MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Automotive Applications of Model Predictive Control

Di Cairano, Stefano; Kolmanovsky, Ilya V.

TR2018-213 December 24, 2019

Abstract

Model Predictive Control (MPC) has been investigated for a significant number of potential applications to automotive systems. The treatment of these applications has also stimulated several developments in MPC theory, design methods and algorithms, in recent years. This chapter provides an overview of automotive applications for which MPC has been considered and approaches to MPC development, deployment and implementation that have been pursued. First, a brief history of MPC applications to automotive systems and features that make MPC appealing for such applications are discussed. Then, for the main automotive control sub-domains, key first principle models and opportunities that these provide for the application of MPC are described. Next, we detail the key steps and guidelines of the MPC design process which is tailored to automotive systems. Finally, we discuss numerical algorithms for implementing MPC, and their suitability for automotive applications.

Handbook of Model Predictive Control (book)

Automotive Applications of Model Predictive Control

Stefano Di Cairano and Ilya V. Kolmanovsky

Model Predictive Control (MPC) has been investigated for a significant number of potential applications to automotive systems. The treatment of these applications has also stimulated several developments in MPC theory, design methods and algorithms, in recent years. This chapter provides an overview of automotive applications for which MPC has been considered and approaches to MPC development, deployment and implementation that have been pursued. First, a brief history of MPC applications to automotive systems and features that make MPC appealing for such applications are discussed. Then, for the main automotive control sub-domains, key first principle models and opportunities that these provide for the application of MPC are described. Next, we detail the key steps and guidelines of the MPC design process which is tailored to automotive systems. Finally, we discuss numerical algorithms for implementing MPC, and their suitability for automotive applications.

I. Model Predictive Control in Automotive Applications

There are very few devices that are as pervasive in our world as cars. Reports show that close to 90 million cars and light commercial vehicles were sold worldwide in 2016. Recent innovations in car mechanics, electronics and software have been fast paced to respond to growing stringency of fuel economy, emissions and safety regulations, as well as to market-driven pressures to provide customers with improved performance, drivability and novel features. Advanced control methods that are capable of optimizing the vehicle operation, and can reduce the time-to-market for increasingly complex automotive systems are clearly needed.

Stefano Di Cairano is Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA, email: dicairano@ieee.org

Ilya V. Kolmanovsky is with Dept. Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA email: ilya@umich.edu

It thus comes as no surprise that, in recent years, a significant interest in model predictive control (MPC) has been shown in the automotive industry. The research on applications of MPC to automotive systems has been steadily growing both in industry and academia to address some of the challenges of this application domain. Yet MPC is a significant step up from the classical control methods, such as PID, and its implementation in industrial practice presents challenges on its own.

The purpose of this chapter is to provide a short tutorial on the development of MPC-based solutions for automotive systems. Towards this end, we first briefly review the history of MPC applications to automotive systems, and we highlight the benefits that MPC can provide as well as the challenges faced by MPC in this domain. Then, given that MPC is a model-based control approach, for the main automotive control areas, such as powertrain control, chassis control and energy management, we describe the key first principle models that can be used for MPC design, and the control objectives which need to be achieved. Next, we detail common steps of MPC design for automotive systems. Finally, we consider the computational aspects that are important for real-time implementation and deployment of MPC solutions on automotive computing platforms.

While this chapter represents a tutorial overview of MPC design for automotive systems based on the author's first-hand experience, due to scope and length limitations it not able to serve as a comprehensive survey of the entire body of literature on automotive applications of MPC. A brief survey is available in [46].

A. A Brief History

Some of the first investigations of MPC for automotive systems can be traced back to the mid ′90s, with [44] where MPC was applied to idle speed control being a notable case. In those years, the numerical algorithms for MPC were too computationally demanding for the "then-current" vehicle micro-controllers, and hence such studies were usually only simulation-based.

Two new developments in the early 2000s gave a significant boost to the investigation of MPC-based automotive control and have led to the rapid growth of related applications and literature. Firstly, the scientific community interest in hybrid dynamical systems led to the development of hybrid MPC [8], which allowed to control processes with switching dynamics. This opened up opportunities for MPC applications to control of transmissions [2], [7], [40], [76], to traction control [13], and to control of semiactive suspensions [36]. Systems with mode-dependent objectives, such as direct injection, stratified charge engines [37], or requiring piecewise linearizations, such as camless engine actuators [21], HCCI engines [10], [66], or vehicle stability control functions [27] could now be handled. Secondly, the application of parametric programming techniques resulted in the development of explicit MPC [9] that synthesizes the control law, and hence avoids the need to run an optimization algorithm online in the micro-controller. This led to the possibility of experimentally testing several controllers in real, production-like, vehicles including, about 12 years after the initial development, a refined MPC-based idle speed control [28], and an MPC-based diesel engine airpath control [62], [73]. From then, the applications of MPC have picked up both in powertrain control [23] and chassis (or vehicle dynamics) control [5], [27], with some industry research centers being at the forefront in developing these applications, see, e.g., [46], [59], [77].

Starting from the mid-2000s, MPC-based control has been considered for hybrid and electric vehicles, including fuel-cell vehicles. Some of the early contributions include [6], [56], [75]. The development of MPC strategies for different hybrid electric powertrain configurations has then been considered in more depth, e.g., for ERAD [68], series [25] and powersplit [12] configurations. Due to the complexity of the hybrid powertrains and the attempt to use MPC to directly optimize fuel consumption, these controllers were often rather difficult to implement in the vehicles. An interesting case is [25], where instead of optimizing directly the fuel consumption, MPC was used as an energy buffer manager to operate the engine smoothly and with slow transients, leading to a design simple enough to be implementable in a prototype production vehicle, yet still achieving significant benefits in terms of fuel economy. The resulting controller in [25] was actually implemented experimentally in such road-capable vehicle, which allowed to assess its performance in production-like computing hardware.

Currently, advanced MPC methods are being investigated both for improving existing features, and for future applications in autonomous, and connected vehicles. Some examples are Lyapunov-based MPC for network control in automotive systems [15], stochastic MPC for cooperative cruise control [72], robust and stochastic MPC for autonomous driving [16], [24], [51], and several applications exploiting V2V and V2I communications [57], [61]. Such an expansion has been also supported by the development of low complexity optimization algorithms that now allow for solving quadratic programs in automotive micro-controllers without the need to generate the explicit solutions, that have combinatorial complexity in terms of memory and computations. Still several challenges in terms of computation, estimation, and deployment remain, that will require significant investigations in the next several years, to increase the range of feasible applications. How ongoing advances in the areas of cloud computing, connectivity, large data sets, and machine learning can help tackle these challenges is also to be fully discovered.

B. Opportunities and Challenges

Due to regulations, competition, and customer demands, automotive control applications are driven by the need for robustness, high performance, and cost reduction all at the same time. The investigation of MPC for several automotive control problems has been mainly pursued due to MPC features that are helpful and effective in addressing such requirements and in achieving optimized operation. The key strengths of MPC are summarized in Table I and discussed next.

Strengths	**Challenges**
Simple multivariable design	High computational load
Constraint enforcement	Process models sometimes unavailable
Inherent robustness	Nonlinearities during transients
Performance optimization	Dependence on state estimate quality
Handling of time delays	Non-conventional design and tuning process
Exploiting preview information	

TABLE I

STRENGTHS AND CHALLENGES FOR MPC IN AUTOMOTIVE APPLICATIONS.

A solid starting point for MPC development is that while the processes and dynamics taking place in the vehicle are interdependent and may be fairly complex, they are well studied and understood, and, for most, detailed models are available. This enables the application of model-based control methods, such as MPC.

Due to the aforementioned requirements, often driven by emissions, fuel consumption, and safety regulations, the number and complexity of actuators for influencing the vehicle operation is increasing. Some interesting examples are turbochargers, variable cam timing, electric motors, variable steering, differential braking, regenerative braking. As more actuators become available, methods that can coordinate them to achieve multiple objectives, i.e., control multivariable, multiobjective systems, may achieve superior performance then control designs that are decoupled into several single-variable loops. MPC naturally handles multivariable systems without additional design complexity, thus simplifying the development of multivariable controllers. This has been demonstrated, for instance, for spark-ignition (SI) engine speed control [23], [29], [44], vehicle-stability control by coordinated steering and braking [27], [31], and airpath control in turbocharged diesel engines [62], [73]. Furthermore, while it may still be difficult to obtain globally robust MPC designs, it is well known that often MPC provides inherent local robustness, as it can be designed to locally recover the LQR behavior, including its gain and phase margin guarantees.

Another advantage is that the tight requirements imposed by operating conditions, regulations, and interactions with other vehicle systems can often be easily formulated in terms of constraints on process variables. By enforcing constraints by design, rather than by time-consuming tuning of gains and cumbersome protection logics, MPC can reduce the development and calibration time by a significant amount [14], [23], [27], [36], [73].

The problem of ensuring high performance can often be approached through the optimization of an objective function. The ability to perform such an optimization is another key feature of MPC. In fact, this was at the root of the interest of several researchers in hybrid and electric vehicles [12], [25], [68], [79]. Even if it may be difficult to directly formulate the automotive performance measures as a cost function for MPC, it is usually possible to determine indirect objectives [25], [29] that, when optimized, imply quasi-optimal (or at least very desirable) behavior with respect to the actual performance measured.

Besides these macro-features, MPC has additional capabilities that are useful in controlling automotive processes. For instance, the capability of including time delay models, possibly of different length in different control channels, is very beneficial, as several engine processes are subject to transport delays and actuator delays. Also, new technologies and regulations in communication and connectivity, outside and inside the car, allows for obtaining preview information that MPC can exploit to achieve superior performance [30], [72]. This is even more relevant in the context of autonomous and connected vehicles [19], due to the available long term information, for instance from mid to long range path planners, and from shared information among vehicles.

However, there are also several challenges to the large scale deployment of MPC in automotive applications [17], which are also summarized in Table I and discussed next.

First, MPC has larger computational load and memory footprint than classical control methods, while automotive micro-controllers are fairly limited in terms of computing power. Since the vehicle must operate in challenging environments, e.g., temperatures ranging from -40^{o}C to $+50^{o}$C, the achievable processor and memory access frequencies are limited. The need to reduce the cost, and the development and validation time often prevents to introduce new processors sized for the need of a specific controller. Rather, the controller must fit in a given processor.

Second, not all the automotive processes have well-developed models. Combustion and battery charging/discharging are examples of processes that are still difficult to model precisely, and suitable models for them still remain an area under study. While some of the gaps can be closed using partially data-driven models, one has to be careful in applying MPC in this setting.

Even for the processes that are better understood, the dynamics are intrinsically nonlinear. This third challenge is more relevant in automotive than in other fields, e.g., in aerospace, because, due to external effects, e.g., the driver, the traffic, the road, many automotive processes are continuously subject to fast transients during which the nonlinearities cannot be easily removed by linearization around a steady state.

A further complicating factor is that several variables in automotive processes are not measured, and the sensors for estimating them may be heavily quantized and noisy. A fourth challenge for MPC, which needs the state value for initializing the prediction model, is the need of state estimators, whose performance will significantly affect the overall performance of the control system. The estimator performance will depend on the sensors that in automotive applications are reduced in number and have limited capabilities, once again due to cost and harsh environment.

Fifth and final challenge, is the difference in the development process of MPC and classical controllers, e.g., PID. While the latter are mostly calibrated by gain tuning, MPC requires prediction model development and augmentation, definition of horizon and cost, and tuning of the weights of the cost function terms. As these are often not taught in basic control design courses, calibration engineers in charge of deploying and maintaining the controllers in the vehicle may find difficulties with the development of MPC. Hopefully, this handbook is a step towards solving this problem.

C. Chapter Overview

The rest of this chapter is structured based on the above discussion of strengths and challenges, with the aim of providing a guide for MPC development in automotive applications.

Due to the model-based nature of MPC, and the need for the MPC developer to acquire an understanding of the process models used for design, we first describe (Section II) the key models to be used for MPC developments in the areas of powertrain (Section II-A), vehicle dynamics (Section II-B) and energy management (Section II-C). Our description of such models provides a starting point for the development of MPC solutions for these applications and enhances the understanding of the opportunities for using MPC in these applications. Then, we provide general guidelines for controller development (Section III). Finally, we discuss the computational challenges and the key features of the algorithms used for MPC deployment in automotive applications (Section IV).

