark ignited (SI) Otto-cycle and compression ignited (CI) Diesel-cycle engines to a new engine concept.

15.4.1 HCCI engine concept

HCCI operation can be two-stroke or four-stroke, and the first studies [489, 493] were performed on two-stroke engines. Later studies [171, 614] on four-stroke engines show that high efficiency can be combined with low NOX emissions for HCCI engines running with a high compression ratio and lean operation. This section will focus exclusively on the four-stroke version of the HCCI engine.

**HCCI cycle** The four-stroke HCCI cycle can be described by its four strokes: intake, compression, expansion, and exhaust. During the intake stroke, a more or less homogeneous mix of fuel and air is inducted into the cylinder. During the compression stroke, this charge is compressed by the upward motion of the piston. Towards the end of the compression stroke, temperature and pressure have reached levels where pre-combustion reactions start to take place. Somewhere near the TDC (top dead center), actual combustion starts (Fig. 15.8). During the initial part of the expansion stroke, the bulk of combustion takes place during the course of a few crank-angle degrees. During the rest of the expansion stroke, the high pressure caused by combustion forces the piston down towards the bottom dead center (BDC). In the exhaust stroke, the upward motion of the piston forces the exhaust gas to leave the cylinder through the exhaust valve. Hence, the coexistence of physics-based continuous-time dynamics with aperiodic four-stroke engine operation determines a clearcut hybrid system.

**Ignition** An HCCI engine, unlike to SI and Diesel-cycle CI engines, has no direct means for controlling ignition timing. The SI engine has spark timing, and the Diesel-cycle CI engine has the start of fuel injection, which both directly control the onset of combustion. However, for an HCCI engine, ignition timing is dictated by the conditions of the charge and the cylinder walls at the time when the intake valve closes. It can only be controlled indirectly through adjustments in the cylinder charge preparation. This is one of the biggest challenges with practical implementation of HCCI engine technology. The following paragraphs will describe the most important variables that control ignition timing for an HCCI engine.

The temperature of the air when it enters the cylinder has a large influence on the charge temperature towards the end of the compression stroke. With a compression ratio of 18:1, a change in intake temperature by 30 K will result in a change in temperature at TDC by almost 100 K. Since temperature is a very important factor
in auto-ignition, an increase in intake temperature will have a very strong advancing influence on ignition timing.

The portion of the exhaust gas that is not expelled during the exhaust stroke and which is called the residual gas, is particularly important for HCCI operation. The thermal energy provided by the residual gas contributes to heating the charge of the following cycle, and affects the crank angle at which ignition takes place. On an engine with variable valve timing, the residual gas fraction can be controlled by early closing of the exhaust valve, which will trap a larger amount of exhaust gas in the cylinder for the following cycle. It is necessary to remember though that exhaust gas also acts as a diluent, and thereby slows down the combustion chemistry. This will tend to retard ignition timing, and with a very high residual-gas fraction this effect will dominate.

Closely related to residual gases is the exhaust gas recirculation (EGR). This refers to exhaust gas that is routed back from the exhaust manifold to the intake manifold. Combined with an EGR cooler, this can be used for diluting the charge and thus lowering the reaction rate. An increase in EGR rate will retard ignition timing.

Another important factor is the cylinder wall temperature. Hot cylinder walls will heat the charge throughout the intake and compression strokes, and will advance ignition timing.

The fuel–air equivalence ratio affects both fuel concentration and oxygen concentration. However, since HCCI engines operate lean, the equivalence ratio has a stronger influence on fuel concentration than on oxygen concentration. The dominating effect of increasing the equivalence ratio, thus, is an increase in fuel concentration,
which will result in a higher reaction rate. Thus, increasing the equivalence ratio serves to advance ignition timing.

Another possible way to control ignition timing is by changing the fuel composition. Addition of a second fuel with higher reactivity will serve to advance ignition timing. Examples are the addition of hydrogen to natural gas and $n$-heptane to iso-octane.

A variable compression ratio provides an effective means of controlling the temperature towards the end of the compression stroke. A higher compression ratio increases the charge temperature near the TDC, and tends to advance ignition timing.

Charge stratification, which means inhomogeneous charge distribution can be used to locally increase the equivalence ratio, and thus the reaction rate, in order to advance the ignition timing. Charge stratification can be achieved through late fuel injection. As a drawback locally high temperatures, result which cause an increase in NO$_X$ production.

Evidently, there are many variables that affect ignition timing, but they all do so in non-trivial ways. Furthermore, many of the variables affect each other as well. Actually, some variables are affected by ignition timing itself, for example the cylinder wall temperature increases with advanced ignition timing. When ignition timing is advanced, the peak cylinder temperature increases which, in turn, causes an increase in cylinder wall temperature. It follows that ignition timing is very sensitive to operating conditions.

### 15.4.2 Control of ignition timing

It is evident from above that ignition control is much more challenging for an HCCI engine than for an SI or Diesel-cycle CI engine. The most readily available means of controlling ignition timing is by adjusting the fuel composition. This does not require any novel mechanical design like variable valve timing or variable compression ratio. It merely requires a doubling of the port fuel injection system.

