Command Governor Approach to Maneuver Limiting in Fighter Aircraft

Daniel Simon, Ola Harkegard and Johan Löfberg

Tweet

LINKÖPINGS UNIVERSITET

Command Governor Approach to Maneuver Limiting in Fighter Aircraft

Daniel Simon*

Linköping University, Linköping, Sweden

Dr. Ola Härkegård†

Saab Aeronautics, Linköping, Sweden

Dr. Johan Löfberg‡

Linköping University, Linköping, Sweden

I.   Introduction

Modern fighter aircraft require maximum control performance in order to have the upper hand in a dogfight or when they have to outmaneuver an enemy missile. Therefore pilots must be able to maneuver the aircraft very close to the limit of what it is capable of while at the same time focus on the tactical tasks of the mission. To enable this, modern flight control systems have automatic systems for angle of attack and load factor limiting.

These types of systems can utilize predictions of the aircraft response to pilot inputs and alter the properties of the closed loop system to minimize the predicted overshoot. Two such design techniques are model predictive control and reference and command governors. Model predictive controllers are most often used as inner loop feedback controllers which alter the control signal as function of the predicted output while reference and command governors are applied in an outer feedback loop around a nominal controller. There can be several benefits from using reference and command governors compared to model predictive controllers. First, the governors can be used as add-ons to existing legacy controllers so there is no need to redo the complete design. Furthermore the nominal inner loop controller can be tuned to achieve good performance in the nominal case, e.g., use nonlinear feedbacks to linearize the closed loop system, and the governor focus on the maneuver limiting task. It also gives a good modularity such that one can replace parts of the control system without the need to redo all of the design. Last but not least from a flight safety perspective it might be easier to certify optimization algorithms running in an outer loop which can be turned off in case of failures without affecting stability.

While model predictive controllers have been extensively investigated for flight control applications [1–28] most of them consider reconfigurable flight control systems and only few focus on envelope protection and maneuver limiting [7, 13, 17, 21]. Even though reference governors have been subject to research for quite some time very little research has been performed on applying reference and command governors to flight control design and maneuver limiting [23, 29–33]. Most of these papers consider simplified conditions with only a single linear or nonlinear system and no complex simulation environments.

In the papers by Petersen et al. [23] and Zinnecker et al. [29] the authors apply reference governors to the control of hypersonic vehicle. In the paper by Zinnecker the focus is mainly on input constraints. Kolmanovsky and Kahveci [30] uses a reference governor to handle control actuator limitations of a UAV glider and compare this to an adaptive anti-windup scheme and in the paper by Martino [31] the author investigates command governors for handling amplitude and rate constraints on a small commercial aircraft. The authors, Ye et al. [32], investigate reference governors for maneuver limiting in high angle of attack maneuvers. They investigate and compare static and dynamic reference governors with a reference governor structure based on a step response model of the closed loop system. The different reference governors are evaluated on a linear aircraft model and the conclusion is that the dynamic reference governor performs much better than the static reference governor but it has a complex maximal admissible set (we will explain the details of reference governors in Section III). The governor structure with a step response model has comparable performance with the dynamic reference governor but the complex admissible set is replaced with a finite horizon approximation which is much simpler to implement. The authors of [33] thoroughly investigates a robust command governor approach for constrained control of aircraft. They apply the robust command governor to one fighter aircraft and one small commercial aircraft with both input and output constraints. The results are promising but the simulations are only done in one envelope point and thus lacks the added complexity of changing dynamics over the envelope.

The drawback with robust command governor designs is that there must be a margin to the constraint at all times to account for disturbances. In our case, when we want to use the command governor to achieve maneuver limiting, this is not an acceptable solution. Instead we need to consider soft constraints, achieved using slack variables. An example of a command governor structure utilising soft contraints via slack variables are given in [34]. Here the authors utilize the slack variables and their weights to prioritise between various important constraints.

An interesting approach to command governor structure is proposed in [35]. The authors considers only second-order systems with input time delays and show that for these systems it is sufficient to consider only four distinct time points per output in the prediction horizon to guarantee constraint satisfaction. This reduces the computational complexity of the online solution of the resulting quadratic program, however it requires that the reference is kept constant over the prediction horizon.