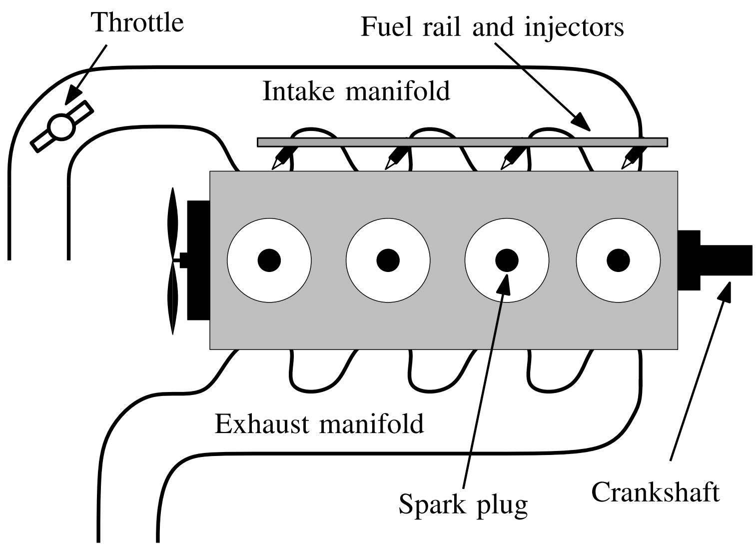

Fig. 1. Schematics of a naturally aspirated spark ignition engine, with focus on the air path.

II. MPC FOR POWERTRAIN CONTROL, VEHICLE DYNAMICS AND ENERGY MANAGEMENT

In this section we consider key automotive control areas in which the application of MPC has been considered and can have an impact. For each area we first describe the key models for model-based control development, and then, in light of these models, we briefly highlight what impact MPC may have.

A. Powertrain Control

Powertrain dynamics involve the generation of engine torque and transfer of such torque to the wheels to generate traction forces.

The engine model describes the effects of the operating conditions and engine actuators on the pressures, flows and temperatures in different parts of the engine, and on the torque that the engine produces. The engine actuators range from the standard throttle, fuel injectors, and spark timing, to more advanced ones, such as variable geometry turbines (VGT), exhaust gas recirculation (EGR) valves, and variable cam timing (VCT) phasers, among others.

The engine model itself is in general composed of two parts, the airpath model, which describes the flow and mixing or the different gases in the engine, and the torque production model, which describes the torque generated from the combustion of the gas mixture.

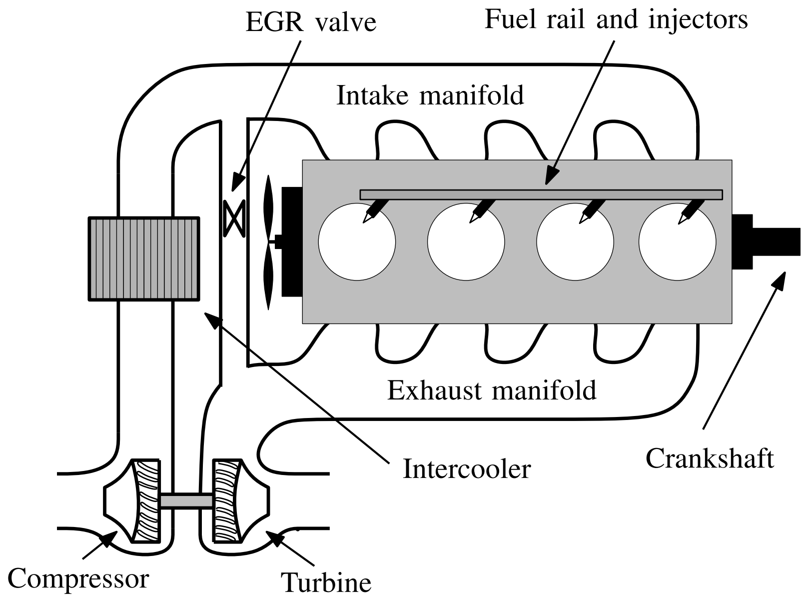

Fig. 2. Schematics of a turbocharged compression ignition engine, with focus on the air path. In comparison with the SI engine in Figure 1, notice the absence of throttle and spark plugs, and the interconnected dynamics of exhaust and intake manifold, through EGR valve and turbine-compressor.

For naturally aspirated spark ignition (SI), i.e., conventional gasoline, engines (see the schematic in Figure 1) the airpath model is relatively simple and represents the cycle averaged dynamics of the pressure in the intake manifold, under an isothermal assumption, and the flow from the throttle to the intake manifold and from the intake manifold into the engine cylinders,

$$\dot{p}_{\rm im} = \frac{RT_{\rm im}}{V_{\rm im}}(W_{\rm th} - W_{\rm cyl}), \tag{1a}$$

$$W_{\rm cyl} = \eta_{\rm vol}\frac{V_d p_{\rm im}}{RT_{\rm im}}\frac{N}{120} \approx \frac{\gamma_2}{\gamma_1} p_{\rm im} N + \gamma_0, \tag{1b}$$

$$W_{\rm th} = \frac{A_{\rm th}(\vartheta)}{\sqrt{RT_{\rm amb}}} p_{\rm amb}\phi\left(\frac{p_{\rm im}}{p_{\rm amb}}\right), \tag{1c}$$

where W, p, T, V, denote mass flow, pressure, temperature, and volume, respectively, ϕ is a nonlinear function which represents the throttle flow dependence on the pressure ratio across the throttle [42, App.C], the subscripts *im*, *th*, *amb*, *cyl* refer to the intake manifold, the throttle, the ambient, and the cylinders, respectively, N is the engine speed, usually in revolutions per minute (RPM), V_d is the engine displacement volume, $\eta_{\rm vol}$ is the volumetric efficiency, R is the gas constant, $A_{\rm th}$ is the throttle effective flow area, which is a function of throttle angle, ϑ, and γ_i, $i \in \mathbb{Z}_{0+}$ denote engine-dependent constants, which are obtained from engine calibration data.

For modern compression ignition (CI), i.e., diesel, engines, (see the schematic in Figure 2), the airpath model is substantially more complex, especially because these engines are usually turbocharged and exploit EGR, which renders the isothermal assumption inaccurate. Furthermore, the EGR valve and the turbocharger effectively couple the intake manifold with the exhaust manifold, which then must be included in the model. As a result, the diesel engine models include pressures, densities (ρ) and burned gas fraction (F) in both the intake, and exhaust (em) manifolds,

$$\dot{p}_{\rm im} = \frac{c_p R}{c_v V_{\rm im}}(W_{\rm com}T_{\rm com} - W_{\rm cyl}T_{\rm im} + W_{\rm egr}T_{\rm em}), \tag{2a}$$

$$\dot{\rho}_{\rm im} = \frac{1}{V_{\rm im}}(W_{\rm com} - W_{\rm cyl} + W_{\rm egr}), \tag{2b}$$

$$\dot{F}_{\rm im} = \frac{(F_{\rm em} - F_{\rm im})W_{\rm egr} - F_{\rm im}W_{\rm com}}{\rho_{\rm im}V_{\rm im}}, \tag{2c}$$

$$\dot{p}_{\rm em} = \frac{c_p R}{c_v V_{\rm em}}(W_{\rm cyl}T_{\rm cyl} - W_{\rm tur}T_{\rm em} - W_{\rm egr}T_{\rm em} - \dot{Q}_{\rm em}/c_p), \tag{2d}$$

$$\dot{\rho}_{\rm em} = \frac{1}{V_{\rm em}}(W_{\rm cyl} - W_{\rm tur} - W_{\rm egr}), \tag{2e}$$

$$\dot{F}_{\rm em} = \frac{(F_{\rm em} - F_{\rm im})W_{\rm egr}}{\rho_{\rm em}V_{\rm em}}, \tag{2f}$$

where c_p, c_v are the gas specific heat at constant pressure and constant temperature, respectively, $\dot{Q}$ is the heat flow, and the subscripts *egr*, *com*, *tur* refer, respectively, to the exhaust gas being recirculated, the compressor, and the turbine.

Equations in (2a) must be coupled with the equations describing the flows. While the cylinder flow equation is the same as in (1b) for the SI engine model, and the EGR flow is controlled by a valve resulting in an equation similar to (1c), the remaining flows are determined by the turbocharger equations,

$$W_{com} = \frac{p_{\rm amb}}{\sqrt{T_{\rm amb}}}\phi_{\rm com}(N_{tc}/\sqrt{T_{\rm amb}}, p_{\rm im}/p_{\rm amb}), \tag{3a}$$

$$W_{tur} = \frac{p_{\rm em}}{\sqrt{T_{\rm em}}}\phi_{\rm tur}(\chi_{\rm vgt}, p_{\rm ep}/p_{\rm em}), \tag{3b}$$

$$\dot{N}_{tc} = \frac{\gamma_3}{J_{\rm tc}}\frac{\eta_{\rm tur}W_{\rm tur}(T_{\rm em} - T_{\rm ep}) - \eta_{\rm com}W_{\rm com}(T_{\rm im} - T_{\rm amb})}{N_{\rm tc}}, \tag{3c}$$

where *ep* refers to the exhaust pipe, $\chi_{\rm vgt}$ is the variable geometry turbine actuator, $N_{\rm tc}$ and $J_{\rm tc}$ are the speed and inertia of the turbocharger, $\phi_{\rm com}$ and $\phi_{\rm tur}$, $\eta_{\rm com}$ and $\eta_{\rm tur}$, are the flow parameter and efficiency of turbine and compressor.

It is worth noting that in recent years downsized gasoline engines that are turbocharged have become more common. Their airpath model is a hybrid between the SI and CI models, since they have SI combustion, and throttle, but also a turbocharger, although, in general, with a smaller fixed geometry turbine, and possibly a wastegate valve instead of the EGR valve [70].

The second part of the engine model is the torque production model, which describes the net torque output generated by the engine. This model has the form,

$$M_{\mathrm{e}} = M_{\mathrm{ind}}(t - t_{\mathrm{d}}) - M_{\mathrm{fr}}(N) - M_{\mathrm{pmp}}(p_{\mathrm{im}}, p_{\mathrm{em}}, N), \tag{4}$$

where M_{ind}, M_{fr}, M_{pmp} are the indicated, friction, and pumping torques, respectively. The indicated torque is the produced torque and its expression depends on the engine type. For SI engines,

$$M_{\mathrm{ind}} \approx \kappa_{\mathrm{spk}}(t - t_{\mathrm{ds}})\gamma_4 \frac{W_{\mathrm{cyl}}}{N}, \tag{5a}$$

$$\kappa_{\mathrm{spk}} \approx \left(\cos(\alpha - \alpha_{\mathrm{MBT}})\right)^{\gamma_5}, \tag{5b}$$

where α and α_{MBT} are the ignition angle and the maximum brake torque ignition angle, and κ_{spk} is the torque ratio achieved by spark ignition timing. Since CI engines do not use spark timing as an actuator, and the air-to-fuel ratio in these engines may vary over a broad range, the indicated torque equation is usually obtained from engine calibration data, e.g., as

$$M_{\mathrm{ind}} = f_{\mathrm{indCI}}(W_f, N, F_{\mathrm{im}}, \delta) \tag{6}$$

where W_f is the fuel flow, and δ corresponds to the fuel injection parameters (e.g., start of injection).

The final component in the engine models represents the transfer of the torque from the engine to the wheels. In general, the engine speed (in RPM) is related to the engine torque M_e, inertia of the crankshaft and flywheel J_e, and load torque M_L by

$$\dot{N} = \frac{1}{J_e}\frac{30}{\pi}(M_{\mathrm{e}} - M_{\mathrm{L}}). \tag{7}$$

The load torque model varies largely depending on whether the vehicle has an automatic transmission, which includes a torque converter, or a manual transmission with dry clutches. Depending on the compliance of the shafts and actuation of the clutches, the steady state component of the torque load is

$$M_{\mathrm{L}} = \frac{r_w}{g_r} F_{\mathrm{trac}} + M_{\mathrm{los}} + M_{\mathrm{aux}},$$

where M_{los}, M_{aux} are the torque losses in the driveline and because of the auxiliary loads, r_w is the wheel radius and g_r is the total gear ratio between wheels and engine shaft, usually composed of final drive ratio, transmission gear ratio, and, if present, torque converter ratio.

1) MPC opportunities in powertrain control: Powertrain control has likely been the first, and probably the largest, application area of MPC in automotive systems. In conventional SI engines, when the driver is pressing on the gas pedal, the vehicle is in the torque control mode and there are basically no degrees of freedom. Thus, the main opportunities for MPC application are in the so-called closed-pedal operation, i.e., when the gas pedal is released, and the vehicle is in a speed control mode.