#### Selection of the feedback structure

The sensitivity of ignition timing to operating conditions does not allow an open-loop solution in the form of a look-up table. Furthermore, the system becomes unstable for some operating conditions at high load. Thus, closed-loop control is an absolute necessity, which poses the question of which variables should be fed back.

Cylinder pressure is the natural choice of the control variable, since ignition is an in-cylinder phenomenon (Fig. 15.8). What characteristic of the cylinder pressure trace reflects when combustion takes place, though?

The crank angle of maximum pressure gives some information about when the bulk of combustion is taking place, but for combustion timing before or near the TDC, this angle tends to gravitate towards TDC due to its dependence on volume in the ideal gas law. Furthermore, for very late combustion timing, the pressure maximum from compression dominates the one from combustion. Another problem is that the maximum is flat, which makes the detection of combustion ambiguous.
The crank angle of maximum pressure derivative suffers from the same problems as the crank angle of maximum pressure in addition to the inherent noise problems with numerical differentiation. Another possibility is to search for the inflection point, where the pressure trace transitions are from negative to a positive second derivative due to the onset of combustion. This also suffers from the problems with numerical differentiation.

It turns out that a first-principles analysis based on the pressure measurements and heat release analysis provides a very robust source of feedback. Heat release analysis applies the first law of thermodynamics to the combustion chamber during the entire combustion event in order to estimate the rate at which chemical energy is converted to thermal energy. If no adjustments are made for heat transfer or flow into and out of crevices, the net heat release is obtained. Integration with respect to the crank angle yields the cumulative heat release, which roughly reflects the mass fraction burned.

Combustion in an HCCI engine is usually very fast. The mass fraction burned usually goes from 10% to 90% in about 5 crank angle degrees, which means that the crank angle of 50% heat release, CA50, provides a very accurate measure of when combustion is taking place. In the following, CA50, combustion timing, and ignition timing will be used interchangeably to denote the crank angle of 50% heat release.

15.4.3 Properties and model of the HCCI engine

Two fuel injectors and one cylinder pressure sensor per cylinder allows separate control loops for each cylinder. Thus, the control structure indicated in Fig. 15.9 can be used for the combustion timing control of each cylinder. Fuel octane number is a measure of a fuel’s resistance to auto-ignition, and can be used as the control input for combustion timing control of an HCCI engine cylinder. When using a mixture of iso-octane and n-heptane, the octane number is, by definition, the percentage of iso-octane.

![Fig. 15.9 Control structure for combustion timing control. The CA50 based on cylinder pressure measurements provides feedback about combustion timing, and octane rating provides a control input.](image)

**Processing cylinder pressure measurements**  Cylinder pressure measurements are normally performed with either piezoelectric elements combined with charge
amplifiers or with fiber-optical sensors. Both methods fail to measure the DC component of the cylinder pressure. A thermodynamically based method of estimating the DC component is detailed in [637], and amounts to estimating an initial pressure and a measurement offset based on pressure measurements during the compression stroke. It is essential to select the crank-angle interval for estimation between the intake-valve closing and the start of combustion for the thermodynamical assumptions to hold.

The cylinder pressure measurements $p_m$ can be decomposed into the actual pressure $p$ and a sensor offset $\Delta p$:

$$p_m = p + \Delta p.$$  

(15.14)

The real pressure can be modeled with polytropic compression

$$p = CV^{-\kappa},$$  

(15.15)

where $V$ is the combustion chamber volume $\kappa$ is the polytropic exponent, and $C$ depends on the initial pressure according to:

$$C = p_0 V_0^{\kappa}.$$  

(15.16)

In [637], a method of estimating the polytropic exponent is also provided. In cases where the polytropic exponent is thought to vary significantly from cycle to cycle, this method can be used. Intake temperature and fuel composition as well as the equivalence ratio affect the polytropic exponent.

![Fig. 15.10 Definitions of some heat-release based cycle parameters.](image)

**Heat release analysis** An analysis of the combustion chamber, based on the first law of thermodynamics, relates the rate at which chemical energy is converted to
thermal energy to the pressure in the combustion chamber. This type of analysis is conventionally called heat release analysis, and can be used to determine when combustion is taking place. This term stems from the simplification that is normally done, in which the charge composition is assumed to be constant and the increase in internal energy is interpreted as heat. If the actual heat transfer to the cylinder walls as well as crevice flow is neglected, (15.17) relates heat release to cylinder pressure:

$$\delta Q_{ch} = \frac{c_v}{R} V dp + \frac{c_p}{R} p dv. \quad (15.17)$$

This equation is integrated over a crank angle interval which includes the whole combustion event. The parameters $c_v$ and $c_p$ are the specific heats at constant volume and pressure, respectively, and technically depend on temperature. However, if the only objective is to determine combustion timing, they can be assumed to be constant. $R$ is the universal gas constant.

The result of the integration of the heat release equation is the cumulative net heat release as a function of crank angle $Q_{ch}(\alpha)$. A typical heat release trace is plotted in Fig. 15.10 together with some definitions. The most important definition in this context is CA50, the crank angle of 50% heat release. Since combustion is very fast, CA50 can be used as a robust source of feedback for combustion timing.

