Real-time optimization and model predictive control for aerospace and automotive applications

Di Cairano, S.; Kolmanovsky, I.V.

TR2018-086 July 13, 2018

Abstract

In recent years control methods based on realtime optimization (RTO) such as model predictive control (MPC) have been investigated for a significant number of applications in the automotive and aerospace (A&A) domains. This paper provides a tutorial overview of RTO in automotive and aerospace, with particular focus on MPC which is probably the most largely investigated method. First, we review the features that make RTO appealing for A&A applications. Then, due to the model-based nature of these control methods, we describe the key first principle models and opportunities that these provide for RTO. Next, we detail the key steps and guidelines of the MPC design process which are tailored to A&A systems. Finally, we discuss numerical algorithms for implementing RTO, and their suitability for implementation in embedded computing platforms to in A&A domains

American Control Conference (ACC)

Real-time optimization and model predictive control for aerospace and automotive applications

Stefano Di Cairano[1] Ilya V. Kolmanovsky[2]

Abstract— **In recent years control methods based on real-time optimization (RTO) such as model predictive control (MPC) have been investigated for a significant number of applications in the automotive and aerospace (A&A) domains. This paper provides a tutorial overview of RTO in automotive and aerospace, with particular focus on MPC which is probably the most largely investigated method. First, we review the features that make RTO appealing for A&A applications. Then, due to the model-based nature of these control methods, we describe the key first principle models and opportunities that these provide for RTO. Next, we detail the key steps and guidelines of the MPC design process which are tailored to A&A systems. Finally, we discuss numerical algorithms for implementing RTO, and their suitability for implementation in embedded computing platforms to in A&A domains.**

I. Introduction

There are very few devices as pervasive in our world as cars. Reports show that close to 90 million cars and light commercial vehicles were sold 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. While different from automotive system in the amount of distribution and accessibility to the general public, aerospace systems are also increasingly pervasive in our lives. Air traffic is constantly increasing and more and more of our services are routed through or provided by satellites. Thus, also in the case of aerospace system there is constant need of increase in fuel efficiency, durability, and robustness to adverse conditions. In summary, in both Automotive and Aerospace (A&A) domains, advanced control methods that are capable of optimizing the system operation and of reducing the time-to-market for increasingly complex systems are clearly needed.

It thus comes as no surprise that, in recent years, a significant interest in control based on Real-Time Optimization (RTO) has been shown in the automotive and aerospace industries. In particular, the research on applications of Model Predictive Control (MPC) to A&A systems has been steadily growing both in industry and academia to address some of the aforementioned needs. Yet MPC is a significant step up from the classical control methods, such as PID, and thus it presents several challenges for implementing it into industrial practice.

This contribution aims at providing a short tutorial on the developments in RTO for A&A systems, with main focus on MPC-based solutions. To this end, first in Section II we highlight the benefits that MPC can provide as well as challenges faced by MPC in these domains. Then, given that RTO is usually a model-based control approach, in Sections III and IV we describe some of the key first principle models that can be used for MPC design, and the control objectives that need to be achieved for the main sub-areas of automotive and aerospace control. Next, In Section V we detail the common steps of MPC design for automotive systems. Finally, in Section VI we consider the computational aspects that are important for real-time implementation and deployment of MPC solutions on computing platforms for automotive and aerospace applications. Conclusions are drawn in Section VII.

II. Automotive and Aerospace Opportunities and Challenges for RTO and MPC

Due to regulations, competition, and customer demands, automotive and aerospace 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 A&A control problems has been mainly pursued due to the features that are helpful and effective in addressing such requirements and optimize operation. The key strengths of MPC are summarized in Table I and discussed next.

A solid starting point for MPC development is that while the processes and dynamics taking place in automotive and aerospace vehicles 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.

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 in the automotive domain are turbochargers, variable cam timing, electric motors, variable steering, differential braking, regenerative braking. Some examples in spacecraft are different propulsion systems, such as chemical and electrical, used in conjunction with momentum exchanging devices such as reaction wheels, control moment gyroscopes, and magnetic attitude actuators, and the use of multiple control surfaces, aileron, flaperon and canard, and thrust vectoring in aircraft.

[1] Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA dicairano@ieee.org

[2] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA ilya@umich.edu

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
Taking advantage of preview	

TABLE I

STRENGTHS AND CHALLENGES FOR MPC IN AUTOMOTIVE AND AEROSPACE APPLICATIONS.

As more actuators become available, methods that can coordinate them to achieve multiple objectives, i.e., capable of controlling multivariable, multiobjective systems, may achieve superior performance with respect to controllers 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 [1]–[3], vehicle-stability control by coordinated steering and braking [4], [5], airpath control in turbocharged diesel engines [6], [7], momentum management in satellites [8], [9]. Furthermore, while it may still be difficult to obtain globally robust MPC designs, it is well known that MPC has 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 regulations and interaction with other vehicles or vehicle systems can often be easily formulated in terms of constraints on process variables. For instance in rendezvous and docking, proper operation can be enforced by imposing constraints enforcing sensor line of sight and soft docking [10], [11]. 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 [3], [5], [7], [12], [13].

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 [14]–[17]. Fuel consumption optimization is also sought in several aerospace applications, see, e.g., [8], [9], to increase lifespan and reduce costs. Even if it may be difficult to directly formulate the A&A performance measures as a cost function for MPC, it is usually possible to formulate indirect objectives [2], [17] that, when optimized, imply quasi-optimal (or at least very desirable) behavior with respect to the actual performance measures.

Besides these macro-features, MPC has additional capabilities that are useful in controlling A&A 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, allow for obtaining preview information that MPC can exploit to achieve superior performance [18], [19]. This is even more relevant in the context of connected and autonomous vehicles [20], due to the available long term information, for instance, from mid range and long range path planners, and from shared information among vehicles. In spacecraft, the capability of exploiting future information allows to take advantage of prediction of periodic disturbances [8].

Another motivating factor is the increase in autonomy achieved by RTO and MPC. This is especially relevant to aerospace applications where flight trajectories subject to state and control constraints may be autonomously recomputed upon discovery, for instance, of deviations, obstacles or failure modes. For a spacecraft operating near a remote planet or an asteroid, the communication delay with earth is too large to attempt to remotely pilot around an obstacle or in a loss of control situations. A similar situation may occur for UAVs due to communication delays through relay satellite network or jammed communications.

However, there are also several challenges to the large scale deployment of MPC in automotive applications [21], which are also summarized in Table I and discussed next.

First, MPC has larger computational load and memory footprint than classical control methods, while embedded micro-controllers for A&A applications are fairly limited in terms of computing power. This is in part due to the harsh environment in which the microcontroller has to operate. For instance, automotive microcontrollers must operate in temperatures ranging from -40^{o}C to $+50^{o}$C, and for spacecraft in high orbits, the temperature range can be from -250^{o}C to $+300^{o}$C. Spacecraft are also subject to large amount of radiation, e.g., from cosmic rays and solar particles, which requires specific hardening techniques, both physical and logical, to be counteracted. As a consequence, the achievable processor and memory access frequencies in such environments are limited. Furthermore, the need of predictable computation times limits the usage of many of the components that make desktop computer faster, such as caches, pipelining, and multi-threading. Finally, cost reduction and lengthy development and validation time often prevents the processor being sized for a specific controller. Rather, the controller must fit in the given target processor.

Second, not all the processes have well-developed models. Processes such as combustion, battery charging/discharging, orbiting around small bodies 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 these settings.

Even for the processes that are better understood, the dynamics are intrinsically nonlinear. The relative motion of satellites operating in specific orbits are governed by approximately linear dynamics, but, while changing orbit, the nonlinearities may be relevant. This is even more pervasive in automotive due to the the to external effects, e.g., the driver, the traffic, the road, continuously subjecting the processes to fast transients during which the nonlinearities are excited.

A further complicating factor is that several variables in A&A processes are not measured, and the sensors for estimating them may be heavily quantized and noisy or remotely located. A fourth challenge for MPC, which needs the system state for initializing the prediction model, is the need of state estimator, whose performance will significantly affect the overall performance of the closed-loop system. The estimator performance depends on the sensors that in automotive may be reduced in number and have limited capabilities, due to cost and harsh environment, while in aerospace may be remotely located, or could be only intermittently used due to power consumption or interferences.

Fifth and final challenge, is the difference in the development process of MPC and classical controllers, e.g., PID. This also complicates the enforcement of the strict validation and verification and certification requirements in aerospace applications. PIDs are calibrated by tuning the gains, and the resulting closed-loop systems are characterized by their frequency responses, i.e., gain and phase at different frequencies, and their stability margins. Since they are the dominant method in control applications, the majority of the standards for control development and certification are defined in terms of these. On the other hand, MPC is calibrated by choosing the prediction model, the horizon, and the cost function and its weights, and the resulting closed-loop systems are characterized by optimality properties, induced Lypaunov functions, sets of recursive feasibility, and convergence rates of the optimization algorithms. The translation of these concepts into those of classical control is not simple, and hence engineers familiar with PIDs may find difficulties with the development and maintenance of MPC, as well as with its certification.

The above challenges have to kept in mind when developing new A&A applications of RTO, so that appropriate risk mitigation strategies can be planned. At the same time, they offer relevant research directions.

III. Automotive Applications and Models

In this section we review key areas of automotive control where RTO has been been investigated. For each, we first describe the key models for control development, and then, based on these, we discuss what impact RTO may have.

A. Powertrain Control

Powertrains are responsible for generating engine torque and transferring such torque to the wheels to achieve traction.