In this paper we extend the previous research and go beyond what has been done before. We implement and analyze command governors for maneuver limiting in the most complex aircraft models available at Saab Aeronautics, surpassed only by real flight testing. The design is made for an aircraft model with nonlinear aerodynamics that also vary with speed and altitude and the implementation is done in Saab Aeronautics development simulator for the JAS 39 Gripen fighter aircraft. Due to the changing dynamics of the aircraft over the flight envelope, we can not use the classical structure of the command governor. Instead we have to adopt command governor structure much like the reference governor based on a step response model [32].

The structure of this paper is as follows. In Section II we will briefly explain the aircraft model and simulation environment that we have used in this study. We will also briefly explain the structure and design of the nominal control system. A background to the theory of reference and command governors are given in Section III together with a discussion on the different architectural design choices we have investigated. The final design and some examples from the simulations are presented and discussed in Section IV and we conclude the paper in Section V.

---

*Industry employed PhD student, Department of Electrical Engineering, dansi@isy.liu.se

†Technical Fellow, Flight Control Systems, ola.harkegard@saabgroup.com

‡Assistant Professor, Department of Electrical Engineering, johanl@isy.liu.se
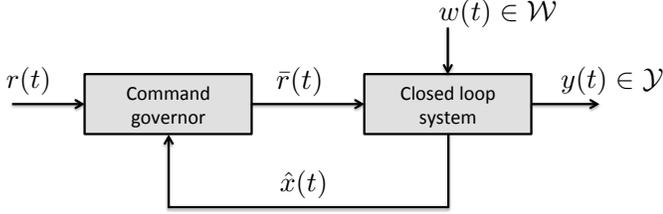
II.   Aircraft Model and Nominal Controller

This section outlines the aircraft simulation model used in this study. We applied the command governor design to the most complex flight dynamical models used at Saab to obtain a proof of concept. The ARES model, as it is called, is described in section A. Section B describes how the nominal controller is designed.

A.   ARES simulation environment

The ARES (Aircraft Rigid body Engineering Simulation) tool is the main flight dynamical simulation tool at Saab Aeronautics and is used in the development program for the JAS 39 Gripen fighter aircraft. It simulates the full nonlinear aircraft [36] with complex models for everything from the rigid body movement, nonlinear aerodynamics and atmosphere to structure loads, hydraulics, engine and sensors, see Figure 1. The model of the aircraft's nonlinear and varying aerodynamics is implemented as a piecewise affine model, scheduled over the different states of the aircraft as well as over the speed and altitude envelope.

Figure 1.   The ARES simulation environment structure with self contained sub-models.

The different sub-models are self contained and coded in Fortran, C and C++ and they are automatically linked together to form a complete simulation model for desktop or simulator analysis. ARES also supports replacement of a submodel with real hardware to form a hardware-in-the-loop simulation. ARES is used for flight dynamical simulations and for flight control law development and clearance in all parts of the development process, from early concept design to final clearance before flight. ARES also has support for generating linear state space models from the nonlinear data which is used in the design and analysis of the flight control system.

In this paper we have used an aerodata model that is similar to the *Generic Aerodata Model* (GAM) which is non-classified and freely distributed via ADMIRE [37]. Also other Gripen fighter specific data have been altered to declassify the research results.

B.   The Nominal Controller

The nominal control system that is implemented in ARES for the AD-MIRE like dynamics is an LQ controller with static feedforward and a proportional and integral feedback from the tracking error. A simplified schematic of the controller structure is shown in Figure 2.

Figure 2.   The nominal controller of the ARES model.

The ARES model is linearized around trimmed level flight at 25 different envelope points (of mach and altitude) in the subsonic region.

The short period dynamics is extracted from the linearized model and used for control law design. The linear model is on the form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$
$$y(t) = Cx(t) + Du(t) \tag{2}$$

where $x(t)$ is angle of attack, the pitch rate and the actuator states, $u(t)$ is the elevator and canard control surface commands and $y(t)$ is angle of attack or normal load factor, depending on which variable we want to control.

From these linearized models the LQ feedback gain, L, and feedforward gain, F, is calculated. The feedforward term is a static gain calculated such that the closed loop system should have a unit static gain, i.e.,

$$C(sI - A + BL)^{-1}F + DF = 1$$

The feedback and feedforward gains are tuned to ideally give the closed loop response $y = G_m(s)r$ where

$$G_m(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

and where the damping, ζ, and frequency, ω_0, vary with speed and altitude.