An example is idle speed control [29] where the spark timing and the throttle are actuated to keep a target speed despite external disturbances. The engine speed must be kept from becoming too small, otherwise the engine may stall, and the throttle and spark timing are subject to physical and operational constraints, for instance due to knocking or misfiring. Thus, the optimal control problem can be formulated as

$$\min_{\alpha,\vartheta} \quad \sum_{t=0}^{T_N} (N(t)-r_N(t))^2 + w_\vartheta \Delta\vartheta(t)^2 + w_\alpha(\alpha(t)-\alpha_r(t))^2 \tag{8a}$$

$$\text{s.t.} \quad \underline{\alpha}(t) \le \alpha(t) \le \overline{\alpha}(t), \quad \underline{\vartheta}(t) \le \vartheta(t) \le \overline{\vartheta}(t), \quad N(t) \ge \underline{N}(t) \tag{8b}$$

where w_ϑ, w_α are positive tuning weights, and r_N, α_r are references that are constant or slowly varying based on engine temperature.

During the deceleration control, the engine speed is controlled to follow a reference trajectory that causes the vehicle to decelerate smoothly and energy-efficiently, and still allows for the engine to rapidly resume torque production, if acceleration is needed, see Figure 3. In this case the problem is similar to (8), except that the reference speed trajectory is time varying and a first order model for it is often available and may be used for preview.

Both idle speed control and deceleration control are multivariable control problems in which actuators are subject to constraints and the dynamics are affected by delays of different lengths in different control channels. Based on guidelines in Table I, both idle speed control and deceleration control are clearly good application areas for MPC. On the other hand, the dynamics are clearly nonlinear in both of these problems. Since idling takes place near a setpoint, a linearized model for idling is fairly accurate. On the other hand, the deceleration control operates in a constant transient, and hence it is often convenient to develop a low-level controller that linearizes the dynamics. In such a control architecture, MPC can exploit constraints to ensure that the interaction with the low level controller is effective. For deceleration control, a low level controller is tasked with delivering the demanded torque, thus transforming the pressure-based model into a torque-based model, where the torque response is modeled as a first order plus delay

$$\dot{N}(t) = \frac{1}{J_e}(\hat{\kappa}_{\text{spk}} M_{\text{air}}(t) + u_{\text{spk}}(t - t_{ds}) - M_L(t)), \tag{9a}$$

$$\dot{M}_{\text{air}}(t) = \frac{1}{\tau_{\text{air}}}(-M_{\text{air}}(t) + u_{\text{air}}(t - t_d(t)), \tag{9b}$$

$$\underline{M}_{\text{air}}(t) \leq M_{\text{air}}(t) \leq \overline{M}_{\text{air}}(t), \tag{9c}$$

$$\underline{\Delta\kappa} M_{\text{air}}(t) \leq u_{\text{spk}}(t - t_{ds}) \leq \overline{\Delta\kappa}\ M_{\text{air}}(t). \tag{9d}$$

The multiplicative relation between spark timing and torque is converted into an additive one subject to linear constraints by introducing a virtual control representing the torque modification obtained from spark actuation. This is possible with MPC due to the capability of handling constraints.

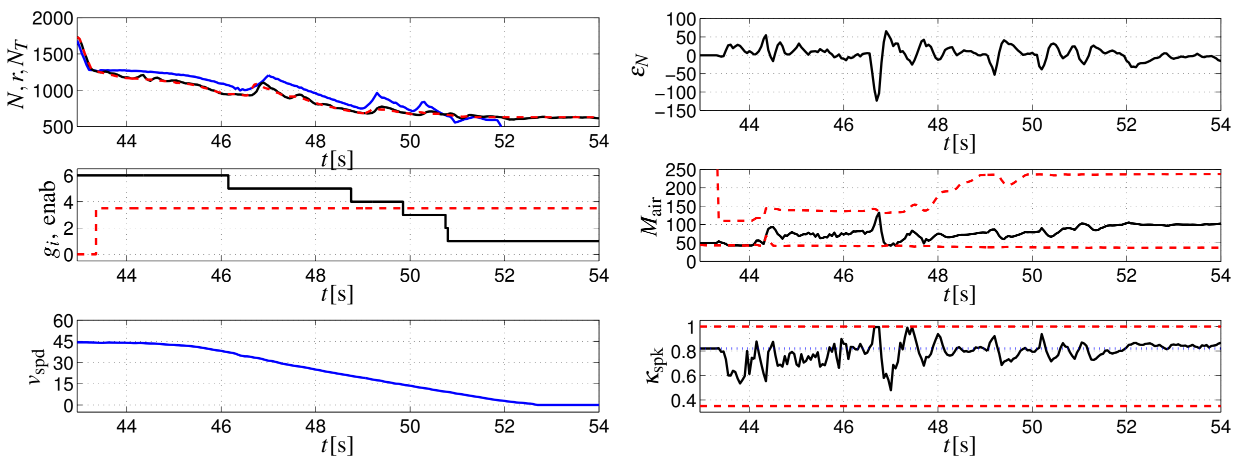

Fig. 3. Experimental test of MPC-based deceleration control from [23]. Engine speed N, reference r, and tracking error ε_N, torque converted turbine speed N_T, gear and controller enabling signal, vehicle speed v_{spd}, torque from airflow M_{air} and torque ratio from spark, κ_{spk} are shown.

CI engines are far more complex and have more degrees of freedom than naturally aspirated gasoline engines, due to EGR, VGT, and multiple fuel injections which must be exploited throughout the entire operating range to achieve a suitable tradeoff between torque delivery and emissions. In general, in diesel engines the fuel flow W_f is determined based on the pedal position and current engine speed, and from that, the setpoints for other variables such as the intake manifold pressure and either mass airflow through the compressor or EGR rate are determined. Then, a feedback controller is developed that actuates the VGT, EGR valve, and possibly intake throttle, to track these setpoints. Also in this case we obtain a multivariable control problem with constraints on actuators and process variables, such as intake and exhaust manifold pressures, EGR rate, turbocharger speed, turbine temperature, compressor surge margin, etc. The MPC solution can be simplified [50] by pursuing a rate-based formulation, constraint remodeling, intermittent constraint enforcement, and by combining with nonlinear static or dynamic inversion.

B. Control of Vehicle Dynamics

Vehicle dynamics models are derived from the planar rigid body equations of motion. For normal driving that involves neither high performance driving nor low speed maneuvers, the single track model, also known as bicycle model, shown in Figure 4, is common. This model is described by

$$m(\dot{v}_x - v_y\dot{\psi}) = F_{xf} + F_{xr}, \tag{10a}$$

$$m(\dot{v}_y + v_x\dot{\psi}) = F_{yf} + F_{yr}, \tag{10b}$$

$$J_z\ddot{\psi} = \ell_f F_{yf} - \ell_j F_{yr}, \tag{10c}$$

where m is the vehicle mass, ψ is the yaw rate, v_x, v_y are the components of the velocity vector in the longitudinal and lateral vehicle direction, J_z is the moment of inertia about the vertical axis, ℓ_f, ℓ_r are the distances of front and rear axles from the center of mass. In (10), F_{ij}, $i \in \{x,y\}$, $j \in \{f,r\}$ are the longitudinal and lateral, front and rear tire forces expressed in the vehicle frame [65],

$$F_{xj} = f_l(\alpha_j, \delta_j, \sigma_j, \mu, F_{zj}), \quad F_{yj} = f_c(\alpha_j, \delta_j, \sigma_j, \mu, F_{zj}), \quad F_{zj} = \frac{\ell_j}{\ell_f + \ell_r} mg, \tag{11}$$

where δ_j is the steering angle at the tires, α_j is the tire slip angle and σ_j is the slip ratio, for front and rear tires $j \in \{f,r\}$, and μ is the friction coefficient between tires and road. The slip angles and the slip ratios relate the vehicle tractive forces with the vehicle velocity and the wheel speeds, thereby coupling the vehicle response with the powertrain response,

$$\alpha_j = \tan^{-1}\left(\frac{v_{y_j}}{v_{xj}}\right), \tag{12a}$$

$$v_{lj} = v_{y_j}\sin\delta_j + v_{xj}\cos\delta_j, \quad v_{cj} = v_{y_j}\cos\delta_j - v_{xj}\sin\delta_j, \tag{12b}$$

$$\sigma_j = \begin{cases} \frac{r\omega_j}{v_{xj}} - 1 & \text{if} \quad v_{xj} > r\omega_j, \\ 1 - \frac{r\omega_j}{v_{xj}} & \text{if} \quad v_{xj} < r\omega_j, \end{cases} \tag{12c}$$

where v_{xj}, v_{y_j}, $j \in \{f,r\}$, are the longitudinal and lateral components of the vehicle velocity vector at the tires. In, (11), the functions f_l, f_c define the tire forces that are in general determined by data or according to a model such as Pacejka's or Lu'Gre [52].

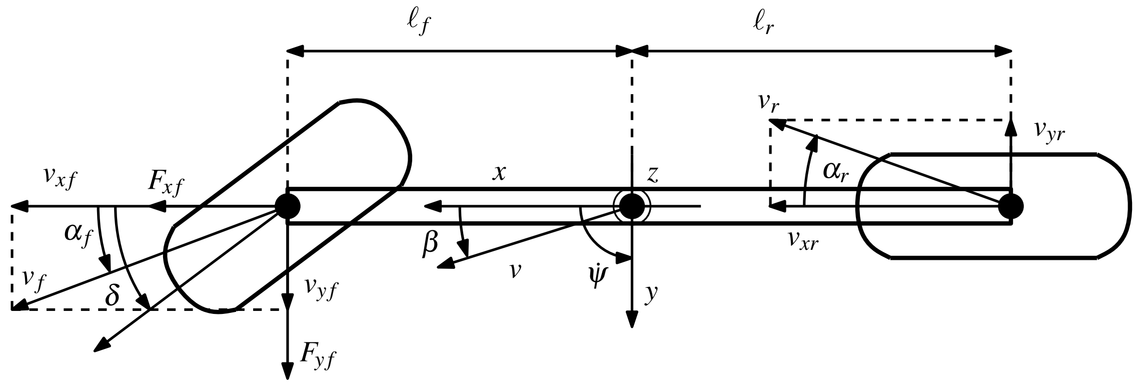

Fig. 4. Schematics of the single track model for the lateral vehicle dynamics. The most relevant vectors for describing the model are also shown.

In the above mentioned normal driving conditions, the longitudinal and lateral dynamics are often decoupled, yielding a lateral dynamics model where v_x is constant, and, with a further linear approximation of the lateral tire forces as a function of the slip angles, resulting in

$$m\dot{v}_y = -\frac{C_f+C_r}{v_x}v_y - \left(v_x - \frac{C_f\ell_f - C_r\ell_r}{v_x}\right)\dot{\psi} + C_f\delta, \tag{13a}$$

$$J_z\ddot{\psi} = -\frac{C_f\ell_f - C_r\ell_r}{v_x}v_y - \frac{C_f\ell_f^2 + C_r\ell_r^2}{v_x}\dot{\psi} + \ell_f C_f\delta + M_{br}, \tag{13b}$$

where we used the relation $\alpha_f = (v_y + \ell_f\dot{\psi})/v_x$, $\alpha_r = (v_y - \ell_r\dot{\psi})/v_x$, and we have included a moment $M_{\rm br}$ that can be generated by applying non-uniform forces at different wheels, for instance by differential braking. In (13), C_f, C_r are the front and rear lateral tire stiffnesses, which correspond to a linear approximation of the lateral tire forces as functions of the slip angles, $F_{yj} = C_j\alpha_j$.

Similarly, the longitudinal dynamics are also simplified by neglecting the lateral dynamics, resulting in

$$m\dot{v}_x = \sum_{j\in f,r} f_l(0,s_j,\mu,F_{zj}) - F_{\rm res} \approx C_f^x\sigma_f + C_r^x\sigma_r - F_{\rm res}, \tag{14a}$$

$$F_{\rm res} = F_{\rm aero} + F_{\rm roll} + F_{\rm grade} \approx \frac{1}{2}\rho_{\rm air}A_f c_d v_x^2 + mgc_r\cos\theta_{\rm rd} + mg\sin\theta_{\rm rd}, \tag{14b}$$

where C_f, C_r are the front and rear longitudinal tire stiffnesses, that represent a linear approximation of the longitudinal tire forces as functions of the slip ratio $F_{xj} = C_j^x\sigma_j$. The slip ratio changes based on the torques exerted on the wheels by the engine and the brakes, thus relating the powertrain and braking system actuation with the vehicle motion. In (14) we have included the effects of resistance forces due to airdrag, rolling, and road grade. Here ρ_{air} is the density of air, A_f is the vehicle frontal area, C_d is the drag coefficient, θ_{rd} is the road grade, c_r is the rolling resistance coefficient, and g is the gravity acceleration. The longitudinal vehicle dynamics can be linked to the powertrain torque production in several ways. For low bandwidth applications, such as cruise control, one can approximate $C_f^x\sigma_f + C_r^x\sigma_r \approx F_{\text{trac}}$, where the driveline shafts are assumed to be rigid. The tractive force F_{trac} is the response of a first order-plus-delay system, representing the force at the wheels applied from the powertrain side,

$$\dot{F}_{\text{trac}} = -\frac{1}{\tau_F}F_{\text{trac}} + \frac{1}{\tau_F}u_F(t-t_F).$$

If shaft compliance is considered, the tractive torque $M_{trac} = F_{trac}/r_w$ is caused by the slip between the wheel half-shafts and the rigid transmission shaft, so that

$$M_{trac} = k_s(\theta_e - \theta_w g_r) + d_s(\dot{\theta}_e - \dot{\theta}_w g_r), \tag{15}$$

where k_s and d_s are the half-shafts stiffness and damping, θ_e, θ_w are the engine and wheel shaft angles, and g_r is the total gear ratio between engine and wheels.