**HCCI engine** The sensitivity of CA50 to changes in fuel octane number varies by orders of magnitude for different operating points (Fig. 15.11). Each line in the plot represents a specific intake temperature and load. Within each line, the fuel octane number has been varied to achieve an interval of combustion timings. The strong nonlinearity of the plant makes a linear controller unsuitable for the task. The situation can be remedied, however, if the sensitivity is mapped over the multidimensional space of operating conditions. This map can be used for gain-scheduling the otherwise linear controller. In [492], a multivariable function is fitted to measure-
ments of the sensitivity of CA50 to changes in octane number for a multitude of operating conditions. In order to get a simple, computationally inexpensive model, the sensitivity is modeled as a product of functions of one variable each. This approach is entirely empirical, but yields a sensitivity model with acceptable residuals (within 3%). The variables that are included in this model are engine speed, inlet air temperature, fuel octane number, fuel mass per cycle, and CA50. A later model revision includes inlet pressure as well.

Figure 15.12 shows the octane number component of the sensitivity function. The sensitivity model is used for scaling the controller gains, which implicitly assumes that the dynamical behavior of the plant is independent of the operating point. Only the DC gain of the plant changes.

A minimum requirement of physical modeling is the explanation of the nature of the in-cylinder pressure traces (Fig. 15.8) where adiabatic compression combines with fuel-dependent auto-ignition [83]. In previous work, modeling choices involved aspects of chemical kinetics, cycle-to-cycle coupling, in-cylinder concentrations of reactants, wall temperature dynamics, pressure dynamics, and auto-ignition timing. Modeling details fell into categories of single-zone model, multi-zone models, multi-dimensional computational fluid dynamics (CFD) models, sometimes combined with exosystem simulation on the form of stochastic disturbances, load modeling, sensor modeling [74].

For each of the approaches investigated, the modeling was based on an open system first law analysis, with steady state compressible flow relations used to model the mass flow through the intake and exhaust valves. The model includes nine states: the temperature $T$, the concentrations of propane $[C_3H_8]$, oxygen $[O_2]$, nitrogen $[N_2]$, carbon dioxide $[CO_2]$, water $[H_2O]$, the mass in the exhaust manifold $m_e$, the internal energy of the product gases in the exhaust $u_e$ and carbon monoxide $[CO]$, the crank angle $\theta$, the cylinder volume $V$, and for HCCI, an integrated Arrhenius rate to capture ignition timing [74]. Moreover, Shaver et al. [581] singled out six distinct stages in modeling of HCCI engine operation: induction, compression, combustion, expansion, exhaust and residence in the exhaust manifold. As both physical aspects and operational aspects require attention, hybrid modeling turns out to be instrumental for HCCI engine control [74].

**Open-loop stability** An interesting phenomenon that appears in some regions of the operating space of an HCCI engine is open-loop instability. This phenomenon results when wall temperature effects dominate the ignition dynamics. Figure 15.13 shows the open-loop behavior at a stable and an unstable operating point, respectively. All control inputs are held constant in both cases.

The cause of instability is the positive thermal feedback provided through the interaction between ignition timing and cylinder wall temperature. A small increase in cylinder wall temperature results in a hotter cylinder charge, which advances ignition timing. Advanced ignition timing, however, results in higher gas temperature and more heat transfer to the walls, thus higher wall temperature. The reversed case is a small drop in cylinder wall temperature, which results in cooler cylinder charge. Ignition timing is retarded, which results in lower gas temperature, which in turn
reduces the heat transfer to the walls, and thus the wall temperature. It is evident that operating points where this effect dominates are unstable.

The positive feedback mentioned above is always present, but not all operating points are unstable (Fig. 15.13). The stabilizing negative feedback is closely related to the destabilizing positive feedback. Early ignition leads to high peak temperature and heat transfer, but this results in lower gas temperature towards the end of the cycle, which means both colder residual gas and a larger amount of it. This reduces the reactivity of the charge for the next cycle, and retards ignition timing. The opposite holds for late ignition. Thus, the residual gas provides the stabilizing negative feedback.

Fig. 15.12 The octane number component of the sensitivity function.

Fig. 15.13 Repeated open-loop operation at one stable and one unstable operating point.
Bibliographical notes

As for stable operation, combustion phasing control design requires appropriate models and system output variables usable for feedback control. Recently, mode-transition operation and control of Diesel-HCCI and SI-HCCI engines and other hybrid control aspects have received attention [582].

In order to avoid difficult conditions of HCCI operation at high loads, mode transition control is relevant and, thus, hybrid modeling for switching between SI and HCCI (or Diesel-HCCI) combustion mode [582].

Recently, successful model-based HCCI engine control using conservation principles in hybrid modeling was reported [85]. Hence, hybrid modeling of HCCI engines suitable for control design is a key issue for progress in this area.
References


518 References


References 523


References


[638] Uppsala University and Aalborg University. UPPAAL version 4.0 online help, 2006. Available at: http://www.it.uu.se/research/group/darts/uppaal/index.shtml.