The pilot command to the controller is the increment in load factor, $\Delta n_{z,cmd}$, from trimmed flight. The reference input, r, to the controller is then calculated from the pilot command as both a total load factor command

$$n_{z,cmd} = \Delta n_{z,cmd} + \cos\theta$$

and an angle of attack command,

$$\alpha_{cmd} = K_{\alpha/n_z}\Delta n_{z,cmd} + \alpha_{trim}$$

The pilot command, $\Delta n_{z,cmd}$, is limited such that the load factor command and angle of attack command stays within the specified design limits

$$-3 \leq n_{z,cmd} \leq 9$$
$$-8 \leq \alpha_{cmd} \leq 18$$

The controller then tracks either the angle of attack command, α_{cmd}, or the load factor, $n_{z,cmd}$, command based on the current speed (over corner speed the controller tracks the load factor command and below corner speed it tracks the angle of attack command). From the selected reference command a nominal response, y_m is calculated as

$$y_m = G_m(s)r$$

where the model, $G_m(s)$, is the same for both angle of attack and load factor reference commands.

The nominal response is then compared to the actual response of the aircraft to calculate the model following error, $e = y - y_m$. The feedback term from the model following error is a proportional and integral term trying to integrate out all model errors.

Due to uncertainties, model errors and disturbances on the real aircraft the true closed loop response is never the ideal, $G_m(s)$. Throughout the flight envelope the actual closed loop response can have an overshoot of the design limits. This overshoot is highly undesirable since it can over stress the aircraft or put it into an unsafe state. Therefore we will add a command governor to the pilot commanded $\Delta n_{z,cmd}$ which alters the command to a new reference command that, as far as possible, will ensure that the design limits on the angle of attack and normal load factor are not exceeded.

III.   Command Governor Design

A.   Review of existing methods

A command governor, or reference governor, is a device that takes the reference input and alters it based on the current estimate of the system state, $\hat{x}(t)$, such that the output from the system remains within certain limits. The general structure of the closed loop system, from Figure 2, and command governor is shown in Figure 3.

Figure 3. The reference governor general structure.

The simplest form of a reference governor is the static reference governor [38–40] which optimizes a scalar gain, γ, such that when $\bar{r}(k) = \gamma r(k)$ is applied to the system the output constraints, $y(k) \in \mathcal{Y}$, are satisfied. This is done by solving the optimization problem [40]

$$\underset{\gamma \in [0,1]}{\text{maximize}} \; \gamma \;\; \text{s.t.} \;\; Ax(k) + B\gamma r(k) \in \mathcal{O}_\infty, \quad Cx(k) + D\gamma r(k) \in \mathcal{Y} \tag{3}$$

where $\mathcal{O}_\infty$ is the *maximal output admissible set* [41] and where $x(k) = \hat{x}(t)$. The maximal output admissible set for a system $x(k+1) = Ax(k) + Bw(k)$, $y(k) = Cx(k) + Dw(k)$ is formally defined as

$$\mathcal{O}_\infty = \left\{ x(k) \in \mathbb{R}^n \mid y(k+i) \in \mathcal{Y} \; \forall \, w(k+i) \in \mathcal{W} \; i \in Z^+ \right\} \tag{4}$$

This can easily be generalized to a definition where the combination of $x(k)$ and a constant reference, $\bar{r}$, is such that $y(k) \in \mathcal{Y}$ for all future time steps.

Since the static reference governor can suffer from oscillations a dynamic reference governor was developed [40] in which the reference was parametrized as

$$\bar{r}(k+1) = \bar{r}(k) + \gamma(r(k) - \bar{r}(k)) \tag{5}$$

and the optimization maximizes $\gamma \in [0, 1]$ such that the successor state is in the admissible set.

The dynamic reference governor is closely related to the more flexible command governor [42] where the reference $\bar{r}$ instead is parametrized as

$$\bar{r}(k+i) = \gamma^i \mu(k) + \nu(k) \tag{6}$$

where γ is a fixed constant $\gamma \in [0, 1)$ and the optimization variables are $\mu(t)$ and $\nu(t)$. The optimization problem is now formulated as

$$(\mu^*, \nu^*) = \arg\min_{\mu, \nu} \|\mu\|_Q^2 + \|\nu - r(k)\|_R^2$$
$$+ \sum_{i=0}^{\infty} \|y(k+i) - \nu\|_P^2$$
$$\text{s.t.} \;\; y(k+i) \in \mathcal{Y} \; \forall \, i \in Z^+$$

where $\mu \in \mathbb{R}^p$ and ν shall be selected from the set of constant signals such that the output in equilibrium is $y \in \mathcal{Y}$.