The engine model describes the effects of the engine actuators and operating conditions on the torque that the engine produces and on pressures, flows and temperatures in different parts of the engine. The engine actuators range from the standard throttle, fuel injectors, and spark timing, to more advanced 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 how the torque is generated from the combustion of the gas mixture.

For basic spark ignition (SI) engines, i.e., conventional gasoline engines without turbocharger, VCT or EGR (see the schematic in Figure 1(a)), the airpath model is relatively simple and includes 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,

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

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

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

where W, p, T, V, denote mass flow, pressure, temperature, and volume, respectively, ϕ is a nonlinear function modeling the throttle flow dependence on the pressure ratio across the throttle [22, 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, V_d is the engine displacement volume, η_{vol} is the volumetric efficiency, R is the gas constant, A_{th} is the throttle effective flow area, which is a function of throttle position, 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 1(b)), the airpath model is substantially more complex, because these engines are usually turbocharged and exploit EGR, which renders the isothermal assumption inaccurate. Furthermore, the EGR valve and the turbocharger couple the intake manifold with the exhaust manifold, which then must be included in the model. As a result, the diesel engine model includes pressures, densities (ρ) and burned gas fraction (F) in both the intake, and exhaust (*em*) manifolds,

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

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

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

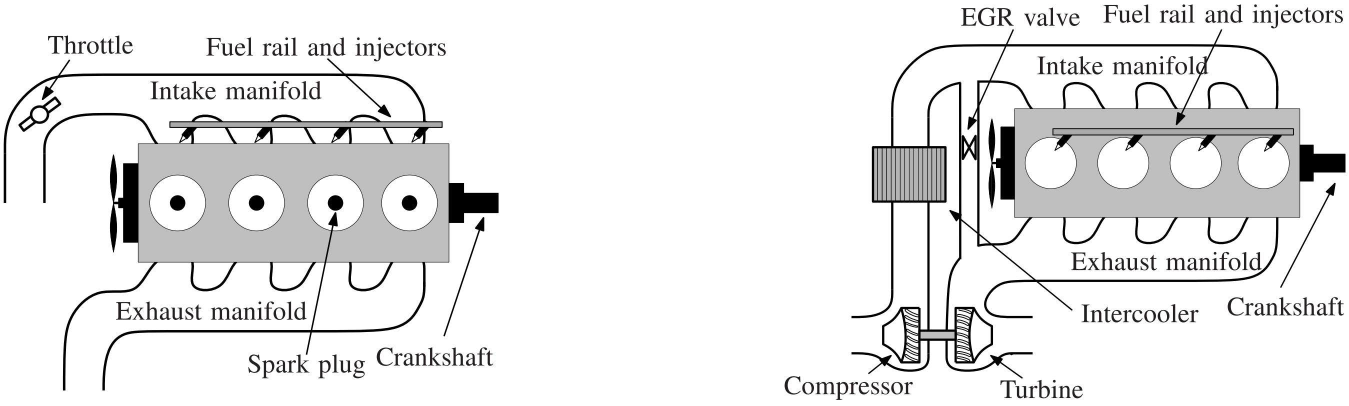
$$\dot{p}_{\mathrm{em}} = \frac{c_p R}{c_v V_{\mathrm{em}}}(W_{\mathrm{cyl}} T_{\mathrm{cyl}} - W_{\mathrm{tur}} T_{\mathrm{em}} - W_{\mathrm{egr}} T_{\mathrm{em}} - \dot{Q}_{\mathrm{em}}/c_p), \tag{2d}$$

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

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

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



(a)

(b)

Fig. 1. Schematics of a spark ignition engine (a), and of a turbocharged compression ignition engine (b).

same for the SI engine, (1b), and the EGR flow is controlled by a valve with equation similar to (1c), the remaining flows are determined by the turbocharger equations,

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

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

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

where ep refers to the exhaust pipe, χ_{vgt} is the variable geometry turbine control, N_{tc} and J_{tc}, are the speed and inertia of the turbocharger, ϕ_{com}, ϕ_{tur}, η_{com}, η_{tur}, are the flow parameter and efficiency of compressor and turbine.

While (1) models a basic case, many modern SI engines have additional degrees of freedom, such as VCT. For instance, due to the recent diffusion of vehicles with downsized powertrains, gasoline engines that are turbocharged have become more common. Their airpath model is a hybrid between the SI and CI models, since the SI combustion is controlled through spark and throttle, but an EGR and turbocharger are also available as actuators, although, in general, with a smaller fixed geometry turbine and a wastegate valve.

The second component of the engine model is the torque production model, which describes the net torque output generated by the engine. This 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 t_d is the torque production delay, and M_{ind}, M_{fr}, M_{pmp} are the indicated, friction, and pumping torques. The indicated torque is the produced torque and its expression depends on the engine type. For SI engines,

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

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

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 may vary over a broad range, the indicated torque equation is usually obtained from engine calibration data, e.g., as

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

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 model represents the torque transfer from the engine to the wheels. The engine speed 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. Ignoring 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 simple 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 are in closed-pedal operation, i.e., when the gas pedal is released, and the vehicle is in the speed control mode.

An example is idle speed control [2] 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, e.g., due to knocking or misfiring. Thus, the optimal control problem can be formulated as

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

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

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

A more challenging application is deceleration control, when 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 2. In this case the problem is similar to (8), but 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 problems. Since idling takes place near a setpoint, a linearized model for idling is fairly accurate. On the other hand, the deceleration control is more involved. For the latter, it is often convenient to develop a low-level controller that linearizes the dynamics. Then, MPC can exploit constraints to ensure that the interaction with such low level controller is effective. For deceleration control, in [3] 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 and the spark actuation delay t_{ds} is also considered

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

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

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

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

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

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 actuates the VGT and EGR valve to track these setpoints. Also in this case the result is 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 [23] by pursuing a rate-based formulation, constraint remodeling, intermittent constraint enforcement, and through a combination with nonlinear static or dynamic inversion.

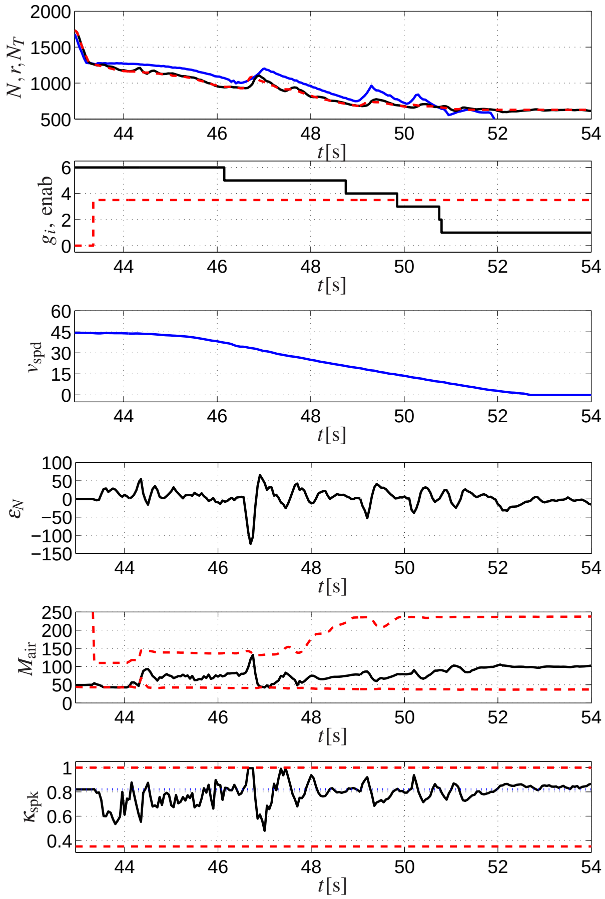


Fig. 2. Experimental test of MPC-based deceleration control from [3]. Engine speed N, reference r, and tracking error ε_N, torque converted turbine speed N_T, gear g_i and controller enabling *enab*, vehicle speed v_{spd}, torque from airflow M_{air} and torque ratio from spark, κ_{spk}.

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, a single track model, also known as bicycle model, shown in Figure 3, is appropriate. This model is described by

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

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

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

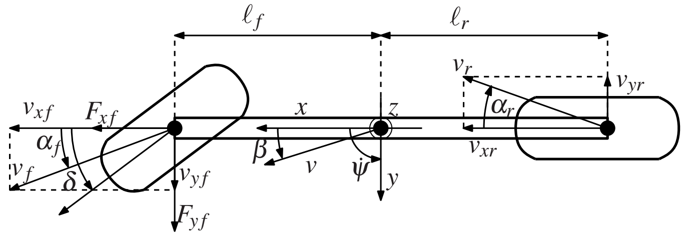


Fig. 3. Schematics of single track model of the lateral vehicle dynamics. The relevant vectors for involved are also shown.

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 [24],

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

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 $j \in \{f,r\}$. The slip angles and the slip ratios relate the tractive forces with the vehicle velocity and wheel speed, thereby coupling the vehicle response with the powertrain response,

$$\begin{aligned} v_{lj} &= v_{y_j} \sin\delta_j + v_{xj}\cos\delta_j, & v_{cj} &= v_{y_j}\cos\delta_j - v_{xj}\sin\delta_j, \\ \alpha_j &= \tan^{-1}\left(\frac{v_{y_j}}{v_{xj}}\right), & \sigma_j &= \frac{r\omega_j - v_{xj}}{\max\{r\omega_j, v_{xj}, \varepsilon\}} \end{aligned} \tag{12}$$

where v_{xj}, v_{y_j}, $j \in \{f,r\}$, are the longitudinal and lateral components of the tire velocity vectors, and ε is a small constant to avoid singularity. In (11), f_l, f_c define the tire forces that are generated as functions of the slip angles and slip ratio, of the vertical force, and of the friction coefficient and are in general determined by data or according to a model such as Paceijka's formula or Lu'Gre [25].