The active control of the vertical vehicle dynamics is mainly obtained by active and semi-active suspensions. The simplest model is the quarter-car model [45], where each suspension is independent from the others. The standard quarter-car model describes the vertical vehicle dynamics as two masses, the unsprung mass M_{us} representing the car wheel, with stiffness k_{us} and damping d_{us}, and the sprung mass M_s, representing one quarter of the car body, connected by a spring-damper, k_s, d_s, the passive component of the suspension, and with a force F_a acting between them. Such force is generated by the active or semi-active suspension actuator.

The equations of motions for the sprung and unsprung mass are

$$M_s\ddot{x}_s = c_s(\dot{x}_{us} - \dot{x}_s) + k_s(x_{us} - x_s) - F_a, \tag{16a}$$

$$M_{us}\ddot{x}_{us} = c_t(\dot{r} - \dot{x}_{us}) + k_s(r - x_{us}) + c_s(\dot{x}_s - \dot{x}_{us}) + k_s(x_s - x_{us}) + F_a, \tag{16b}$$

where x_s is the position of the sprung mass, x_{us} is the position of the unsprung mass, F_a is the actuator force, and r is the local road height with respect to the average.

The objective of the suspension control is to limit tire deflections, hence ensuring that the vehicle maintains good handling, to limit suspension deflections, hence ensuring that the suspension does not run against its hard stops causing noise, vibrations and harshness (NVH) and wear, and to limit sprung mass accelerations, hence resulting in a comfortable ride. The type of actuator, e.g., hydraulic, electromagnetic, etc., and its overall capabilities, e.g., active or semi-active, may require additional models for the actuator dynamics, and possibly constraints limiting its action, such as force ranges or passivity constraints.

1) MPC opportunities in vehicle dynamics: MPC of longitudinal vehicle dynamics has been applied for adaptive cruise control (ACC), see, e.g., [59]. The objective of adaptive cruise control is to track a vehicle reference speed r while ensuring a separation distance d from the preceding vehicle, related to the head-away time T_h, and comfortable ride, all of which can be formulated as

$$\min_{F_{\text{trac}}} \quad \sum_{t=0}^{T_N} (v_x(t) - r_v(t))^2 + w_F \Delta u_F(t)^2 \tag{17a}$$

$$\text{s.t.} \quad \underline{F}_{\text{trac}} \leq F_{\text{trac}}(t) \leq \overline{F}_{\text{trac}}, \tag{17b}$$

$$d(t) \geq T_h v_x(t), \tag{17c}$$

where w_F is a positive tuning weight. For ACC, interesting opportunities are opened when a stochastic description of the velocity of the traffic ahead is available or can be estimated [11], or, in the context of V2V and V2I, when there is perfect preview through communication [72]. Also, using an economic cost can help reduce fuel consumption, with minimal impact on travel time [63]. Additional potential applications in longitudinal vehicle dynamics still to be investigated in depth are launch control and gear shifting. More recent applications involve braking control for collision avoidance systems, see, e.g., [57] possibly by using again V2X to exploit preview information.

The interest on MPC for lateral dynamics spans multiple applications, especially lateral stability control and lane keeping, up to autonomous driving. A challenging case [27] is the coordination of differential braking moment and steering to enforce cornering, i.e., yaw rate reference r_ψ tracking, and vehicle stability, i.e., avoiding that the slip angles become so large that the vehicle spins out of control. Such a problem is challenging due to its constrained multivariable nature and to the need to consider nonlinear tire models. A viable approach is to consider piecewise linear tire models, resulting in the optimal control problem

$$\min_{\Delta\delta, M_{\text{br}}} \quad \sum_{t=0}^{T_N} (\dot{\psi}(t) - r_\psi(t))^2 + w_\delta \Delta\delta(t)^2 + w_{br} M_{\text{br}}(t)^2 \tag{18a}$$

$$\text{s.t.} \quad |M_{\text{br}}(t)| \leq \overline{M}_{\text{br}}, \quad |\delta(t) - \delta(t-1)| \leq \overline{\Delta\delta}, \quad |\delta(t)| \leq \overline{\delta}, \tag{18b}$$

$$f_{cj}(\alpha_j) = \begin{cases} -d_j\alpha_j + e_j & \text{if} \quad \alpha_j > p_j, \\ C_j\alpha_j & \text{if} \quad |\alpha_j| \leq p_j, \\ d_j\alpha_j - e_j & \text{if} \quad \alpha_j < -p_j, \end{cases} \tag{18c}$$

$$|\alpha_j| \leq \overline{\alpha}_j, \tag{18d}$$

where w_δ, w_{br} are positive tuning weights, and then using either a hybrid MPC or a switched MPC, where the current linear model is applied for prediction during the entire horizon.

As for the vertical dynamics, MPC offers interesting possibilities for active suspension control when preview of the road is available [30], [58], for instance obtained from a forward looking camera. MPC may also be beneficial in semi-active suspension control, since the passivity condition

$$F_a(\dot{x}_s - \dot{x}_{us}) \geq 0, \tag{19}$$

can be enforced in MPC as a constraint, which ensures that the only commands that are realizable by a semi-active actuator are actually issued. Constraint (19) is nonlinear, but can be enforced by mixed-logical constraints

$$[\delta_v = 1] \quad \leftrightarrow \quad [\dot{x}_s - \dot{x}_{us} \geq 0], \tag{20a}$$

$$[\delta_F = 1] \quad \leftrightarrow \quad [F_a \geq 0], \tag{20b}$$

$$[\delta_v = 1] \quad \leftrightarrow \quad [\delta_F = 1], \tag{20c}$$

where δ_v, δ_F are auxiliary integer variables, thus resulting in a hybrid MPC [36].

Finally, for more advanced systems that aim at coordinating all the suspensions in the vehicle [39], the multivariable nature of MPC can be effective.

C. Energy Management in Hybrid Vehicles

The novel element in hybrid powertrains is the presence of multiple power generation devices, e.g., engine, motor, generator, and energy storage devices, e.g., fuel tank, battery, flywheel. The most common hybrid vehicles are hybrid electric vehicles (HEV) where the internal combustion engine is augmented with electric motors and generators, and batteries for energy storage. For HEV there are a number of possible component topologies that determine the configurations of the power coupling, the most common being series, parallel, powersplit, and electric rear axle drive (ERAD).

The presence of multiple power generation devices requires modeling the power balance. A general model for the mechanical power balance that ultimately describes the amount of power delivered to the wheels is

$$P_{\text{veh}} = P_{\text{eng}} - P_{\text{gen}} + P_{\text{mot}} - P_{\text{mec}}^{\text{los}}, \tag{21}$$

where P_{veh} is the vehicle power for traction, P_{eng} is the engine power, P_{gen} is the mechanical power used to generate electrical energy to be stored in the battery, P_{mot} is the electrical power used for traction and $P_{\text{mec}}^{\text{los}}$ are the mechanical power losses. Note that in advanced HEV architectures, such as

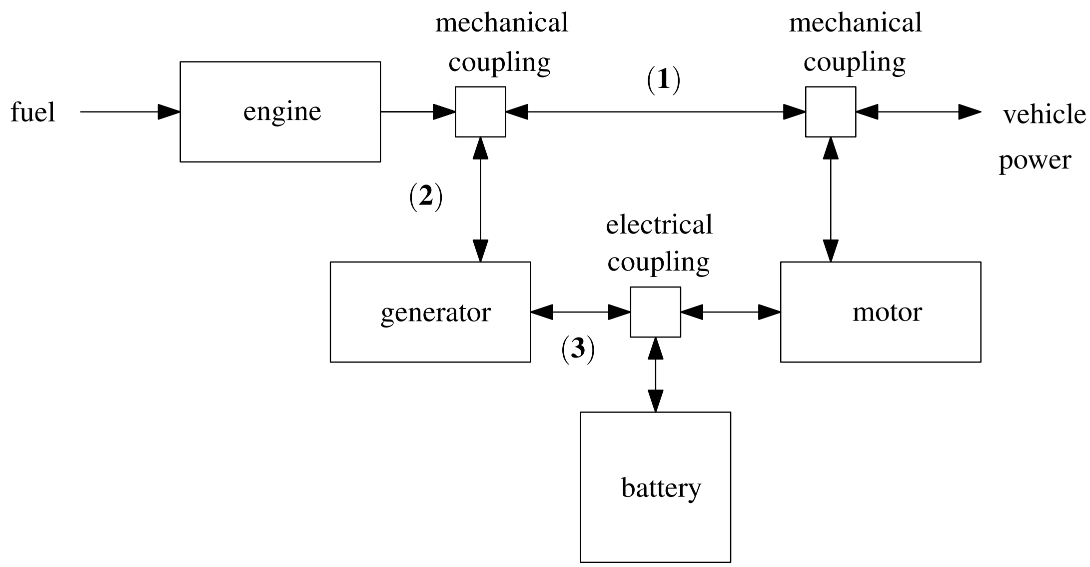

Fig. 5. Schematic of a powersplit HEV architecture. The arrows indicate the allowed power flow directions. The series HEV architecture is obtained by removing the link (1), thus the mechanical couplings are simply mechanical connections. The parallel HEV architecture is obtained by removing the generator and hence links (2) and (3).

the powersplit architecture in Figure 5, motoring and electric energy generation can be accomplished by multiple components, since, despite the names that indicate their preferred usage, both the motor and the generator can convert mechanical energy into electrical energy, and the other way around.

The electrical power balance, that is used to determine the power delivered to/from the battery, is often modeled as

$$P_{\text{bat}} = P_{\text{mot}} - P_{\text{gen}} + P_{\text{gen}}^{\text{los}} + P_{\text{mot}}^{\text{los}}, \tag{22}$$

where P_{bat} is the power flowing from the battery and $P_{\text{gen}}^{\text{los}}$, $P_{\text{mot}}^{\text{los}}$ are the losses in electric energy generation and in the electric motoring, respectively.

Since, as opposed to the fuel tank, the battery power flow is bi-directional and the stored energy is usually quite limited, the energy stored in the battery should be tracked and it is in fact the main state of the HEV energy model. The energy stored in the battery is related to the stored charge, which is normalized with respect to the maximum to obtain the battery state of charge (SoC) $SoC = \frac{Q_{\text{bat}}}{Q_{\text{max}}}$. The battery power, voltage, and current are related by

$$P_{\text{bat}} = (V_{\text{bat}}^{\text{oc}} - I_{\text{bat}} R_{\text{bat}}) I_{\text{bat}},$$

where $V_{\text{bat}}^{\text{oc}}$ is the open circuit battery voltage, I_{bat} is the battery current and R_{bat} is the battery internal resistance. This results in the state of charge dynamics

$$\dot{SoC} = -\frac{V_{\text{bat}}^{\text{oc}} - \sqrt{V_{\text{bat}}^{\text{oc}\,2} - 4R_{\text{bat}}P_{\text{bat}}}}{2R_{\text{bat}}Q_{\max}}.$$

Considering a power coupling that is kept under voltage control and representing the battery as a large capacitor, i.e., ignoring internal resistance, we obtain a simpler representation

$$\dot{SoC} = -\eta_{\text{bat}}(P_{\text{bat}}, SoC)\frac{P_{\text{bat}}}{V_{\text{bat}}^{\text{cc}}Q_{\max}},$$

where η_{bat} is the battery efficiency, that, for feedback control, can be well approximated by one or two constants [25], the latter case modeling different efficiencies in charging and discharging.

The main novel control problem in HEV powertrains is the management of the energy in order to minimize the consumed fuel subject to charge sustaining constraints

$$\min \quad \int_{t_i}^{t_f} W_f(t)dt \tag{23a}$$

$$\text{s.t.} \quad SoC(t_i) = SoC(t_f), \tag{23b}$$

where the fuel flow W_f is related to the engine power by a function that depends on the engine operation, $W_f = f_f(P_{\text{eng}}, N)$. As opposed to conventional powertrains, in most HEV configurations, even for a given engine power and wheel speed, there are degrees of freedom in selecting the engine operating point, i.e., engine speed and engine torque, that can be leveraged by the energy management strategy.