A generalization of the command governor is the extended command governor [43, 44] in which the decaying sequence $\gamma^i \mu(k)$ in the parametrization of $\bar{r}(k)$ is replaced by a fictitious system output, $\bar{C}\bar{x}(k)$, where the fictitious state evolve according to $\bar{x}(k+1) = \bar{A}\bar{x}(k)$. The matrix $\bar{A}$ should be stable but otherwise the matrices $\bar{A}$ and $\bar{C}$ can be chosen arbitrarily and it is easy to see that with a certain choice of $\bar{A}$ and $\bar{C}$ we recover the original command governor formulation. Gilbert and Ong [43] propose a shift sequence for the fictitious system, i.e.,

$$\bar{A} = \begin{bmatrix} 0 & I & 0 & \cdots & \\ 0 & 0 & I & 0 & \cdots \\ & & 0 & \ddots & \\ & & & 0 & I \\ & & & 0 & 0 \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} I & 0 & 0 & \cdots \end{bmatrix}$$

while Kalabic et al. [44] propose to use Laguerre sequences.

These basic principles discussed above have been extended to cover, e.g., nonlinear systems, robust reference governors and other types of structures. An excellent survey of the different types of reference and command governors is given in the paper by Kolmanovsky et al. [45].

B. Command governor design for aircraft maneuver limiting

Let us now, for the remaining part of this section, consider the different architectural choices that we have to make in order to find a suitable command governor design for the maneuver limiting application. The use of an output admissible set, $\mathcal{O}_\infty$, in our implementation of the command governor is not suitable since the dynamics of the aircraft varies over the altitude and speed envelope. The output admissible set which is calculated based on the dynamics has to either be calculated offline in advance based on a set of design points or recalculated in every iteration of the command governor. The second approach is to complex since it would require to much computational power in the iterative procedure of determining the polytopic shape of the set. In fact the number of inequalities describing the polytopic set might not even be finitely determined [41]. The first approach would not require the extensive online calculations but instead the sets must be *robust output admissible* with respect to the model errors that come from the changing dynamics and this defeats the purpose of our implementation. Furthermore no theoretical guarantees can be made when changing, or recalculating, the admissible set between two iterations.

An alternative approach is similar to the one in reference [32]. Simply constrain the N-step forward predictions of the model output, i.e.,

$$\underline{y} - s(k+i) \le y(k+i) \le \bar{y} + s(k+i), \quad \forall \, i = 1, \ldots, N \tag{7}$$

where $\underline{y}$ and $\bar{y}$ are the upper and lower limits for the output and $s(k+i) \ge 0$ is a slack variable added to soften the constraints. This of course does not give any guarantees that the constraints can be fulfilled for all future time steps, but it serves our purpose since we want to use it as a soft maneuver limit.

The use of the N-step prediction as constraints requires that the prediction model is explicitly implemented in the command governor. A very simple choice is to use a discrete time implementation of the desired closed loop response of the aircraft with the nominal controller.

$$G_m(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

From the nominal controller we get, in each sample time, the current desired damping, ζ, and frequency, ω_0. The discretization can be done in several different ways, e.g., using Euler forward, $s = \frac{1}{T_s}(q-1)$, Tustin's approximation, $s = \frac{2}{T_s}\frac{(q-1)}{(q+1)}$, or zero-order-hold. The Euler forward gives very simple analytical expressions for how the coefficients, a, b and c in the discrete time step response model, $y(k+1) = ay(k) + by(k-1) + cu(k-1)$, depend on the current damping and frequency, which are frequently updated.

$$a = 2(1 - T_s\zeta\omega_0), \quad b = 2\zeta\omega_0 T_s - 1 - w_0^2 T_s^2, \quad c = w_0^2 T_s^2$$

However the Euler discretization requires a sufficiently high sampling rate to be accurate enough, see Figure 4. A high sampling rate requires a large number of prediction steps, N, to achieve a sufficiently long prediction time. The objective of having a long prediction time is that we want the command governor to react earlier when a constraint violation is predicted.

If we instead use Tustin's approximation or zero-order-hold discretization we can implement a lower sampling rate in the prediction model and thus reduce the number of prediction steps required to achieve the same prediction time. Although the zero-order-hold is the most accurate approximation, it is exact in the sampling instants, it doesn't have the simple analytical expressions for the dependence of damping and frequency in the model coefficients.