In the 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 functions 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 $\alpha_f = (v_y + \ell_f\dot{\psi})/v_x$, $\alpha_r = (v_y - \ell_r\dot{\psi})/v_x$, and we have included the moment M_{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

$$m\dot{v}_x = C_f^x\sigma_f + C_r^x\sigma_r - F_{\text{res}}, \tag{14a}$$

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

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 acceleration due to gravity. 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 applied from the powertrain, to the wheels F_{trac} is the response of a first order-plus-delay system including a multiplicative factor modeling losses, $\gamma_{\text{ls}} \in (0,1)$,

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

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

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

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.

1) MPC opportunities in vehicle dynamics: MPC for longitudinal vehicle dynamics has been applied to adaptive cruise control (ACC), see, e.g., [26]. 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

$$\min_{F_{\text{trac}}} \quad \sum_{t=0}^{T_N}(v_x(t) - r_v(t))^2 + w_F\Delta u_F(t)^2 \tag{16a}$$

$$\text{s.t.} \quad \underline{F}_{\text{trac}} \le F_{\text{trac}}(t) \le \overline{F}_{\text{trac}}, \tag{16b}$$

$$d(t) \ge T_h v_x(t), \tag{16c}$$

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 [27], or, in the context of V2V and V2I, when there is perfect preview through communication [19]. Also, using an economic cost can help reduce fuel consumption, with minimal impact on travel time [28]. Additional applications in longitudinal vehicle dynamics that have been explored but still need to be investigated in depth are launch control and gear shifting. More recent applications involve braking control for collision avoidance systems, see, e.g., [29], again, possibly by using V2X to exploit preview information.

The interest in lateral dynamics spans multiple applications, especially lateral stability control and lane keeping, up to autonomous driving. A challenging case [5] 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 the need to consider nonlinear tire models. A viable approach is to approximate the tire model as piecewise linear, resulting in the optimal control problem

$$\min_{\Delta\delta, M_{\rm br}} \sum_{t=0}^{T_N} (\dot\psi(t) - r_\psi(t))^2 + w_\delta \Delta\delta(t)^2 + w_{br} M_{\rm br}(t)^2 \quad (17a)$$

$$\text{s.t. } |\delta(t) - \delta(t-1)| \leq \overline{\Delta\delta}, \quad |\delta(t)| \leq \overline{\delta}, \quad (17b)$$

$$(17c)$$

$$|M_{\rm br}(t)| \leq \overline{M}_{\rm br}, \quad |\alpha_j| \leq \overline{\alpha}_j, \quad (17d)$$

$$f_{c\,j}(\alpha_j) = \begin{cases} -d_j\alpha_j + e_j & \text{if} \quad \alpha_j > p_j, \\ C_j\alpha_j & \text{if} \quad |\alpha_j| \leq p_j, \\ d_j\alpha_j - e_j & \text{if} \quad \alpha_j < -p_j, \end{cases} \quad (17e)$$

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. Figure 4 shows experimental results in a slalom tests where MPC controls the slip angles to be near the peaks where the maximum tire forcs are achieved.

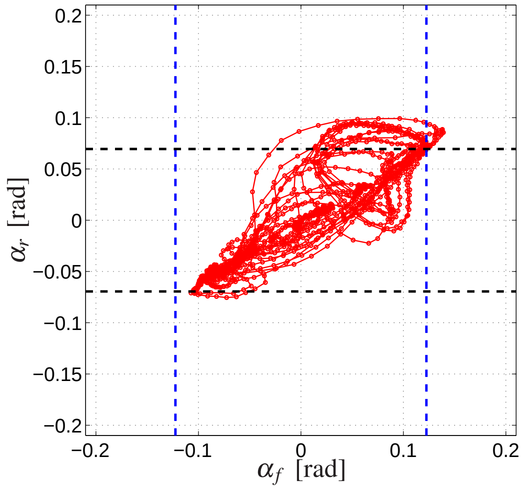


Fig. 4. Results of MPC control of lateral vehicle dynamics in a in a slalom test on snow. Phase plane plot of the tire slip angles; the dashed lines indicates the angles p_j, $j \in \{f, r\}$ where maximum force is achieved.

As for the vertical dynamics, MPC offers interesting possibilities for active suspension control when preview of the road is available [18], [30], for instance obtained from a forward looking camera.

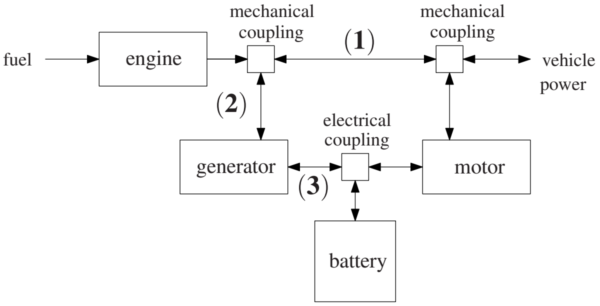


Fig. 5. Schematic of a powersplit HEV architecture. The arrows indicate the allowed power flow directions. The series HEV architecture is obtained by removing the link (1), thus the mechanical couplings are simply mechanical connections. The parallel HEV architecture is obtained by removing the generator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

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 powertrains 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 several 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

$$P_{\rm veh} = P_{\rm eng} - P_{\rm gen} + P_{\rm mot} - P_{\rm mec}^{\rm los}, \quad (18)$$

where $P_{\rm veh}$ is the vehicle power for traction, $P_{\rm eng}$ is the engine power, $P_{\rm gen}$ is the mechanical power used to generate electrical energy to be stored in the battery, $P_{\rm mot}$ is the electrical power used for traction and $P_{\rm mec}^{\rm los}$ are the mechanical power losses. In some HEV architectures, such as the powersplit architecture in Figure 5, motoring and electric energy generation can be accomplished by multiple components, since, despite the names indicating the most efficient usage, both the motor and the generator can operate in both modes.

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

$$P_{\rm bat} = P_{\rm mot} - P_{\rm gen} + P_{\rm gen}^{\rm los} + P_{\rm mot}^{\rm los}, \quad (19)$$

where $P_{\rm bat}$ is the power flowing from the battery and $P_{\rm gen}^{\rm los}$, $P_{\rm mot}^{\rm los}$ 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 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) $SoC = \frac{Q_{\text{bat}}}{Q_{\max}}$. 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 under voltage control and representing the battery as an ideal 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 control purposes may be approximated by one or two constants [17], the latter modeling different charging and discharging efficiencies.

The main novel control problem in HEV powertrains is the management of the energy in order to minimize the fuel consumption, subject to a charge sustaining constraint

$$\min \quad \int_{t_i}^{t_f} W_f(t) dt \tag{20a}$$

$$\text{s.t.} \quad SoC(t_i) = SoC(t_f), \tag{20b}$$

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 (20a), 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 (20b). The cost function can also include additional terms such as SoC reference tracking. Constraints on the various power flows and battery SoC can also be included, resulting in

$$\begin{aligned} \min_{P_{\text{bat}},\cdots} \quad & F_N(SoC(T_N)) + \sum_{t=0}^{T_N - 1} W_f(t) + w_{soc}(SoC(t) - r_{SoC}(t))^2 \\ \text{s.t.} \quad & P_{\text{eng}} - P_{\text{gen}} + P_{\text{mot}} - P_{\text{mec}}^{\text{los}} = P_{\text{drv}}, \\ & \underline{SoC} \le SoC(t) \le \overline{SoC}, \quad |P_{\text{bat}}| \le \overline{P}_{\text{bat}}. \end{aligned} \tag{21}$$

The mechanical power equation (18) is enforced as a constraint in (21) to ensure that the vehicle power is equal to the driver-requested power. The actual degrees of freedom vary with the HEV architecture. Ignoring the gear shifting strategy, which is usually separately optimized, for a powersplit architecture shown in Figure 5 there are two degrees of freedom. 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, there is one degree of freedom. Exploiting the simplicity of the latter, an MPC was developed in [17] which was deployed on a fully functional, production-prototype, series HEV that was also road driven, see Figure 6.

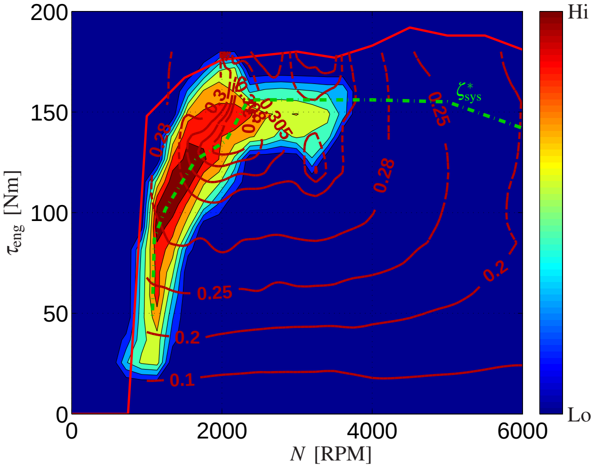


Fig. 6. Distribution of series HEV engine operating points on an experiment on UDDS cycle. Dash green line indicates maximal efficiency operating points for given power output.

In recent years several advanced MPC methods have been applied to HEV energy management, including stochastic MPC [31] where the driver-requested power is predicted using statistical models learned from data during vehicle operation [32].

IV. Aerospace Applications and Models

MPC and RTO have been extensively investigated for aerospace control applications, see for instance the survey [33], and [34] and the accompanying special issue of *AIAA Journal of Guidance, Dynamics and Control* that introduce computational guidance and control as a paradigm shift in aerospace control applications.