1) MPC opportunities in hybrid vehicles: Due to the focus on optimizing the energy consumption, subject to constraints on power flows and battery state of charge, HEV energy management has been a clear target for MPC application.

The key idea is to construct a finite horizon approximation of the fuel consumption cost function (23a), augmented with a term penalizing large differences of SoC at the end of the horizon, which can be interpreted as the augmented Lagrangian form of the charge sustaining constraint (23b). The cost function can also include additional terms such as SoC reference tracking. Furthermore, constraints on the various power flows and battery SoC can also be included, resulting in

$$\min_{P_{\text{bat}},\dots} \quad F_N(SoC(T_N)) + \sum_{t=0}^{T_N-1} W_f(t) + w_{soc}(SoC(t) - r_{SoC}(t))^2 \tag{24a}$$

$$\text{s.t.} \quad P_{\text{eng}} - P_{\text{gen}} + P_{\text{mot}} - P_{\text{mec}}^{\text{los}} = P_{\text{drv}}, \tag{24b}$$

$$\underline{SoC} \leq SoC(t) \leq \overline{SoC}, \tag{24c}$$

$$|P_{\text{bat}}| \leq \overline{P}_{\text{bat}}. \tag{24d}$$

The mechanical power equation (21) is enforced as a constraint in (24) to ensure that the vehicle power is equal to the driver-requested power P_{drv}. The actual number of degrees of freedom in (24) varies with the HEV architecture. For a powersplit architecture, shown in Figure 5, in its entirety, it is equal to two. For a parallel architecture, where there is no generator, and for a series architecture, where there is no pure mechanical connection between engine and wheels, it is equal to one. Exploiting the simplicity of the latter, an MPC was developed in [25] which was deployed on a prototype production series HEV that was fully road drivable.

It is interesting to note that the cost function in HEV energy management is of economic type and in retrospect, HEV energy management was probably the first real-world application of economic MPC, showing in fact the possibilty of steady state limit cycles or offsets [25]. In recent years multiple advanced MPC methods have been applied to HEV energy management, including stochastic MPC in [69] where the driver-requested power is predicted using statistical models, possibly learned from data during vehicle operation.

D. Other applications

Given that the field of automotive control is large, it is it is impossible to provide a comprehensive account for all automotive applications of MPC in a single chapter. In this chapter we focused on the three areas described above, which have been very actively researched over the last few years. However, there are several other applications that could be noted, including, among others, emission control in SI, e.g., [71], [74], and CI engines, e.g., [47], [48] engines, transmission control, e.g., [2], [7], [40], [76], control of gasoline turbocharged engines, e.g., [1], [70], control of homogeneous combustion compression ignition (HCCI) engines, e.g., [10], [66], and energy management of fuel-cell vehicles, e.g., [3], [4], [75].

III. MPC Design Process in Automotive Applications

This section aims at providing guidelines for a design process for MPC in automotive applications. While not a standard, it has been applied by the authors in multiple designs that were eventually tested in vehicles, and it has been proved useful and effective in those applications. We focus on linear MPC, because, as discussed later, this has been so far the main method used in automotive applications, primarily due to computational and implementation requirements. However, the design process extends almost directly to nonlinear MPC.

The MPC design amounts to properly constructing components of the finite horizon optimal control problem to achieve the desired specifications and control-oriented properties. We consider the finite horizon optimal control problem

$$\min_{U(t)} \quad x'_{t+N|t}Px_{t+N|t} + \sum_{k=0}^{N-1} z'_{t+k|t}Qz_{t+k|t} + u'_{t+k|t}Ru_{t+k|t} \tag{25a}$$

$$x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t}, \tag{25b}$$

$$y_{t+k|t} = Cx_{t+k|t} + Du_{t+k|t}, \tag{25c}$$

$$z_{t+k|t} = Ex_{t+k|t}, \tag{25d}$$

$$u_{t+k|t} = \kappa_f x_{t+k|t}, \quad k = N_u, \ldots, N-1, \tag{25e}$$

$$x_{t|t} = x(t), \tag{25f}$$

$$\underline{y} \leq y_{t+k|t} \leq \overline{y}, \quad k = N_i, \ldots N_{cy}, \tag{25g}$$

$$\underline{u} \leq u_{t+k|t} \leq \overline{u}, \quad k = 0, \ldots N_{cu}-1, \tag{25h}$$

$$H_N x_{t+N|t} \leq K_N, \tag{25i}$$

where the notation $t+k|t$ denotes the k-step prediction from measurements at time t, $U(t) = \{u_t \ldots u_{t+N-1}\}$ in the control input sequence to be optimized, x, u, y, z are the prediction model state, input, constrained outputs, and performance output vectors, $\underline{u}, \overline{u}, \underline{y}, \overline{y}$ are lower and upper bounds on input and constrained output vectors, P, Q, R are weighting matrices, N, N_{cu}, N_{cy}, N_u are non-negative integers defining the horizons, κ_f is the terminal controller, and H_N, K_N define the terminal set. Next, we discuss the role of each of these components in achieving the specifications that are common on automotive applications.

A. Prediction Model

Several dynamical processes occurring in automotive applications are well studied and have readily available physics-based models, some of which have been described in Sections II-A–II-C. In MPC design it is desirable to start from such physics-based models. However, many of them may be of unnecessarily high order, may be nonlinear, and may have several parameters to be estimated for different vehicles. Hence, usually the first step in MPC design is to refine the physics based model by:

- simplifying the model to capture the relevant dynamics based on the specific application and controller requirements, e.g., the sampling period, by linearization, model order reduction, etc;
- estimating the unknown parameters by gray-box system identification methods, e.g., linear/nonlinear regression, step response analysis, etc.;
- time-discretizing the dynamics to obtain in a discrete-time prediction model.

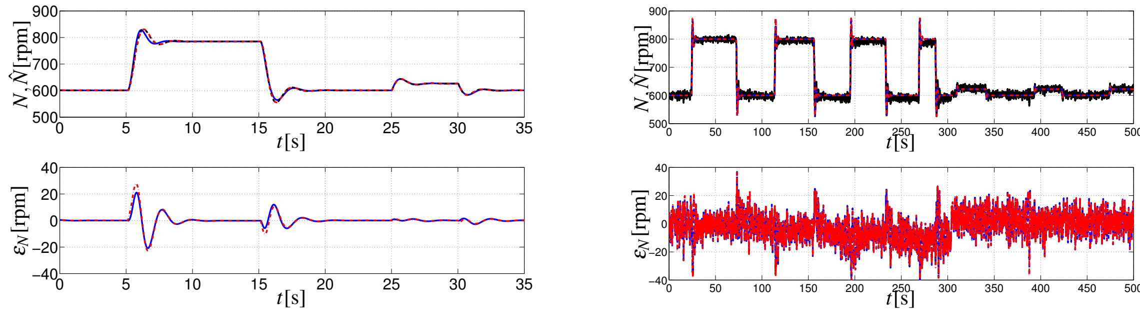

Fig. 6. Validation of identification of engine model for idle speed control from nonlinear model data (left) and experimental data (right) in throttle and spark up-down steps from [29]. Upper plot: data (solid), continuous-time linear model (dash), discrete-time linear model (dash-dot). Lower plot: continuous-time model error (solid) discrete-time model error (dash-dot).

Even for the relatively simple case of idle speed control, in [29] due to computational requirements, the powertrain model (1a), (4) is linearized around the nominal idle operating point. The model structure is known from physics, and it consists of two transfer functions, from throttle and spark to engine speed, each of second order and subject to delays, and the the latter has an additional stable zero. The model parameters are identified from the step responses, with the models for the delays removed during identification, to be added again later, see Figure 6.

The result of the first model construction step is usually a linear, discrete-time constrained model for the physical process,

$$x_m(t+1) = A_m x_m(t) + B_m u_m(t), \tag{26a}$$

$$y_m(t) = C_m x_m(t) + D_m u_m(t), \tag{26b}$$

$$z_m(t) = E_m x_m(t), \tag{26c}$$

where $x_m \in \mathbb{R}^{n_m}$ is the state vector, $u_m \in \mathbb{R}^{m_m}$ is the input vector $y_m \in \mathbb{R}^{p_m}$ is the constrained output vector, and $z_m \in \mathbb{R}^{q_m}$ is the performance output vector.

The process model (26) usually needs to be augmented with additional states and artificial dynamics in order to achieve the problem specifications, such as tracking of references, non-zero, and possibly unknown steady state input values, rejection of certain classes of disturbances, e.g., constant, sinusoidal, etc. Further modifications may be made to account for additional information available in the system, such as preview on disturbances or references, or known models for those. Typical augmentations are the incremental input formulation

$$u(t+1) = u(t) + \Delta u(t),$$

the inclusion of integral action to track constant references and reject constant unmeasured disturbances,

$$\iota(t+1) = \iota(t) + T_s C_\iota z(t),$$

and the inclusion of disturbance models

$$\begin{aligned} \eta(t+1) &= A\eta(t), \\ d(t) &= C_d \eta(t), \end{aligned}$$

where the disturbance model state η is measured in the case of measured disturbances, while in the case of unmeasured disturbance it is estimated from disturbance observers. A case of particular interest is the inclusion of buffers, which allow to account for time delays and preview on disturbances and references,

$$\begin{aligned} \xi(t+1) &= \begin{bmatrix} 0 & I \\ 0 & c \end{bmatrix} \xi(t), \\ \chi(t) &= \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \xi(t), \end{aligned}$$

where c is usually either 1 or 0 depending on whether, after the delay window, the last value in the buffer is to be held constant or set to 0. Note that an exact linear model of the delay buffer can be formulated in discrete-time, albeit with the resolution of the sampling period, while in continuous-time one must resort to Padé approximations, that may introduce fictitious non-minimum phase behaviors and mislead the control decisions.

Due to its intrinsic feedforward-plus-feedback nature, MPC is often applied for reference tracking. However, the application of MPC to these problems is not as simple as for linear controllers, because the constraints usually prevent from simply "shifting the origin" to translate the tracking problem into a regulation problem. In automotive applications, it is also difficult, in general, to compute the equilibrium associated with a certain reference value r, due to the uncertainties in the model and the unmeasured disturbances. If one wants to avoid adding a disturbance observer, it may be effective to apply the velocity (or rate-based) form of the model, where both the state and input are differentiated, and the tracking

Specification	Model Augmentation
Piecewise constant reference or measured disturbance	Incremental input
Measured non-predictable disturbance	Constant disturbance model
Previewed reference/disturbance	Preview reference/disturbance buffer
Known time delay	Delay buffer
Unmeasured constant disturbance	Output/tracking error integral action Output disturbance and observer
Reference tracking	Reference model and tracking error, velocity form

TABLE II

LIST OF COMMON SPECIFICATIONS AND RELATED AUGMENTATIONS TO THE PROCESS MODEL TO HANDLE THEM.

error e_m is included as an additional state

$$\Delta x_m(t+1) = A_m \Delta x_m(t) + B_m \Delta u_m(t), \tag{27a}$$

$$e_m(t) = e_m + E_m \Delta x_m(t) + \Delta r(t), \tag{27b}$$

$$y_m(t) = y_m(t-1) + C_m \Delta x_m(t) + D_m \Delta u_m(t), \tag{27c}$$

where Δr is the change in reference value. For MPC applications one may need to add integrators to reformulate y_m in terms of the state and input changes, Δx_m, Δu_m, except for the cases where the constraints are originally in differential form.

The more common augmentations in relations to specifications usually found in automotive control applications are summarized in Table II. Applying all the augmentations results in a higher order prediction model,

$$x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t}, \quad x = \begin{bmatrix} x_m \\ x_p \end{bmatrix} \in \mathbb{R}^n, \; u = \begin{bmatrix} u_m \\ u_p \end{bmatrix} \in \mathbb{R}^m, \tag{28a}$$

$$y_{t+k|t} = Cx_{t+k|t} + Du_{t+k|t}, \quad y = \begin{bmatrix} y_m \\ y_p \end{bmatrix} \in \mathbb{R}^p, \tag{28b}$$

$$z_{t+k|t} = Ex_{t+k|t}, \qquad z = \begin{bmatrix} z_m \\ z_p \end{bmatrix} \in \mathbb{R}^q, \tag{28c}$$

where x, u, y, z are the prediction model state, input, constrained outputs, and performance output vectors, x_p, u_p, y_p, z_p are the augmented state, input, constrained outputs, and performance output vectors.