Implementing both the Euler forward discretization and Tustin's approximation reveals that the longer prediction time that can be achieved with Tustin (with the same number of prediction steps as with Euler forward) does not give any improvement in the performance of
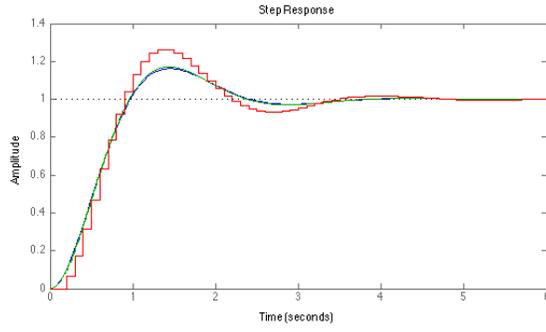
Figure 4. The step response for a continuous time system (blue line) and for the corresponding discrete time system using Euler forward discretization with sampling time $T_s = 0.01$ s (green line) and $T_s = 0.1$ s (red line).
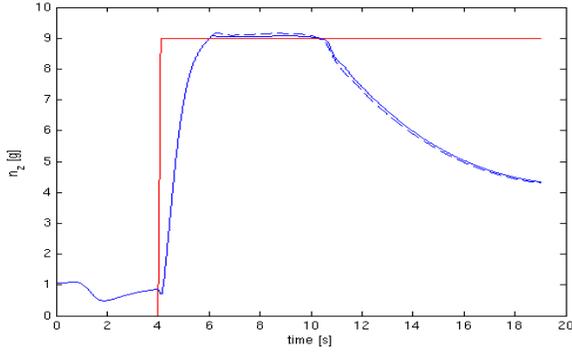

Figure 5. Load factor response for the command governor with Euler discretization of the response model (solid blue line) and with the Tustin discretization (dashed blue line). Red line is pilot commanded load factor.

the command governor compared to the Euler approximation, see Figure 5. In this figure we can see that the load factor responses are almost identical for the two discretization methods. The reason for this is probably that the desired closed loop response model, $G_m(q)$, that we use as prediction model, does not accurately enough model the true closed loop system to benefit from the longer prediction time that we get with Tustin's approximation.

An attempt to solve this can be to add a model correction term, ε, to the prediction model

$$y_m(k) = G_m(q)\bar{r}(k) + \varepsilon(k)$$

and then estimate the correction term online. A simple way to do this has been suggested in literature to handle nonlinear systems [45]. This technique uses the output from the linear prediction model of the system

$$y_{linear}(k) = G_m(q)\bar{r}(k)$$

and compare that to the true closed loop system output, $y(k)$, to estimate a constant model correction term

$$\varepsilon(k+i) = y(k) - y_{linear}(k) \quad \forall\, i = 0, \ldots, N$$

and then add this to the prediction model output.

However when implementing this model error estimation in the simulation environment only very little performance is gained, see Figure 6.

For our transfer function model the correction factor, ε, adds a constant offset correction to the predictions and does not capture dynamical model errors such as e.g., errors in the damping. Since the nominal closed loop system has integral action the steady state error is minimal and the true model error probably comes from dynamical properties.

As discussed in the beginning of this section, there exist several different possibilities to parametrize the command governor output $\bar{r}(k)$
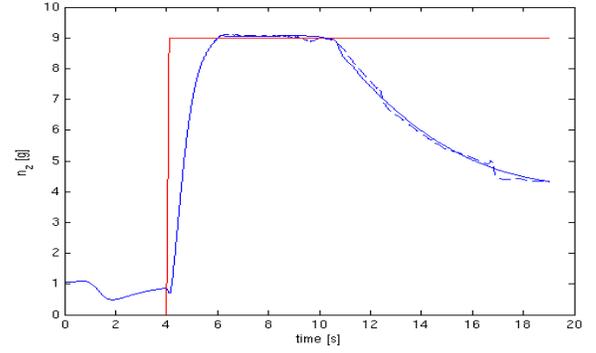
Figure 6. Load factor response for the command governor with model error estimator (dashed blue line) and without the model error estimator (solid blue line).

and to formulate the objective function. An intuitive and straight forward formulation of the objective function is to simply penalize the difference between the pilot commanded reference, $r(k)$, and the applied reference, $\bar{r}(k)$, as

$$\sum_{i=0}^{N} (\bar{r}(k+i) - r(k))^2 \tag{8}$$

This objective function can be used with both static and dynamic reference governor formulations. However for our purposes the standard formulation of the dynamic reference governor (5) is not suitable since it requires that the governor is robustly designed with respect to constraint violations. If, e.g., a disturbance enters the system at time k the parametrization of $\bar{r}$ does not allow it to be reduced from the value at time k, since $\gamma \geq 0$, and hence cannot counteract the disturbance. In our application of maneuver limiting we want the reference governor to allow the output $y(k)$ to reach the limit, but as far as possible not overshoot it. Therefore do we want to allow the applied reference $\bar{r}$ to be reduced if disturbances or model errors predict an overshoot.