In aircraft, MPC has been of interest as an advanced solution for control of gas turbine engines [35], [36] as these have many constraints, including high pressure and low pressure compressor surge/stall margins, core and fan over speed limits, over temperature limits, lean combustion blowout limits, etc. MPC solutions have also been considered for integrated thrust and electrical power delivery in more electric aircraft, see e.g., [37]. One of the current frontiers for the use of MPC is maneuver and gust load alleviation for flexible aircraft [38]. In these applications, MPC solutions must accommodate high order models resulting from finite element or modal analysis, multiple control effectors, lack of full state measurement, gust disturbances, and be feasible for implementation at high sampling rates.

In spacecraft control, many RTO applications have been considered. In particular, advanced solutions for powered descent and landing based on real-time convex optimization and ideas of lossless convexification has been developed and flight tested [39]–[43]. A challenging application to combined station-keeping and momentum-management to enable future Earth-Moon halo orbit tracking missions has been considered in [44]. Next, we provide an introductory overview of some of the relevant models and of the control problems that can be addressed by MPC in such applications.

A. *Control of Spacecraft Attitude*

Spacecraft attitude control involves control of spacecraft orientation. Suppose the spacecraft orientation relative to an inertial frame is specified by 3-2-1 Euler angles ϕ (roll), θ (pitch), and ψ (yaw). Assuming the spacecraft bus fixed frame is a principal frame, the spacecraft attitude kinematics and dynamics equations are given by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \times \frac{1}{\cos(\theta)} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \tag{22a}$$

$$\begin{bmatrix} J_1\dot{\omega}_1 + (J_3 - J_2)\omega_2\omega_3 \\ J_2\dot{\omega}_2 + (J_1 - J_3)\omega_1\omega_3 \\ J_3\dot{\omega}_3 + (J_2 - J_1)\omega_1\omega_2 \end{bmatrix} = - \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix}, \tag{22b}$$

where ω_1, ω_2, ω_3 are components of angular velocity vector expressed in spacecraft body fixed frame. The state vector is six dimensional, $X = [\phi\ \theta\ \psi\ \omega_1\ \omega_2\ \omega_3]'$, and the control vector, which encompasses components of control moment, has the form, $U = \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}'$.

1) MPC opportunities in attitude control: For attitude control, MPC allows to enforce constraints, from simple torque actuator constraints $|M_i| \leq \bar{M}_i$, $i = 1,\ldots,3$, to inclusion zone constraints, $f_{\text{izn}}(\theta,\phi,\psi) \in \mathscr{I}$, modeling, for instance, requirements to maintain Line-of-Sight, and exclusion zone constraints, $f_{\text{xzn}}(\theta,\phi,\psi) \notin \mathscr{E}$, modeling, for instance, requirements to avoid exposure of instruments to high energy/radiation sources, e.g., the sun.

For applying linear MPC, the attitude dynamics can be linearized around the origin resulting in three uncoupled double integrators about each of three individual axes. Appropriate choices of the sampling period are in the order of a fraction of a second to several seconds, depending on the agility of the spacecraft and the target maneuvers. The closed-loop response of a basic MPC with a nonlinear attitude model is shown in Figure 7.

The basic design can be extended in several directions. In [45] a spacecraft with reaction wheel actuators is considered, and a special formulation of the cost with the virtual reference is exploited to expand the closed-loop domain of attraction, along with a low complexity addition of the integral action performed at the reference and outside of MPC controller, to guarantee offset-free tracking of attitude set points. A dual projected gradient algorithm which is suitable for implementation in fixed point arithmetic is applied to solve online the resulting quadratic program (QP). In another extension, [46], MPC is used to implement reaction wheel desaturation maneuvers through either gravity gradients or magnetic torquers while maintaining spacecraft pointing within specified constraints. The treatment involves modifying spacecraft model with reaction wheel states and adding gravity gradient and magnetic actuation effects.

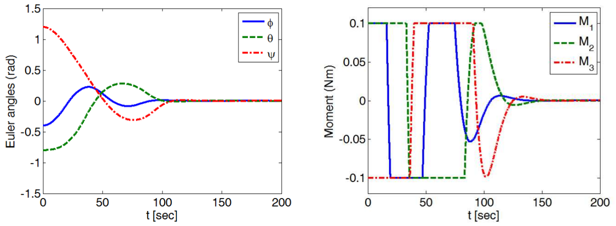


Fig. 7. The time histories of Euler angles and control moments.

Another interesting opportunity is based on the fact that a globally stabilizing control law for attitude control must be discontinuous, due to the topological properties of $SO(3)$. However, nonlinear MPC can result in discontinuous feedback policies, and in [47] it was shown that $SO(3)$-based MPC can globally stabilize the spacecraft attitude.

B. *Control of Spacecraft Relative Motion*

Spacecraft relative motion control refers to the control of spacecraft motion (translational or both translational and rotational) relative to a nominal orbital position as required by missions involving, e.g., rendezvous, docking, proximity operations, circumnavigation, orbit adjustment or maintenance. While in the past spacecraft translational and orbital transfer problems have been addressed by open loop maneuver planning and optimal control techniques, recent interest in MPC has been strongly motivated by MPC ability to deal with various constraints such as thrust limits, Line-of-Sight (LoS) cone, soft landing and obstacle avoidance, while improving robustness to modeling errors. Applications of MPC to relative motion control have been summarized in a recent tutorial [48]. Some predictive control solutions have been tested in-orbit, e.g. for PRISMA mission or developed for missions around other planets [49].

Hill-Clohessy-Wiltshire (HCW) equations prescribe the spacecraft relative coordinates in the so called Hill's (or local vertical, local horizontal LVLH) frame attached to the nominal orbital position (see Figure 8). For circular orbits the HCW equations are time-invariant and are linearized as

$$\ddot{x} - 3n^2x - 2n\dot{y} = \frac{F_x}{m_c}, \tag{23a}$$

$$\ddot{y} + 2n\dot{x} = \frac{F_y}{m_c}, \tag{23b}$$

$$\ddot{z} + n^2z = \frac{F_z}{m_c}, \tag{23c}$$

where n is the nominal orbital rate, (x,y,z) are spacecraft relative coordinates in LVLH frame, m_c is the mass of spacecraft, and F_x, F_y and F_z are thrust forces. Alternatively, if an impulsive thrust approximation is made, the thrust

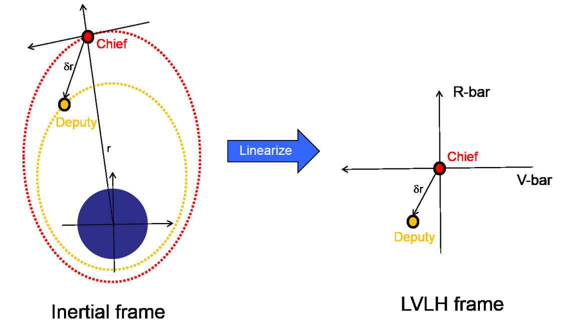


Fig. 8. Motion in ECI frame and motion in the relative (LVLH) frame.

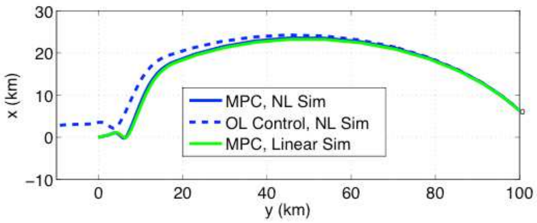


Fig. 9. Open-loop and MPC-based relative motion maneuvers.

forces can be replaced by velocity changes as inputs [11]. The in-orbital plane dynamics (x,y) are decoupled from the out of the orbital plane dynamics z, and there is an asymmetry in the dynamics between x (radial or R-bar) direction and y (in orbital track or V-bar) direction.

1) MPC opportunities in relative motion control: For spacecraft relative motion control, MPC provides increased robustness against model error and perturbances. Figure 9 illustrates how an open-loop maneuver computed based on (23) misses the target when simulated on the nonlinear model. However, MPC reaches the target by recomputing the solution in a receding horizon manner.

For spacecraft docking, the implementation of MPC benefits from decomposing the maneuver into two phases, a rendezvous phase and a docking phase with different cost functions and constraints [11]. In rendezvous phase, LoS cone constraints are not applied and, to increase the domain of attraction, the spacecraft position is controlled to a virtual set-point, the difference of which and the target is also penalized in the cost. In the docking phase, activated when the spacecraft is sufficiently close to the target, LoS, thrust direction, and soft docking constraints are imposed, and the virtual set-point is removed. The thrust constraints are imposed in both phases of the maneuver. Figure 10 illustrates a typical maneuver resulting from such a decomposition.

The requirements of more complex missions that involve avoiding debris or docking to rotating targets can be addressed by dynamically reconfiguring the constraints during the maneuver [10]. For instance, a nonconvex constraint of avoiding a debris can be handled by separating the spacecraft from a debris by a rotating hyperplane [10], [11]. This avoids the need for integer variables as in [50], and results in an optimization problem that at each stage is a convex QP.

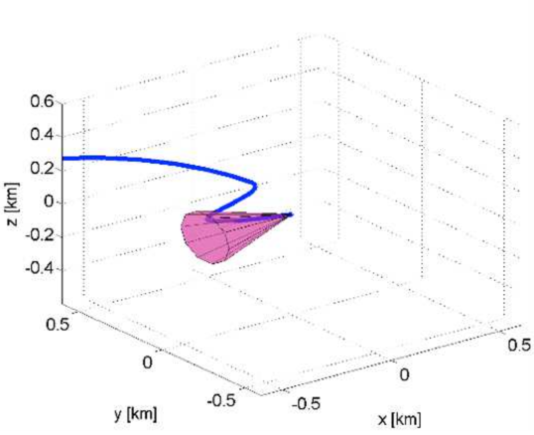


Fig. 10. Spacecraft maneuver with rendezvous and docking phases and LoS cone constraints.