B. Horizon and Constraints

The constraints are usually enforced on the constrained output vector, and on the input. While enforcing the constraints directly on the states is certainly possible, it is more convenient to introduce the vector y specifically for this use, which allows to enforce state, mixed state-inputs, and, possibly even pure input constraints, through a single vector. Thus, in general the constraints are formulated as

$$y_{t+k|t} \in \mathscr{Y}_m, \quad u_{t+k|t} \in \mathscr{U}_m, \tag{29}$$

where and $\mathscr{Y}_m$ and $\mathscr{U}_m$ are the admissible sets for constrained output and input vectors, respectively. Enforcing constraints on the prediction model y and u, which include augmentation, usually allows to formulate (29) as simple bounds

$$\underline{y} \leq y_{t+k|t} \leq \overline{y}, \quad \underline{u} \leq u_{t+k|t} \leq \overline{u}, \tag{30}$$

which are easier to specify and to handle in an optimization problem.

In automotive applications, the sampling period T_s is often equal to the period of the control cycle for the function being developed. However, for the prediction model to be accurate, it is expected that the sampling period T_s is small enough to allow for 3-10 steps in the settling of the fastest dynamics, following a reference step change. If this is not the case, upsampling or downsampling may be advised, where the prediction model sampling period and the control loop period are different, and appropriate strategies, such as move blocking or interpolation, are applied to bridge such differences.

The choice of the prediction horizon N is related to the prediction model dynamics. In general, N should be slightly larger, e.g., $1.5\times$-$3\times$, than the number of steps for the settling of the slowest (stable) prediction model dynamics, following a reference step change. This requirement relates to the choice of the sampling period so that the total amount of prediction steps is expected to be 5-30 times the ratio between the slowest and the fastest (stable) system dynamics. To be more correct, since the controller usually alters the response of the open-loop system, the relevant settling time for the choice of the prediction horizon is that of the closed-loop system, which usually leads to an iterative selection procedure.

For adjusting the computational requirements in solving the MPC problem, other horizons can be defined. The control horizon N_u determines the number of control steps left as free decision variables to the controller, where for $k \geq N_u$ the input is not an optimization variable but rather assigned by a pre-defined terminal controller,

$$u_{t+k|t} = \kappa_f x_{t+k|t}, \qquad k = N_u, \ldots, N-1. \tag{31}$$

The constraint horizons, for outputs and inputs, N_{cy}, N_{cu}, respectively, determine for how many steps the constraints are enforced,

$$\underline{y} \leq y_{t+k|t} \leq \overline{y}, \quad k = N_i, \ldots N_{cy}, \qquad \underline{u} \leq u_{t+k|t} \leq \overline{u}, \quad k = 0, \ldots N_{cu}-1, \tag{32}$$

where $N_i = \{0,1\}$ depending on whether output constraints are enforced or not at the initial step, which is reasonable only if the input directly affects the constrained outputs. By choosing N_u, one determines the number of optimization variables, $n_v = N_u m$ and by choosing N_{cy}, N_{cu}, one determines the number of constraints, $n_c = 2(p(N_{cy}+N_i)+m(N_{cu}))$. This determines the size of the optimization problem.

C. Cost Function, Terminal Set and Soft Constraints

The cost function encodes the objectives of MPC, and their priority. The control specifications are formulated as variables that are to be controlled to 0. The variables are either a part of the process model (26), or are included in the prediction model (28) by the augmentations discussed in Section III-A. In general the MPC cost function is

$$J_t = x'_{t+N|t} P x_{t+N|t} + \sum_{k=0}^{N-1} z'_{t+k|t} Q z_{t+k|t} + u'_{t+k|t} R u_{t+k|t}, \tag{33}$$

where the performance outputs of the prediction model $z = Ex$ can model objectives such as tracking, e.g., by $z = Cx - C_r x_r$, where x_r is the reference model state, and $r = C_r x_r$ is the current reference. In (33), $Q \geq 0$, $R > 0$ are the matrix weights that determine the importance of the different objectives: for a diagonal weight matrix Q, the larger the i^{th} diagonal component, the faster the i^{th} performance output

Specification	Corresponding weights
Regulation/Tracking error	weight on plant output error
Energy	weight on plant input
Noise, vibration, and harshness (NVH)	weight on plant output acceleration, and plant input rate of change
Consistency	weight on plant output velocity, and plant input rate of change
Comfort	weight on plant output acceleration and jerk, and model input rate of change

TABLE III

LIST OF "DOMAIN TERMS" SPECIFICATIONS AND WEIGHTS THAT OFTEN AFFECT THEM.

will be regulated to 0. It is important to remember that weights determine relative priorities between objectives. Hence, increasing the j^{th} performance output weight may slow down the regulation of the i^{th} performance output.

Very often, the control specifications are given in terms of "domain quantities", such as comfort, NVH (noise, vibration, harshness), consistency, i.e., repeatability of the behavior, and it is not immediately clear how to map them to corresponding weights. While the mappings tend to be application dependent, some of the common mappings are reported in Table III, where we stress that the outputs and inputs, and their derivatives, refer to the plant outputs, which may be part of the performance outputs or inputs, depending on the performed model augmentations.

In (33), $P \geq 0$ is the terminal cost, which is normally used to guarantee at least local stability. There are multiple ways to design P. The most straightforward and more commonly used in automotive applications is to choose P to be the solution of the Riccati equation, constructed from A, B in the prediction model (28), and Q, R in the cost function (33),

$$P = A'PA + Q - A'PB(B'PB + R)^{-1}B'PA.$$

This method can be used, after some modifications, also for output tracking [26]. Alternative approaches are based on choosing P to be the solution of a Lyapunov equation, for systems that are asymptotically stable, or on choosing P as the closed-loop matrix of the system stabilized by a controller, which, for linear plants and controllers, can be computed via LMIs. The terminal controller $u = \kappa_f x$ used after the end of the control horizon is then chosen accordingly, being either the LQR controller, constantly 0, or the stabilizing controller, respectively.

The use of terminal set constraint $H_N x_{t+N|t} \leq K_N$ to guarantee recursive feasibility and stability in the feasible domain of the MPC optimization problem has seen a fairly limited use in automotive applications, also due to the many disturbances and modeling errors acting on the prediction model, which may cause infeasibility of such constraint, due to the need to keep the horizon short because of computational requirements. In practice, for ensuring that the optimization problem admits a solution, constraint softening is often applied. A quadratic program (QP) with soft constraints can be formulated as

$$\min_{v,s} \quad \frac{1}{2}v'Hv + \rho s^2 \tag{34a}$$

$$\text{s.t.} \quad Hv \leq K + Ms, \tag{34b}$$

where s is the slack variable for softening the constraints, M is a vector of 0 and 1 that determines which constraints are actually softened, and ρ is the soft penalty, where usually $\rho I \gg Q$. In general, only output constraints are softened, because input constraints should always be feasible for a well-formulated problem. More advanced formulations with multiple slack variables giving different priorities to different constraints are also possible, as well as different weighting functions for the constraint violation, such as using the absolute value of s.

IV. Computations and Numerical Algorithms

As mentioned in Section I, a key challenge for implementing MPC in automotive applications is accommodating its significantly larger computational footprint when compared to standard automotive controllers, i.e., PID. As MPC is based on solving a finite time optimal control problem, the MPC code is significantly more complex, and it may involve iterations, checking of termination conditions, and sub-routines, as opposed to the integral and derivative updates and a “one-shot” computation of the three terms in the PID feedback law.

The embedded software engineers that are ultimately responsible for controller deployment need to face this additional complexity, and need to move from a controller that evaluates a function, to a controller that executes an algorithm. For the technology transfer of MPC from research to production to have some chances of success, one must, at least initially, reduce such a gap as much as possible by proposing

	Clock	Instructions	Instr./s	RAM	ROM
dCPU	1000s MHz	CISC	100s GIPS	10s GB	1000s GB
aMCU	100s MHz	RISC	1000s MIPS	1000s kB	10s MB

TABLE IV

A COMPARISON OF CURRENT CHARACTERISTIC RANGES FOR DESKTOP PROCESSORS (DCPU) AND AUTOMOTIVE MICRO-CONTROLLERS (AMCU).

simple MPC implementations, and then gradually move towards more advanced implementations when confidence in MPC builds up.

Furthermore, cost is a key driver in the automotive development. Automotive engineers often consider advanced control methods as a pathway to reducing the cost of sensors and actuators, while still achieving robustness and efficiency through software. If the control algorithms are so complex that they require the development of new computational platforms, their appeal is significantly reduced. Hence, the control developers should always strive to fit the controller in the existing computing hardware, rather than assume that computing hardware that is able to execute it will become available.

The common practice found in many research papers of extrapolating the real-time behavior of MPC in an automotive micro-controller unit (aMCU) from the one that is seen in a desktop CPU (dCPU) is potentially very misleading as aMCUs and dCPUs have significantly different capabilities, see Table IV. First, one needs to consider that powerful aMCUs usually run more than ten feedback loops, and probably an even larger number of monitoring loops, and hence the actual computation power available for a single controller is only a fraction of what is available in the entire aMCU. The difference between instruction sets (RISC vs CISC) and the simpler structure of the aMCU memory architecture results in a significantly different numbers of instructions per seconds (IPS) for the aMCUs with respect to dCPUs. Most of the differences are due to the need for the aMCU to work in extreme environments, e.g., ambient temperature ranges between -40^{o}C and $+50^{\text{o}}$C, and even higher near the engine, in which a dCPU is not required to operate and may be even prevented from starting. This is also the cause of the major differences in the size of execution memory, which is normally DRAM in dCPU, but is in general permanent (e.g., SRAM, EPROM, or Flash) in aMCU, with higher cost and physical size, and hence lower quantities and speeds. In fact, for several embedded platforms, memory access may actually be the bottleneck [81].

Because of this, and since also processor-specific code optimization, by engineers or custom compilers, may play a very significant role, the evaluation of the computational load of an MPC controller in an aMCU can only be extrapolated by (in order of precision)

- computing the worst case number of operations per type, the number of instruction per operation type, and hence the total number of instructions per controller execution,
- executing the controller on the specific platform, computing the cost per iteration, and estimating the allowed number of iterations per sampling period,
- evaluating the execution time in a dCPU, estimating the ratio of IPS between dCPU and aMCU, and using that to estimate the execution time in aMCU.

However, according to Table IV, what is often restricting is the memory, both for execution and data. Hence the memory occupancy of the controller is something to be very mindful of. Indeed, PIDs need a minimal amount of memory, 3 gains, 2 extra variables for integral and derivative error, and very few instructions. MPC requires significantly more program and data memory than PID, and hence a careful choice of the numerical algorithm is often critical to the success of the application. Based on the previous discussions, algorithms with limited memory usage and relatively simple code may be preferred, at least initially.

A. Explicit MPC

Explicit MPC has had a significant impact on the implementation of the first MPC solutions in experimental vehicles. Some examples, tested in fully functional and road-drivable (and in several cases road-driven) vehicles are in [13], [23], [25], [27], [29], [59], [62], [73], [76], [82].

In explicit MPC, the optimizer of the MPC problem is obtained by evaluating a pre-computed function of the current prediction model state, possibly including references and other auxiliary states,

$$u_{\text{mpc}}(x) = \begin{cases} F_1 x + G_1 & \text{if} \quad H_1 x \leq K_1, \\ & \vdots \\ F_s x + G_s & \text{if} \quad H_s x \leq K_s, \end{cases} \tag{35}$$

where s is the total number of regions, see Figure 7 for some examples.

The main advantages of explicit MPC and the reasons it has arguably become, until today, the primary method of MPC deployment in automotive applications are:

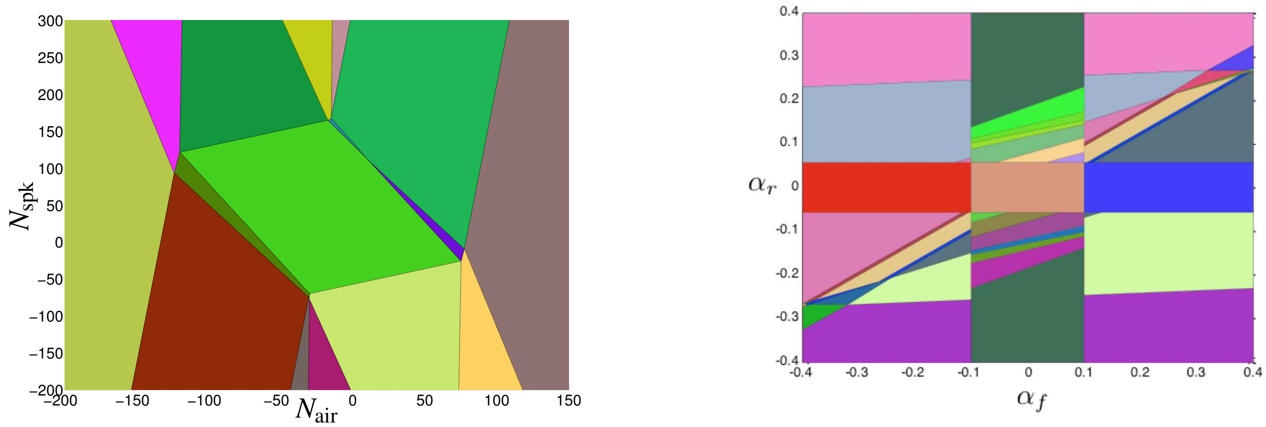

Fig. 7. Section of the partitions of the explicit multivariable linear MPC for idle speed control from [29] (left) and for the switched linear MPC steering controller from [27].