The experimental testing of relative motion maneuvers is feasible in simple robotic testbeds where distance and time scaling is used to map the motion in orbital plane to the motion of the ground robots on the floor [51]. More complex test beds have been also developed such as POSEIDYN [52].

C. Spacecraft motion control near an asteroid or a comet

The spacecraft control near a small body (an asteroid or a comet) are involved due to the uncertainty of the gravitation field and other effects such as comet outgassing pressure that are difficult to characterize in advance of the mission. At the same time, given that the small body forces are relatively weak, MPC-based feedback control can be very effective in robustifying spacecraft maneuvers to the uncertainties in small body environment.

The relative motion of a spacecraft in an asteroid fixed frame is described by [53],

$$\begin{aligned}\ddot{x} &= F_{g,x} + 2n\dot{y} + n^2 x + u_x,\\ \ddot{y} &= F_{g,y} - 2n\dot{x} + n^2 y + u_y,\\ \ddot{z} &= F_{g,z} + u_z,\end{aligned}$$

where n is the rotation rate of the asteroid, $F_{g,x}$, $F_{g,y}$ and $F_{g,z}$ are the components of the gravitational force per unit mass exerted on the center-of-mass of the spacecraft by the asteroid, and u_x, u_y, and u_z are the control inputs which correspond to thrust induced accelerations. The gravity force is computed from the potential, given as a summation of spherical harmonics,

$$G = -\frac{\mu}{R_0}\sum_{l=0}^{\infty}\sum_{m=0}^{n}(C_{l,m}V_{l,m} + S_{l,m}W_{l,m}),$$

where μ is the asteroids gravitational constant, R_0 is the radius of a reference sphere, $C_{l,m}$ and $S_{l,m}$ are the Stokes coefficients, and $V_{l,m}$ and $W_{l,m}$ are functions of position calculated using the Montenbruck recursion scheme [54]. The gravity force is the gradient of the potential $[F_{g,x}\ F_{g,y}\ F_{g,z}]' = -\nabla G$. Taking $\xi = [\ x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ ]' = [r'\quad v']'$ as the relative state vector, and $u = [\ u_x\ u_y\ u_z\ ]'$ as the control vector, the equations of motion can be as affine-in-controls,

$$\dot{\xi}(t) = A_c\xi(t) + F_g(\xi(t)) + B_c u(t).$$

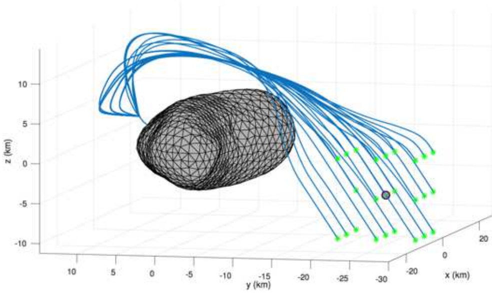


Fig. 11. Asteroid landing by nonlinear MPC from several initial conditions.

1) MPC opportunities in control near asteroids: Due to the difficulties in exactly modeling the gravity field near the small body, the robustness due to the receding horizon nature of MPC is appealing. Due to nonlinear gravity effects, a nonlinear MPC design is beneficial where the prediction model can be computed [55] by a truncation of the potential to the 4th order using a constant density ellipsoid approach [53]. Such a prediction model can be configured without extensive observations of the asteroid [55]. The overall maneuver can be partitioned into two phases, the circumnavigating phase and the landing phase. During the circumnavigation phase, MPC allows to enforce a nonlinear safety ellipsoid constraint to prevent the spacecraft from colliding with the asteroid, while the spacecraft is controlled to a waypoint close to the target landing location. During the landing phase, MPC allows to enforce a convex paraboloid and a position overshoot constraint to bound the spacecraft position, while the spacecraft is stabilized to 2 m above the asteroid surface, at which point a terminal landing phase can be initiated or a probe can be lowered. The results from simulations of asteroid landing, with added noise to position and velocity measurements/estimates, are given in Figure 11.

D. Station keeping for Low thrust satellites

Geostationary communication satellites must be maintained in a specific window around the nominal orbit position and attitude to provide reliable communication links. This operation, known as station keeping (SK), is obtained in conventional satellites by adjusting the orbit with impulsive maneuvers once every several days.

In recent years there has been a significant increase in interest in satellites with electric propulsion, which has much higher efficiency than conventional (chemical) propulsion, but can generate only a fraction of the thrust. Thus, with electric propulsion, SK must operate almost continuously, but at the same time the reduced thrust allows to use propulsion for tasks requiring finer actuation. One such example is managing the momentum of the reaction wheels used for attitude control, i.e., unloading the momentum by reducing the reaction wheels speed to avoid saturation, while compensating the induced torque disturbances via the thrusters.

1) MPC opportunities in satellites station keeping: MPC for simultaneous station keeping and momentum management in low thrust satellites has been investigated in [8], [9]. The orbital dynamics model is based on the relative motion equations (23) with respect to a virtual satellite in the nominal orbit with a nominal disturbance predictor [8]. The linearized attitude dynamics is obtained from (22), possibly in closed-loop with an attitude controller [9]. The station keeping window is enforced by the constraints

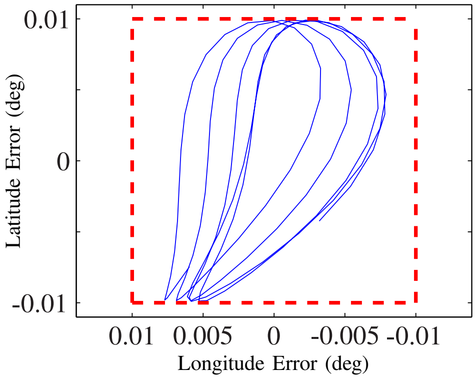


Fig. 12. Station keeping simulation of a low thrust geostationary satellite controlled by MPC over 5 orbits. SK window in dash red.

$$|y| \leq \bar{r}\tan(\lambda_{\text{long}}), \quad |z| \leq \bar{r}\tan(\lambda_{\text{lat}}), \tag{24}$$

where $\bar{r}$ is the nominal orbit altitude, and λ_{long}, λ_{lat} are the maximal allowed deviations in latitude and longitude, respectively, typically on the order of 10^{-2} degrees, see Fig. 12. While orbital and attitude dynamics are independent, they may be coupled by the propulsion system that generates both forces and torques [9]. Such coupling can be expressed as constraints on the thrust vector F_{th} that produces the spacecraft force and torque vectors, F_{sc}, T_{sc},

$$0 \leq H_{\text{th}}F_{\text{th}} \leq f_{\max}, \quad T_{\text{sc}} = D_T F_{\text{th}}, \quad F_{\text{sc}} = D_F F_{\text{th}}. \tag{25}$$

where D_T, D_F, H_{th}, are matrices related to the geometry of the propulsion system and $f_{\max}$ is the vector of maximum thrust. The MPC cost function can be formulated to optimize concurrently the objectives of fuel consumption and reaction wheels speed management.

V. MPC Design Process in Automotive and Aerospace Applications

This section aims at providing guidelines for a design process for MPC in A&A applications. While not a standard, it has been applied by the authors in multiple designs several of which were experimentally tested, and it has been proved useful and effective. We focus on linear MPC, because this has been so far the main method in A&A applications, primarily due to computational and implementation requirements. However, the general concepts can be applied also to nonlinear MPC with relatively few modifications.

We consider the finite horizon optimal control problem

$$\min_{U(t)} \; x'_{N|t}Px_{N|t} + \sum_{k=0}^{N-1} z'_{k|t}Qz_{k|t} + u'_{k|t}Ru_{k|t} \tag{26a}$$

$$\text{s.t. } x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \tag{26b}$$

$$y_{k|t} = Cx_{k|t} + Du_{k|t}, \tag{26c}$$

$$z_{k|t} = Ex_{k|t}, \tag{26d}$$

$$u_{k|t} = \kappa_f x_{k|t}, \quad k = N_u, \dots, N-1, \tag{26e}$$

$$x_{0|t} = x(t), \tag{26f}$$

$$\underline{y} \le y_{k|t} \le \overline{y}, \quad k = N_i, \dots N_{cy}, \tag{26g}$$

$$\underline{u} \le u_{k|t} \le \overline{u}, \quad k = 0, \dots N_{cu}-1, \tag{26h}$$

$$H_N x_{N|t} \le K_N, \tag{26i}$$

where the notation $k|t$ denotes the k-steps prediction from measurements at time t, $U(t) = \{u_{0|t} \dots u_{N-1|t}\}$, 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, 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.

The MPC design amounts to properly constructing components of the finite horizon optimal control problem to achieve the desired specifications and control-oriented properties, as we discuss next.

A. Prediction Model

Several A&A processes are well studied and have readily available models, some of which have been described in Sections III, IV. In MPC design it is desirable to start from such physics oriented models. However many of them may be of unnecessarily high order, may be nonlinear, and may require estimation of several parameters. Hence, usually the first step in MPC design is to refine the physics based model by:

- simplifying the model to capture the relevant dynamics for the specific application and controller requirements, 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 for use in a discrete-time prediction model.

Even for the simple case of idle speed control, in [2] due to computational requirements, the powertrain model (1), (4), (5) is linearized around the nominal idle operating point, with the models for the delays removed during identification, to be added again later. The model structure is known from physics and the parameters are identified from step responses.