- Simple execution code: explicit MPC is a lookup table of affine controllers, selected by evaluating linear inequalities.
- Basic operations: the controller implementation requires only sums, multiplications, and comparisons.
- Predictability: the worst case number of operations is easily computed.

Additional benefits include the possibility of computing explicit MPC, with few modifications, also for hybrid and switched systems, which allowed for the application of switched and hybrid MPC to automotive control problems [13], [27], [59], [62], [73], [76], the possibility of building explicitly the closed-loop system and hence studying its local and global stability, and the possibility of using only the regions that are most often visited, while enforcing a backup controller otherwise, thus reducing memory requirements.

On the other hand the explicit MPC data memory occupancy and worst case number of operations grows proportionally to the number of active-sets of constraints n_{as}, and hence exponentially with the number of constraints/variables according to

$$n_{as} \leq \sum_{h=0}^{\min\{n_c,n_v\}} \begin{pmatrix} n_c \\ h \end{pmatrix}, \tag{36}$$

where n_v is the number of variables, and n_c is the number of constraints, both of which are proportional to the length of the horizon(s). Because of (36), explicit MPC is limited to applications with relatively short horizon, few control inputs, and few constraints. Another limitation is that the algorithm to construct, the explicit law (35) is too complex to be implemented in the vehicle, and hence explicit MPC cannot be easily tuned or adjusted after deployment.

B. Online MPC

In problems with many control inputs, long prediction horizons and many constraints, explicit MPC may be too complex to store for real-time usage, or even to compute. Also, if the prediction model changes over time, it is very difficult to adjust accordingly the explicit solution, while it is relatively simple to update the data of the optimal control problem. In these cases, the online solution of the MPC optimal control problem may be preferable, and hence online MPC has been applied to automotive problems with the above features.

Among such problems, in [31], a quadratic programming solver and a nonlinear programming solver were used online for controlling an autonomous vehicles, which enabled using a long horizon to take maximum advantage of the known reference trajectory. The nonlinear programming solver was based on sequential quadratic programming, and both the nonlinear program and quadratic programming solvers used active-set methods. In [32] an online active-set solver was used to solve quadratic programs for controlling MAP and MAF in a diesel engine using EGR and VGT. The online solver was used due to the many constraints imposed by the problem, and the need to update the matrices of the model depending on the current operating point, i.e., using in fact a linear-parameter varying model of the diesel engine. In [5], due to the need for using a relatively long horizon for vehicle dynamics cornering performance and stability control at the limit of performance, an interior point solver was used online. While the solvers have been tested in real vehicles, the computing platforms were dedicated rapid prototyping units and custom micro-controllers that may be more capable than production aMCU, in particular because they are dedicated to the controller being developed, while aMCUs run multiple controllers.

Active-set and interior-point methods have fast convergence, but they often use linear algebra libraries for solving systems of linear equations at each iteration of the optimization. Using these libraries may require a fairly large amount of memory, for both data and code storage, and for execution, and achieving portability of these libraries to certain micro-controllers may not be straightforward. Alternatively, first-order methods often have slower convergence, but have much simpler code and hence require less memory and are independent of third parties libraries.

First order methods are essentially based on updating the current solution $z_s^{(h)}$ in the direction indicated by a function h_s of the gradient of the cost function J with a stepsize α_s, followed by the projection onto the feasible set $\mathscr{F}$

$$\hat{z}_s^{(h+1)} = \hat{z}_s^{(h)} - \alpha_s(z_s^{(h)}) \cdot h_s\left(\frac{d}{dz_s}J(z_s)\Big|_{z_s^{(h)}}\right), \tag{37a}$$

$$z_s^{(h+1)} = \mathrm{proj}_{\mathscr{F}}\left(\hat{z}_s^{(h+1)}\right), \tag{37b}$$

where z_s may contain additional variables other than those of the original optimization problem, and the choice of the stepsize α_s, of the function h_s, and of the additional variables, differentiate the methods.

In recent years, several low complexity first-order methods for MPC have been proposed, based on Nesterov's fast gradient algorithm [67], Lagrangian methods [55], nonnegative least squares [22], and alternating direction method of multipliers (ADMM) [35], [38], [64]. For instance, in [24] the method in [22] was used for vehicle trajectory tracking.

C. Nonlinear MPC

Most of the previous discussion is focused on linear MPC because, at least until very recently, that was the only class of MPC that realistically could have been implemented in automotive systems. Nonlinear MPC is significantly more complex, which is to a large extent due to the algorithm for solving the nonlinear programming problem. For instance, in [31] the authors after implementing a nonlinear method, chose to implement a linear time-varying method based on local linearization and quadratic programming, to reduce the computational load. As another example, the paper [41] concerned with diesel engine air path control states that "currently it is not possible to implement NMPC in real time due to the limited computational power available". However, this is starting to change in more recent years, in part thanks to the research aimed at tailoring nonlinear solvers to MPC problems.

Some applications of nonlinear MPC to automotive systems [34], [53], [80] are based on the C/GMRES method reported in [60]. This method appears quite effective if the objective is to solve a nonlinear optimal control problem with equality constraints, only few inequality constraints, and few changes of the active-set. This is due to the changes to the active sets possibly causing discontinuities in the dual variables, and sometimes also in the primal variables, see, e.g., [54], which is in conflict with the smooth update of the continuation methods. Various inequality constraint enforcement techniques for diesel engines in the context of the method in [60] are considered and compared in [49].

More recently [1], [33], some applications are being investigated based on the so-called real-time iteration (RTI) scheme [43], which is based on combining effective integrators for multiple-shooting methods with sequential quadratic programming, where usually only one step of optimization is actually performed.

Expanding the reliability and reducing the computational cost of nonlinear MPC methods will probably be a key effort in the upcoming years to allow for significant use in automotive applications.

V. Conclusions and Future Perspectives

MPC has been extensively employed for research in automotive control, and is maturing for product deployment. The main opportunities are in using MPC for optimal multivariable constrained control, for exploiting preview information, and for handling systems subject to time delays. As a consequence, MPC has been investigated in several applications in powertrain, lateral and longitudinal vehicle dynamics, and energy management of HEV. In the upcoming years the number of applications of MPC to autonomous vehicles [31], [51], [57] is expected to grow, where MPC may need to be integrated with higher level planning methods [19], [24], decision logics [78], and connected cooperative driving [61], [72], possibly within some kind of distributed architecture.

While many challenges presented by MPC deployment have been overcome by research on specific applications, research efforts are still necessary to address them in general ways, and some challenges for product deployment are still not entirely solved. These include the construction of models, and effective approximations thereof, the numerical algorithms for linear [22], [38], [64], [67] and nonlinear MPC [43], [60], the calibration and reconfiguration methods [18], [20], and the design process, to which, hopefully, this chapter has contributed.

References

[1] Albin, T., Ritter, D., Abel, D., Liberda, N., Quirynen, R., Diehl, M.: Nonlinear MPC for a two-stage turbocharged gasoline engine airpath. In: Proc. 54th IEEE Conf. on Dec. and Control, pp. 849–856 (2015)

[2] Amari, R., Alamir, M., Tona, P.: Unified MPC strategy for idle-speed control, vehicle start-up and gearing applied to an Automated Manual Transmission. In: Proc. IFAC World Congress. Seoul, South Korea (2008)

[3] Arce, A., Alejandro, J., Bordons, C., Ramirez, D.R.: Real-time implementation of a constrained MPC for efficient airflow control in a pem fuel cell. IEEE Trans. Industrial Electronics **57**(6), 1892–1905 (2010)

[4] Bambang, R.T., Rohman, A.S., Dronkers, C.J., Ortega, R., Sasongko, A., et al.: Energy management of fuel cell/battery/supercapacitor hybrid power sources using model predictive control. IEEE Trans. Industrial Informatics **10**(4), 1992–2002 (2014)

[5] Beal, C.E., Gerdes, J.C.: Model predictive control for vehicle stabilization at the limits of handling. IEEE Trans. Contr. Systems Technology **21**(4), 1258–1269 (2013)

[6] Beck, R., Richert, F., Bollig, A., Abel, D., Saenger, S., Neil, K., Scholt, T., Noreikat, K.E.: Model predictive control of a parallel hybrid vehicle drivetrain. In: Proc. 44th IEEE Conf. on Dec. and Control, pp. 2670–2675 (2005)

[7] Bemporad, A., Borrelli, F., Glielmo, L., Vasca, F.: Optimal piecewise-linear control of dry clutch engagement. In: 3rd IFAC Workshop on Adv. Automotive Control, pp. 33–38. Karlsruhe, Germany (2001)

[8] Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica **35**(3), 407–427 (1999)

[9] Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.: The Explicit Linear Quadratic Regulator for Constrained Systems. Automatica **38**(1), 3–20 (2002)

[10] Bengtsson, J., Strandh, P., Johansson, R., Tunestal, P., Johansson, B.: Model predictive control of homogeneous charge compression ignition (hcci) engine dynamics. In: Proc. IEEE Int. Conf. on Control Applications, pp. 1675 –1680. Munich, Germany (2006)

[11] Bichi, M., Ripaccioli, G., Di Cairano, S., Bernardini, D., Bemporad, A., Kolmanovsky, I.: Stochastic model predictive control with driver behavior learning for improved powertrain control. In: Proc. 49th IEEE Conf. on Dec. and Control, pp. 6077–6082. Atlanta, GA (2010)

[12] Borhan, H., Vahidi, A., Phillips, A.M., Kuang, M.L., Kolmanovsky, I.V., Di Cairano, S.: MPC-based energy management of a power-split hybrid electric vehicle. IEEE Trans. Contr. Systems Technology **20**(3), 593–603 (2012)

[13] Borrelli, F., Bemporad, A., Fodor, M., Hrovat, D.: An MPC/hybrid system approach to traction control. IEEE Trans. Contr. Systems Technology **14**(3), 541–552 (2006)

[14] Canale, M., Milanese, M., Novara, C.: Semi-active suspension control using fast model-predictive techniques. IEEE Trans. Contr. Systems Technology **14**(6), 1034–1046 (2006)

[15] Caruntu, C.F., Lazar, M., Gielen, R.H., van den Bosch, P., Di Cairano, S.: Lyapunov based predictive control of vehicle drivetrains over can. Control Engineering Practice **21**(12), 1884–1898 (2013)

[16] Carvalho, A., Gao, Y., Lefevre, S., Borrelli, F.: Stochastic predictive control of autonomous vehicles in uncertain environments. In: 12th Int. Symp. Adv. Vehicle Control (2014)

[17] Di Cairano, S.: An industry perspective on MPC in large volumes applications: Potential benefits and open challenges. In: 4th IFAC Symp. Nonlinear Model Predictive Control, pp. 52–59 (2012)

[18] Di Cairano, S.: Model adjustable predictive control with stability guarantees. In: Proc. American Contr. Conf., pp. 226–231 (2015)

[19] Di Cairano, S.: Control and optimization of autonomous vehicles. In: IEEE-VTS Connected and Autonomous Vehicles Summer School (2016). Http://resourcecenter.vts.ieee.org/vts/product/events/VTSEVTWPI003

[20] Di Cairano, S., Bemporad, A.: Model predictive control tuning by controller matching. IEEE Trans. Automatic Control **55**(1), 185–190 (2010)

[21] Di Cairano, S., Bemporad, A., Kolmanovsky, I., Hrovat, D.: Model predictive control of magnetically actuated mass spring dampers for automotive applications. Int. J. Control **80**(11), 1701–1716 (2007)

[22] Di Cairano, S., Brand, M., Bortoff, S.A.: Projection-free parallel quadratic programming for linear model predictive control. Int. J. Control **86**(8), 1367–1385 (2013)

[23] Di Cairano, S., Doering, J., Kolmanovsky, I.V., Hrovat, D.: Model predictive control of engine speed during vehicle deceleration. IEEE Trans. Contr. Systems Technology **22**(6), 2205–2217 (2014)

[24] Di Cairano, S., Kalabić, U., Berntorp, K.: Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds. In: Proc. 55th IEEE Conf. on Dec. and Control, pp. 709–714 (2016)