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

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

$$y_m(t) = C_m x_m(t) + D_m u_m(t), \tag{27b}$$

$$z_m(t) = E_m x_m(t), \tag{27c}$$

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 (27) 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. Further modifications may be made to account for additional information available in the system, such as preview on disturbances or references, or known disturbance models. 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 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 the last value in the buffer is to be held constant or set to 0. An exact linear model of the delay buffer can be formulated only in discrete-time, 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". In some applications, it may be difficult, to compute the equilibrium associated with a certain reference value r due to the uncertainty of 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) model form [23], where both the state and input are differentiated, and the tracking error e_m is included as an additional state,

$$\Delta x_m(t+1) = A_m \Delta x_m(t) + B_m \Delta u_m(t), \tag{28a}$$

$$e_m(t) = e_m + E_m \Delta x_m(t) + \Delta r(t), \tag{28b}$$

$$y_m(t) = y_m(t-1) + C_m \Delta x_m(t) + D_m \Delta u_m(t), \tag{28c}$$

where Δr is the change in reference value. For MPC applications one may need to add integrators to compute y_m from

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.

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 A&A applications are summarized in Table II. Applying all the augmentations result in a higher order prediction model (26b)–(26d),

$$x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \quad x = \begin{bmatrix} x_m \\ x_p \end{bmatrix}, \quad u = \begin{bmatrix} u_m \\ u_p \end{bmatrix}, \tag{29a}$$

$$y_{k|t} = Cx_{k|t} + Du_{k|t}, \quad y = \begin{bmatrix} y_m \\ y_p \end{bmatrix}, \tag{29b}$$

$$z_{k|t} = Ex_{k|t}, \quad z = \begin{bmatrix} z_m \\ z_p \end{bmatrix}, \tag{29c}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$, $z \in \mathbb{R}^q$ are the prediction model state, input, constrained outputs, and performance output vectors, x_p, u_p, y_p, z_p are the ancillary 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 enforcement of the constraints directly on the states is certainly possible, it is more convenient to introduce the constrained output vector y specifically for this use, which allows to enforce state, mixed state-inputs, and also pure input constraints through a single vector. Thus, the constraints may be formulated as

$$y_{k|t} \in \mathscr{Y}_m, \quad u_{k|t} \in \mathscr{U}_m, \tag{30}$$

where and $\mathscr{Y}_m$ and $\mathscr{U}_m$ are the admissible sets for constrained output and input vectors, respectively. Enforcing constraints on the constrained output and input vectors of the prediction model, which include augmentation, usually allows to formulate (30) as simple bounds in (26g), (26h),

$$\underline{y} \leq y_{k|t} \leq \overline{y}, \quad \underline{u} \leq u_{k|t} \leq \overline{u}, \tag{31}$$

which are easier to specify and handle in the optimization.

In A&A applications, the sampling period T_s of the prediction model (29) is often equal to the period of the control cycle for the function being developed. However, for (29) to be accurate, T_s should be small enough to allow for 3-10 steps in the settling of the fastest dynamics, following a step response. 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×-3×, than the number of steps for the settling of the slowest (stable) prediction model dynamics, following a step response. 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. More correctly, 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.

Other horizons can be considered, for adjusting the computational requirements in solving the MPC problem. 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 no longer an optimization variable, but rather assigned by the pre-defined terminal controller (26e),

$$u_{k|t} = \kappa_f x_{k|t}, \; k = N_u, \ldots, N-1. \tag{32}$$

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

$$\underline{y} \leq y_{k|t} \leq \overline{y}, \; k = N_i, \ldots N_{cy}, \; \underline{u} \leq u_{k|t} \leq \overline{u}, \; k = 0, \ldots N_{cu}-1, \tag{33}$$

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, $n_c = 2(p(N_{cy}+1-N_i)+mN_{cu})$. This determines the size of the optimization problem. Move blocking, which allows the input to change only at a subset of the sampling instants in the prediction horizon, can also be used to reduce the number of decision variables.

C. Cost Function, Terminal Set and Soft Constraints

The cost function determines the objectives of MPC, and their priority. To construct that, the control specifications are converted into variables that are to be controlled to 0. Such variables are either a part of the process model (27), or are included in the prediction model (29) by the augmentations in Section V-A. In general the MPC cost function (26a) is

$$J_{N|t} = x'_{N|t} P x_{N|t} + \sum_{k=0}^{N-1} z'_{k|t} Q z_{k|t} + u'_{k|t} R u_{k|t}, \tag{34}$$

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 plant input rate of change

TABLE III

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

where the performance outputs of the prediction model $z = Ex$ can model objectives such as tracking $z = Cx - C_r x_r$, where x_r is the reference model state, and $r = C_r x_r$ is the current reference. In (34), $Q \geq 0$, $R > 0$ are the matrix weights that determine the importance of the different objectives: for a diagonal weight matrix Q, the larger the i^{th} component, the faster the i^{th} performance output 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 mapping tends to be application dependent, some of the common cases 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 (34), $P \geq 0$ 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 A&A applications is to choose P to be the solution of the Riccati equation, constructed from A, B in the prediction model (29), and Q, R in the cost function (34),

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

This method can be used, after some modifications, also for output tracking [56]. Alternative approaches are based on the solution of the Lyapunov equation for systems that are asymptotically stable, or on the solution of an LMI to determine a stabilizing controller and the corresponding closed-loop Lyapunov function. 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 (26i) to guarantee recursive feasibility and stability in the feasible domain of the MPC optimization problem has seen a limited use in A&A applications. This is mainly 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. Instead, for ensuring that the optimization problem admits a solution, constraint softening is often applied. A QP with soft constraints can be formulated as

$$\min_{v,s} \quad \frac{1}{2}v'Hv + \rho s^2 \tag{35a}$$

$$\text{s.t.} \quad Hv \leq K + Ms, \tag{35b}$$

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.

While in general one may want to have ρ extremely large to better emulate hard constraints, this may end up worsening the problem conditioning with the result that the solution time increases, and the solution quality may also decrease.

VI. COMPUTATIONS AND NUMERICAL ALGORITHMS

As discussed in Section II, a key challenge for implementing MPC is accommodating its significantly larger computational footprint when compared to standard controllers, i.e., PID. As MPC is based on a solution to 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 integrator 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 considering a controller that evaluates a function, to a controller that executes an algorithm. Furthermore, cost is a key driver in the automotive and aerospace development. 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 computational hardware, rather than assume that computer hardware will become available that is able to execute it.

The common practice found in many research papers of extrapolating the real-time behavior of MPC in an automotive/spacecraft micro-controller unit (aMCU/sMCU) from the one that is seen in a desktop CPU (dCPU) is potentially very misleading as xMCUs and dCPUs have significantly different capabilities, see Table IV. First, one needs to consider that powerful xMCUs 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

	Freq.	**Instr./s**	**RAM**	**ROM**
dCPU	1000s MHz	100s GIPS	10s GB	1000s GB
aMCU	100s MHz	1000s MIPS	1000s kB	10s MB
sMCU	1000s MHz	10s GIPS	100s MB	1000s MB

TABLE IV
COMPARISON OF COMMON CHARACTERISTICS FOR DESKTOP PROCESSORS (DCPU), AUTOMOTIVE (AMCU) AND SPACECRAFT (SMCU) MICRO-CONTROLLERS.

single controller is only a fraction of what is available in the entire xMCU. The difference between instruction sets (RISC vs CISC) and the simpler structure of the xMCU memory architecture results in a significantly different numbers of instructions per seconds (IPS) for the xMCUs with respect to dCPUs.

For aMCUs, most of the differences are due to the need to work in extreme environments, e.g., temperature range between -40^{o}C and $+50^{o}$C, in which a dCPU is not required to operate, and may be even prevented from starting in such conditions. 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 is actually the bottleneck [57].

For sMCUs, while more powerful computing hardware is permissible for some aerospace systems, such as communication satellites, the considerations of weight and electrical power consumption often restrict the available onboard computing power. Such weight constraints are particularly severe for smaller electric battery powered vehicles, e.g., flapping wing aircraft or smaller UAVs. The restrictions due to the need to operate in harsh environments become even more significant in aerospace applications, due to temperature that can range between -200^{o}C and $+400^{o}$C degrees, and presence of harmful radiations requiring specific processes of radiation-hardening.

Because of this, and since also processor-specific code optimization by the engineers or by the custom compilers, may play a very significant role, the evaluation of the computational load of an MPC controller in an xMCU 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 xMCU, and using that to estimate the execution time in xMCU.

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 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 vehicles. Some examples, tested in fully functional and road-drivable (and in several cases road-driven) vehicles are in [2], [3], [5]–[7], [17], [26], [58]–[60]. For spacecraft attitude control, due to decoupled form and symmetry of (22b), explicit MPC is advantageous in terms of memory and computations, and in [10] explicit MPC for relative motion was developed.

In explicit MPC, the optimizer of (26) is obtained by evaluating a pre-computed piecewise affine function of the prediction model state (29),

$$u_{\rm mpc}(x) = \begin{cases} F_1 x + G_1 & \text{if} \quad H_1 x \leq K_1, \\ & \vdots \\ F_s x + G_s & \text{if} \quad H_s x \leq K_s, \end{cases} \tag{36}$$

where s is the total number of regions.

The main advantages of explicit MPC and the reasons for its appeal in A&A applications are:

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

Additional benefits include the possibility of computing explicit MPC, with few modifications, also for switched and hybrid systems, which allowed for the application of switched and hybrid MPC in automotive control [5]–[7], [26], [58], [59], the possibility of constructing the closed-loop dynamics and hence to analyze its local and global stability, and the possibility of storing only the most useful regions, while resorting to 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,

$$n_{as} \leq \sum_{h=0}^{\min\{n_c, n_v\}} \left(\begin{array}{c} n_c \\ h \end{array} \right), \tag{37}$$

where n_v is the number of variables, and n_c is the number of constraints. Because of (37), explicit MPC is limited to applications with a relatively short horizon, few control inputs, and few constraints. Another limitation is that the construction of (36) is often too complex to be implemented in micro-controllers, and hence explicit MPC cannot be easily tuned or changed after deployment.

B. Online MPC

Due to (37), In problems with many control inputs, long prediction horizons and many constraints, explicit MPC is be too complex to compute or to store for real-time usage. Also, if the prediction model changes over time, it is very difficult to adjust the explicit solution, while it is relatively simple to update the data of the optimal control problem. In these cases, the online solution of (26) may be preferable.

Among such cases, in [4], a QP solver and a nonlinear solver were used online for control of autonomous vehicles, and this enabled using a long horizon to take maximum advantage of the known reference trajectory. The nonlinear solver was based on sequential quadratic programming (SQP), and both the nonlinear and QP solvers used active-set methods. In [61] an active-set solver was used to solve QPs for controlling manifold pressure and airflow in a diesel engine using EGR and VGT. The solver was chosen 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 [62], 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. 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 significant amount of both data and execution memory, and achieving portability of the libraries to micro-controllers may not be straightforward. Alternatively, first-order methods may have slower convergence, but have much simpler code and hence require less memory and do not require additional 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 projection onto the feasible set $\mathscr{F}$

$$\hat{z}_s^{(h+1)} = \hat{z}_s^{(h)} - \alpha_s(z_s^{(h)}) \cdot h_s \left(\frac{d}{dz_s} J(z_s) \Big|_{z_s^{(h)}} \right), \quad (38a)$$

$$z_s^{(h+1)} = \mathrm{proj}_{\mathscr{F}} \left(\hat{z}_s^{(h+1)} \right), \quad (38b)$$

where z_s may contain additional variables with respect to those in 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 first-order methods for MPC have been proposed, based on Nesterov's fast gradient algorithm [63], [64], Lagrangian methods [65], nonnegative least squares [66], and alternating direction method of multipliers [67]–[69]. For instance, in [70] the method in [66] was used for vehicle trajectory tracking.

Another advantage of first order methods is that, due to their simple structure, it is often possible to derive a worst case number of iteration [63], [64]. This allows to "size" the processor to the optimization algorithm. On the other hand in many real A&A applications, the processor is already fixed at the time the control algorithm is designed. Hence, it may be even more effective to use algorithms that allow for early termination while still providing some guarantees, at least with respect to constraint satisfaction, such as, for instance, primal active set methods [71].

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 could have been realistically implemented in most A&A systems. Nonlinear MPC (NMPC) is significantly more complex, which is to a large extent due to the algorithm for solving the nonlinear programming problem. For instance, in [4] 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, [72] 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.

For spacecraft control near an asteroid, in [55] the nonlinear optimization problem in NMPC, is solved by a nonlinear primal dual filter-line search interior point solver. Sensitivity based warm-starts are used to reduce the computational burden of the algorithm by taking advantage of information from previous solutions, which is based on the receding horizon nature of MPC. It shall be noted however that, if the previous information are outdated, warm-starting may end up increasing the computational burden, probably more so in interior point than in active set methods.

Some applications of NMPC to automotive systems [73]–[75] are based on the C/GMRES method in [76]. 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. The changes to the active sets may cause discontinuities in the dual variables, and sometimes also in the primal variables, see, e.g., [77], which is in conflict with the smooth update of the continuation method. Various inequality constraint enforcement techniques for diesel engines in the context of [76] are compared in [78].

More recently [79]–[81], some applications are being investigated based on the Real-Time Iteration (RTI) scheme [82], which is based on combining effective integrators for multiple-shooting methods with SQP, where usually only one step of optimization is actually performed.

Expanding the reliability and reducing the computational burden of NMPC will probably be a key effort in the upcoming years to allow for significant use in A&A applications.

VII. Conclusions

RTO and MPC are appealing for aerospace and automotive (A&A) applications that increasingly require high performance multivariable control under stringent constraints and can benefit from preview information. First principles model with parameters identified from data and augmented with auxiliary states are used to construct the prediction model. Such model, the cost function and the constraints are used to formulate an optimal control problem that enforces the application specifications. While several success stories have been reported, challenges to broader practical use and production deployment of MPC in A&A remain, e.g., in the numerical algorithms and their computational load, in the reliability of the approximated models, in the calibration process, leaving significant room for further research efforts.

References

[1] D. Hrovat, "MPC-based idle speed control for IC engine," in *Proc. of FISITA conference*, Prague, Czech Rep., 1996.

[2] S. Di Cairano, D. Yanakiev, A. Bemporad, I. V. Kolmanovsky, and D. Hrovat, "Model predictive idle speed control: Design, analysis, and experimental evaluation," *IEEE Trans. Contr. Systems Technology*, vol. 20, no. 1, pp. 84–97, 2012.

[3] S. Di Cairano, J. Doering, I. V. Kolmanovsky, and D. Hrovat, "Model predictive control of engine speed during vehicle deceleration," *IEEE Trans. Contr. Systems Technology*, vol. 22, no. 6, pp. 2205–2217, 2014.

[4] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Contr. Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.

[5] S. Di Cairano, H. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Trans. Contr. Systems Technology*, vol. 21, no. 4, pp. 1236–1248, 2013.

[6] P. Ortner and L. del Re, "Predictive Control of a Diesel Engine Air Path," *IEEE Trans. Contr. Systems Technology*, vol. 15, no. 3, pp. 449–456, 2007.

[7] G. Stewart and F. Borrelli, "A Model Predictive Control Framework for Industrial Turbodiesel Engine Control," in *Proc. 47th IEEE Conf. Dec. and Control*, Cancun, Mexico, Dec 2008, pp. 5704–5711.

[8] A. Weiss, U. Kalabić, and S. Di Cairano, "Model predictive control for simultaneous station keeping and momentum management of low-thrust satellites," in *Proc. American Contr. Conf.*, 2015.

[9] A. Walsh, S. Di Cairano, and A. Weiss, "MPC for coupled station keeping, attitude control, and momentum management of low-thrust geostationary satellites," in *Proc. American Contr. Conf.*

[10] S. Di Cairano, H. Park, and I. Kolmanovsky, "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *Int. J. Rob. Nonlin. Control*, vol. 22, no. 12, pp. 1398–1427, 2012.

[11] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," *IEEE Trans. Contr. Systems Technology*, vol. 23, no. 4, pp. 1638–1647, 2015.

[12] N. Giorgetti, A. Bemporad, E. H. Tseng, and D. Hrovat, "Hybrid model predictive control application towards optimal semi-active suspension," *Int. J. Control*, vol. 79, no. 5, pp. 521–533, 2006.

[13] M. Canale, M. Milanese, and C. Novara, "Semi-active suspension control using fast model-predictive techniques," *IEEE Trans. Contr. Systems Technology*, vol. 14, no. 6, pp. 1034–1046, 2006.

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

[15] H. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, I. V. Kolmanovsky, and S. Di Cairano, "MPC-based energy management of a power-split hybrid electric vehicle," *IEEE Trans. Contr. Systems Technology*, vol. 20, no. 3, pp. 593–603, 2012.

[16] F. Yan, J. Wang, and K. Huang, "Hybrid electric vehicle model predictive control torque-split strategy incorporating engine transient characteristics," *IEEE Trans. Vehicular Tech.*, vol. 61, no. 6, pp. 2458–2467, 2012.

[17] S. Di Cairano, W. Liang, I. V. Kolmanovsky, M. L. Kuang, and A. M. Phillips, "Power smoothing energy management and its application to a series hybrid powertrain," *IEEE Trans. Contr. Systems Technology*, vol. 21, no. 6, pp. 2091–2103, 2013.

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

[19] T. Stanger and L. del Re, "A model predictive cooperative adaptive cruise control approach," in *Proc. American Contr. Conf.*

[20] S. Di Cairano, "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.

[21] ——, "An industry perspective on MPC in large volumes applications: Potential benefits and open challenges," in *4th IFAC Symp. Nonlinear Model Predictive Control*, 2012, pp. 52–59.

[22] J. B. Heywood, *Internal Combustion Engine Fundamentals*. New York, USA: McGraw-Hill, 1988.

[23] M. Huang, K. Zaseck, K. Butts, and I. Kolmanovsky, "Rate-based model predictive controller for diesel engine air path: Design and experimental evaluation," *IEEE Trans. Contr. Systems Technology*, vol. 24, no. 6, pp. 1922–1935, 2016.

[24] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] R. N. Jazar, *Vehicle dynamics: theory and application*. Springer Science & Business Media, 2013.

[26] G. Naus, J. Ploeg, M. Van de Molengraft, M. Heemels, and M. Steinbuch, "Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach," *Control Eng. Practice*, vol. 18, no. 8, pp. 882–892, 2010.

[27] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *Proc. 49th IEEE Conf. Dec. and Control*, 2010, pp. 6077–6082.

[28] E. Ozatay, S. Onori, J. Wollaeger, U. Ozguner, G. Rizzoni, D. Filev, J. Michelini, and S. Di Cairano, "Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution," *IEEE Trans. Intelligent Transp. Systems*, vol. 15, no. 6, pp. 2491–2505, 2014.

[29] L. Makarem and D. Gillet, "Model predictive coordination of autonomous vehicles crossing intersections," in *16th Int. IEEE Conf. Intell. Transportation Systems*, 2013, pp. 1799–1804.