[25] Di Cairano, S., Liang, W., Kolmanovsky, I.V., Kuang, M.L., Phillips, A.M.: Power smoothing energy management and its application to a series hybrid powertrain. IEEE Trans. Contr. Systems Technology **21**(6), 2091–2103 (2013)

[26] Di Cairano, S., Pascucci, C.A., Bemporad, A.: The rendezvous dynamics under linear quadratic optimal control. In: Proc. 51st IEEE Conf. on Dec. and Control, pp. 6554–6559 (2012)

[27] Di Cairano, S., Tseng, H., Bernardini, D., Bemporad, A.: Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. IEEE Trans. Contr. Systems Technology **21**(4), 1236–1248 (2013)

[28] Di Cairano, S., Yanakiev, D., Bemporad, A., Kolmanovsky, I., Hrovat, D.: An MPC design flow for automotive control and applications to idle speed regulation. In: Proc. 48th IEEE Conf. on Dec. and Control, pp. 5686–5691 (2008)

[29] Di Cairano, S., Yanakiev, D., Bemporad, A., Kolmanovsky, I.V., Hrovat, D.: Model predictive idle speed control: Design, analysis, and experimental evaluation. IEEE Trans. Contr. Systems Technology **20**(1), 84–97 (2012)

[30] Donahue, M.D., Hedrick, J.K.: Implementation of an active suspension preview controller for improved ride comfort. In: R. Johansson, A. Rantzer (eds.) Nonlinear and hybrid systems in automotive control, vol. 146, pp. 1–22. Springer London (2003)

[31] Falcone, P., Borrelli, F., Asgari, J., Tseng, H., Hrovat, D.: Predictive active steering control for autonomous vehicle systems. IEEE Trans. Contr. Systems Technology **15**(3), 566–580 (2007)

[32] Ferreau, H.J., Ortner, P., Langthaler, P., Del Re, L., Diehl, M.: Predictive control of a real-world diesel engine using an extended online active set strategy. Annual Reviews in Control **31**(2), 293–301 (2007)

[33] Frasch, J.V., Gray, A., Zanon, M., Ferreau, H.J., Sager, S., Borrelli, F., Diehl, M.: An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. In: Proc. European Control Conf., pp. 4136–4141 (2013)

[34] Gagliardi, D., Othsuka, T., del Re, L.: Direct C/GMRES control of the air path of a diesel engine. In: Proc. 19th IFAC World Congress, pp. 3000–3005 (2014)

[35] Ghadimi, E., Teixeira, A., Shames, I., Johansson, M.: Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. IEEE Trans. Automatic Control **60**(3), 644–658 (2015)

[36] Giorgetti, N., Bemporad, A., Tseng, E.H., Hrovat, D.: Hybrid model predictive control application towards optimal semi-active suspension. Int. J. Control **79**(5), 521–533 (2006)

[37] Giorgetti, N., Ripaccioli, G., Bemporad, A., Kolmanovsky, I., Hrovat, D.: Hybrid Model Predictive Control of Direct Injection Stratified Charge Engines. IEEE/ASME Trans. Mechatronics **11**(5), 499–506 (2006)

[38] Giselsson, P., Boyd, S.: Linear convergence and metric selection for Douglas-Rachford splitting and ADMM. IEEE Trans. Contr. Systems Technology **62**(2), 532–544 (2017)

[39] Gohrle, C., Schindler, A., Wagner, A., Sawodny, O.: Design and vehicle implementation of preview active suspension controllers. IEEE Trans. Contr. Systems Technology **22**(3), 1135–1142 (2014)

[40] Hatanaka, T., Yamada, T., Fujita, M., Morimoto, S., Okamoto, M.: Explicit receding horizon control of automobiles with continuously variable transmissions. In: Nonlinear model predictive control, *Lec. Not. in Computer Science*, vol. 384, pp. 561–569. Springer (2009)

[41] Herceg, M., Raff, T., Findeisen, R., Allgower, F.: Nonlinear model predictive control of a turbocharged diesel engine. In: Proc. IEEE Int. Conf. on Control Applications, pp. 2766–2771 (2006)

[42] Heywood, J.: Internal Combustion Engine Fundamentals. McGraw-Hill, New York, USA (1988)

[43] Houska, B., Ferreau, H.J., Diehl, M.: An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. Automatica **47**(10), 2279–2285 (2011)

[44] Hrovat, D.: MPC-based idle speed control for IC engine. In: Proc. of FISITA conference (1996)

[45] Hrovat, D.: Survey of advanced suspension developments and related optimal control applications. Automatica **33**(10), 1781–1817 (1997)

[46] Hrovat, D., Di Cairano, S., Tseng, H., Kolmanovsky, I.: The development of model predictive control in automotive industry: A survey. In: Proc. IEEE Int. Conf. on Control Applications, pp. 295–302. Dubrovnik, HR (2012)

[47] Hsieh, M.F., Wang, J.: Diesel engine selective catalytic reduction (scr) ammonia surface coverage control using a computationally-efficient model predictive control assisted method. In: ASME Dynamic Systems Control Conf., pp. 865–872 (2009)

[48] Hsieh, M.F., Wang, J., Canova, M.: Two-level nonlinear model predictive control for lean NOx trap regenerations. Jour. Dynamic Systems, Measurement, and Control **132**(4), 041,001 (2010)

[49] Huang, M., Nakada, H., Butts, K., Kolmanovsky, I.: Nonlinear model predictive control of a diesel engine air path: A comparison of constraint handling and computational strategies. In: 5th IFAC Symp. Nonlinear Model Predictive Control, pp. 372–379 (2015)

[50] Huang, M., Zaseck, K., Butts, K., Kolmanovsky, I.: Rate-based model predictive controller for diesel engine air path: Design and experimental evaluation. IEEE Trans. Contr. Systems Technology **24**(6), 1922–1935 (2016)

[51] Jalalmaab, M., Fidan, B., Jeon, S., Falcone, P.: Model predictive path planning with time-varying safety constraints for highway autonomous driving. In: Int. Conf. Adv. Robotics, pp. 213–217 (2015)

[52] Jazar, R.N.: Vehicle dynamics: theory and application. Springer Science & Business Media (2013)

[53] Kaijiang, Y., Mukai, M., Kawabe, T.: Performance of an eco-driving nonlinear MPC system for a power-split hev during car following. SICE J. Control, Measurement, and System Integration **7**(1), 55–62 (2014)

[54] Kalabic, U., Gupta, R., Di Cairano, S., Bloch, A., Kolmanovsky, I.: Constrained spacecraft attitude control on so (3) using reference governors and nonlinear model predictive control. In: Proc. American Contr. Conf., pp. 5586–5593 (2014)

[55] Kögel, M., Findeisen, R.: Fast predictive control of linear systems combining nesterov's gradient method and the method of multipliers. In: Proc. 50th IEEE Conf. on Dec. and Control, pp. 501–506 (2011)

[56] Koot, M., Kessels, J.T., De Jager, B., Heemels, W., Van den Bosch, P., Steinbuch, M.: Energy management strategies for vehicular electric power systems. IEEE Trans. Vehicular Technology **54**(3), 771–782 (2005)

[57] Makarem, L., Gillet, D.: Model predictive coordination of autonomous vehicles crossing intersections. In: 16th Int. IEEE Conf. Intell. Transportation Systems, pp. 1799–1804 (2013)

[58] Mehra, R.K., Amin, J.N., Hedrick, K.J., Osorio, C., Gopalasamy, S.: Active suspension using preview information and model predictive control. In: Proc. IEEE Int. Conf. on Control Applications, pp. 860–865. IEEE (1997)

[59] Naus, G., Ploeg, J., Van de Molengraft, M., Heemels, W., Steinbuch, M.: Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach. Control Engineering Practice **18**(8), 882–892 (2010)

[60] Ohtsuka, T.: A continuation/GMRES method for fast computation of nonlinear receding horizon control. Automatica **40**(4), 563–574 (2004)

[61] Ong, H.Y., Gerdes, J.C.: Cooperative collision avoidance via proximal message passing. In: Proc. American Contr. Conf., pp. 4124–4130 (2015)

[62] Ortner, P., del Re, L.: Predictive Control of a Diesel Engine Air Path. IEEE Trans. Contr. Systems Technology **15**(3), 449–456 (2007)

[63] Ozatay, E., Onori, S., Wollaeger, J., Ozguner, U., Rizzoni, G., Filev, D., Michelini, J., Di Cairano, S.: Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution. IEEE Trans. Intelligent Transp. Systems **15**(6), 2491–2505 (2014)

[64] Raghunathan, A.U., Di Cairano, S.: Infeasibility detection in alternating direction method of multipliers for convex quadratic programs. In: Proc. 53rd IEEE Conf. on Dec. and Control, pp. 5819–5824 (2014)

[65] Rajamani, R.: Vehicle dynamics and control. Springer Science & Business Media (2011)

[66] Ravi, N., Liao, H.H., Jungkunz, A.F., Widd, A., Gerdes, J.C.: Model predictive control of hcci using variable valve actuation and fuel injection. Control Engineering Practice **20**(4), 421–430 (2012)

[67] Richter, S., Jones, C.N., Morari, M.: Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. IEEE Trans. Automatic Control **57**(6), 1391–1403 (2012)

[68] Ripaccioli, G., Bemporad, A., Assadian, F., Dextreit, C., Di Cairano, S., Kolmanovsky, I.: Hybrid Modeling, Identification, and Predictive Control: An Application to Hybrid Electric Vehicle Energy Management. In: Hybrid Systems: Computation and Control, *Lec. Not. in Computer Science*, vol. 5469, pp. 321–335. Springer (2009)

[69] Ripaccioli, G., Bernardini, D., Di Cairano, S., Bemporad, A., Kolmanovsky, I.: A stochastic model predictive control approach for series hybrid electric vehicle power management. In: Proc. American Contr. Conf., pp. 5844–5849. Baltimore, MD (2010)

[70] Santillo, M., Karnik, A.: Model predictive controller design for throttle and wastegate control of a turbocharged engine. In: Proc. American Contr. Conf., pp. 2183–2188 (2013)

[71] Schallock, R., Muske, K., Peyton Jones, J.: Model predictive functional control for an automotive three-way catalyst. SAE Int. J. Fuels and Lubricants **2**(1), 242–249 (2009)

[72] Stanger, T., del Re, L.: A model predictive cooperative adaptive cruise control approach. In: Proc. American Contr. Conf., pp. 1374–1379 (2013)

[73] Stewart, G., Borrelli, F.: A Model Predictive Control Framework for Industrial Turbodiesel Engine Control. In: Proc. 48th IEEE Conf. on Dec. and Control, pp. 5704–5711. Cancun, Mexico (2008)

[74] Trimboli, S., Di Cairano, S., Bemporad, A., Kolmanovsky, I.: Model predictive control for automotive time-delay processes: an application to air-to-fuel ratio. In: Proc. 8th IFAC Workshop on Time-delay Systems, pp. 1–6 (2009)

[75] Vahidi, A., Stefanopoulou, A.G., Peng, H.: Current management in a hybrid fuel cell power system: A model-predictive control approach. IEEE Trans. Contr. Systems Technology **14**(6), 1047–1057 (2006)

[76] Van Der Heijden, A.C., Serrarens, A.F.A., Camlibel, M.K., Nijmeijer, H.: Hybrid optimal control of dry clutch engagement. Int. J. Control **80**(11), 1717–1728 (2007)

[77] Vermillion, C., Butts, K., Reidy, K.: Model predictive engine torque control with real-time driver-in-the-loop simulation results. In: Proc. American Contr. Conf., pp. 1459–1464 (2010)

[78] Wongpiromsarn, T., Topcu, U., Murray, R.M.: Receding horizon temporal logic planning. IEEE Trans. Automatic Control **57**(11), 2817–2830 (2012)

[79] Yan, F., Wang, J., Huang, K.: Hybrid electric vehicle model predictive control torque-split strategy incorporating engine transient characteristics. IEEE Trans. Vehicular Technology **61**(6), 2458–2467 (2012)

[80] Yu, K., Mukai, M., Kawabe, T.: A battery management system using nonlinear model predictive control for a hybrid electric vehicle. In: IFAC Symp. Adv. on Automotive Control, pp. 301–306 (2013)

[81] Yu, L., Goldsmith, A., Di Cairano, S.: Efficient convex optimization on gpus for embedded model predictive control. In: Proc. ACM General Purpose GPUs, pp. 12–21 (2017)

[82] Zhao, D., Liu, C., Stobart, R., Deng, J., Winward, E., Dong, G.: An explicit model predictive control framework for turbocharged diesel engines. IEEE Trans. Industrial Electronics **61**(7), 3540–3552 (2014)