[30] R. K. Mehra, J. N. Amin, K. J. Hedrick, C. Osorio, and S. Gopalasamy, "Active suspension using preview information and model predictive control," in *Proc. IEEE Conf. Control Applications*, 1997, pp. 860–865.

[31] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky, "A stochastic model predictive control approach for series hybrid electric vehicle power management," in *Proc. American Contr. Conf.*, 2010, pp. 5844–5849.

[32] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Trans. Contr. Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2014.

[33] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, "Model predictive control in aerospace systems: Current state and opportunities," *J. Guidance, Control, and Dynamics*, 2017.

[34] P. Lu, "Introducing computational guidance and control," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, p. 193, 2017.

[35] J. Fuller, D. Seto, and R. Meisner, "Optimization-based control for flight vehicles," in *AIAA Guidance, Navigation, and Control Conference*, 2000, pp. 1–11.

[36] J. W. Fuller, "Constrained dynamic inversion control and its application to turbomachinery," in *SAE Power Systems Conference, SAE Technical Paper 2010-01-1737*, 2010.

[37] J. Seok, I. Kolmanovsky, and A. Girard, "Coordinated model predictive

control of aircraft gas turbine engine and power system," *J. Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2538–2555, 2017.

[38] H.-G. Giesseler, M. Kopf, P. Varutti, T. Faulwasser, and R. Findeisen, "Model predictive control for gust load alleviation," in *4th IFAC Symp. Nonlinear Model Predictive Control*, 2012, pp. 27–32.

[39] L. Blackmore, "Autonomous precision landing of space rockets," *National Academy of Engineering, The Bridge on Frontiers of Engineering*, vol. 4, no. 46, pp. 15–20, 2016.

[40] B. Açıkmeşe and S. R. Ploen, "Convex programming approach to powered descent guidance for Mars landing," *J. Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

[41] B. Açıkmeşe, J. M. Carson, and L. Blackmore, "Lossless convexification of the soft landing optimal control problem with non-convex control bound and pointing constraints," *IEEE Trans. Contr. Systems Technology*, vol. 21, no. 6, pp. 2104–2113, 2013.

[42] D. Dueri, B. Açıkmeşe, D. P. Scharf, and M. W. Harris, "Customized real-time interior-point methods for onboard powered descent guidance," *J. Guidance, Control, and Dynamics*, no. 40, pp. 197–212, 2017.

[43] D. P. Scharf, D. Dueri, B. Açıkmeşe, J. Benito, and J. Casoliva, "Flight testing of real-time convex optimization based guidance algorithm G-FOLD - guidance for fuel optimal large divert," *J. Guidance, Control, and Dynamics*, no. 40, pp. 213–229, 2016.

[44] U. Kalabic, A. Weiss, S. Di Cairano, and I. Kolmanovsky, "Stationkeeping and momentum-management on halo orbits around L2: Linear-quadratic feedback and model predictive control approaches," *Adv. in the Astronautical Sciences*, vol. 155, pp. AAS 15–307, 2015.

[45] A. Guiggiani, I. Kolmanovsky, P. Patrinos, and A. Bemporad, "Fixed-point constrained model predictive control of spacecraft attitude," in *Proc. American Contr. Conf.*, 2015, pp. 2317–2322.

[46] ——, "Constrained model predictive control of spacecraft attitude with reaction wheels desaturation," in *Proc. European Control Conf.*

[47] U. V. Kalabić, R. Gupta, S. Di Cairano, A. M. Bloch, and I. V. Kolmanovsky, "MPC on manifolds with an application to the control of spacecraft attitude on SO(3)," *Automatica*, vol. 76, pp. 293–300, 2017.

[48] E. N. Hartley, "A tutorial on model predictive control for spacecraft rendezvous," in *European Control Conf.*, 2015, pp. 1355–1361.

[49] M. Saponara, V. Barrena, A. Bemporad, E. Hartley, J. M. Maciejowski, A. Richards, A. Tramutola, and P. Trodden, "Model predictive control application to spacecraft rendezvous in mars sample return scenario," *Prog. Flight Dynamics, GNC, and Avionics*, vol. 6, pp. 137–158, 2013.

[50] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *J. Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002.

[51] A. Goodyear, C. Petersen, J. Pierre, C. Zagaris, M. Baldwin, and I. Kolmanovsky, "Hardware implementation of model predictive control for relative motion maneuvering," in *Proc. American Contr. Conf.*, 2015, pp. 2311–2316.

[52] J. Virgili-Llop, C. Zagaris, H. Park, R. Zappulla II, and M. Romano, "Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers," *CEAS Space Journal*, 2017.

[53] D. J. Scheeres, *Orbital Motion in Strongly Perturbed Environments: Applications to Asteroid, Comet and Planetary Satellite Orbiters*. Springer, 2012.

[54] O. Montenbruck and E. Gill, *Satellite orbits: models, methods and applications*. Springer Science & Business Media, 2012.

[55] D. Liao-McAlester, W. Dunham, and I. Kolmanovsky, "Model predictive control strategies for constrained soft landing on an asteroid," in *AIAA/AAS Astrodynamics Spec. Conf.*, 2016, aIAA Paper 2016-5507.

[56] S. Di Cairano, C. A. Pascucci, and A. Bemporad, "The rendezvous dynamics under linear quadratic optimal control," in *Proc. 51st IEEE Conf. Dec. and Control*, 2012, pp. 6554–6559.

[57] L. Yu, A. Goldsmith, and S. Di Cairano, "Efficient convex optimization on gpus for embedded model predictive control," in *Proc. ACM General Purpose GPUs*, 2017, pp. 12–21.

[58] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, "An MPC/hybrid system approach to traction control," *IEEE Trans. Contr. Systems Technology*, vol. 14, no. 3, pp. 541–552, 2006.

[59] A. C. Van Der Heijden, A. F. A. Serrarens, M. K. Camlibel, and H. Nijmeijer, "Hybrid optimal control of dry clutch engagement," *Int. Journal of Control*, vol. 80, no. 11, pp. 1717–1728, 2007.

[60] D. Zhao, C. Liu, R. Stobart, J. Deng, E. Winward, and G. Dong, "An explicit model predictive control framework for turbocharged diesel engines," *IEEE Trans. Ind. Electronics*, vol. 61, no. 7, pp. 3540–3552, 2014.

[61] H. J. Ferreau, P. Ortner, P. Langthaler, L. Del Re, and M. Diehl, "Predictive control of a real-world diesel engine using an extended online active set strategy," *Annual Reviews in Control*, vol. 31, no. 2, pp. 293–301, 2007.

[62] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans. Contr. Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2013.

[63] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Trans. Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2012.

[64] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Trans. Automatic Control*, vol. 59, no. 1, pp. 18–33, 2014.

[65] M. Kögel and R. Findeisen, "Fast predictive control of linear systems combining nesterov's gradient method and the method of multipliers," in *Proc. 50th IEEE Conf. Dec. and Control*. IEEE, 2011, pp. 501–506.

[66] S. Di Cairano, M. Brand, and S. A. Bortoff, "Projection-free parallel quadratic programming for linear model predictive control," *Int. J. Control*, vol. 86, no. 8, pp. 1367–1385, 2013.

[67] P. Giselsson and S. Boyd, "Linear convergence and metric selection for douglas-rachford splitting and admm," *IEEE Trans. Automatic Control*, vol. 62, no. 2, pp. 532–544, 2017.

[68] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems," *IEEE Trans. Automatic Control*, vol. 60, no. 3, pp. 644–658, 2015.

[69] A. U. Raghunathan and S. Di Cairano, "Infeasibility detection in alternating direction method of multipliers for convex quadratic programs," in *Proc. 53rd IEEE Conf. Dec. and Control*, 2014, pp. 5819–5824.

[70] S. Di Cairano, U. Kalabić, and K. Berntorp, "Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds," in *Proc. 55th IEEE Conf. Dec. and Control*, 2016.

[71] R. Quirynen, A. Knyazev, and S. Di Cairano, "Block structured preconditioning within an active-set method for real-time optimal control," in *Proc. European Control Conf.*, 2018.

[72] M. Herceg, T. Raff, R. Findeisen, and F. Allgowe, "Nonlinear model predictive control of a turbocharged diesel engine," in *IEEE Int. Conf. on Control Applications*, 2006, pp. 2766–2771.

[73] Y. Kaijiang, M. Mukai, and T. Kawabe, "Performance of an eco-driving nonlinear MPC system for a power-split hev during car following," *SICE Journal of Control, Measurement, and System Integration*, vol. 7, no. 1, pp. 55–62, 2014.

[74] K. Yu, M. Mukai, and T. Kawabe, "A battery management system using nonlinear model predictive control for a hybrid electric vehicle," in *IFAC Symp. Adv. on Automotive Control*, 2013, pp. 301–306.

[75] D. Gagliardi, T. Othsuka, and L. del Re, "Direct c/gmres control of the air path of a diesel engine," in *Proc. 19th IFAC World Congress*, 2014, pp. 3000–3005.

[76] T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.

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

[78] M. Huang, H. Nakada, K. Butts, and I. Kolmanovsky, "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*, 2015, pp. 372–379.

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

[80] T. Albin, D. Ritter, D. Abel, N. Liberda, R. Quirynen, and M. Diehl, "Nonlinear MPC for a two-stage turbocharged gasoline engine airpath," in *Proc. 54th IEEE Conf. Dec. and Control*, 2015, pp. 849–856.

[81] R. Quirynen, K. Berntorp, and S. Di Cairano, "Embedded optimization algorithms for steering in autonomous vehicles based on nonlinear model predictive control," in *Proc. American Contr. Conf.*, 2018.

[82] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.