If we Instead choose to use a static reference governor $\bar{r}(k) \in \mathcal{Y}$ with $\bar{r}(k+i) = \bar{r}(k)\ \forall\ i = 1, \ldots, N$. This is a very attractive choice since the optimization problem then only has one free variable to optimize over and it becomes almost trivial to solve. However the static reference governor might suffer from oscillations in the closed loop response [40]. Even though our simulations do not indicate that oscillations could arise this study is not enough to make conclusive statements about that.

An alternative is to let $\bar{r}(k+i)$ be a sequence of free optimization variables in an MPC like fashion. This gives the algorithm maximum flexibility in the choice of future applied reference inputs but with the cost of increasing the dimension of the optimization variable from 1 to N. The simulations of the command governor does not indicate a large enough performance gain to motivate such an increase in complexity. In fact for some simulations there are no performance increase by using a sequence of reference signals in comparison to using a constant reference, see Figure 7.

The best alternative seems to be the command governor [42] or the extended command governor [43] parametrization of $\bar{r}$. The command governor gives the algorithm good flexibility but with limited complexity increase. For the command governor approach with $\bar{r}(k+i) = \gamma^i \mu(k) + \nu(k)$ we can adopt a similar formulation of the objective as in Bemporad [42] and formulate it such as

$$\underset{\mu, \nu}{\text{minimize}}\ \beta_1 \mu(k)^2 + \beta_2 (\nu(k) - r(k))^2 \tag{9}$$

where the first term can be viewed as a penalty on the changes in the sequence of $\bar{r}$ and the second term is penalty on the stationary deviation from desired reference.

Finally the objective function must also include some penalty on the slack variables, $s(k+i)$, used to soften the constraints (7). The most common choices are to use a quadratic or a linear penalty. The
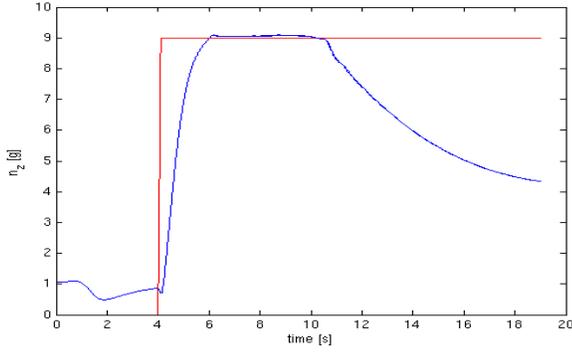
Figure 7. A comparison between a command governor with a constant reference signal and one with a sequence of reference signals in the prediction model. Blue dashed line is the command governor with a reference sequence and the solid blur line is the command governor with a constant reference over the prediction horizon.

linear penalty has the advantage that if it is high enough it will force the slack to be zero if there exist such a feasible solution and only non-zero otherwise. On the other hand with a linear penalty the control signal has a tendency to have more abrupt changes while it is fairly smooth with quadratic penalty on the slack.

IV. Simulation Results

In this section we will discuss the achieved simulation results when implementing an outer loop command governor in the ARES simulation environment at Saab Aeronautics for maneuver limiting. The implemented command governor is formulated as the minimization problem

$$\underset{\mu,\nu,s_\alpha,s_{n_z}}{\text{minimize}} \ \beta_1\mu(k)^2 + \beta_2(\nu(k)-\Delta n_{z,cmd}(k))^2$$

$$+ \sum_{i=1}^{N}(s_\alpha(k+i)^2 + s_{n_z}(k+i)^2) \qquad (10)$$

subject to the following constraints

$$\alpha(k+i) = G_m(q)\bar{\alpha}_{cmd}(k+i) \qquad (11a)$$
$$n_z(k+i) = G_m(q)\bar{n}_{z,cmd}(k+i) \qquad (11b)$$
$$\bar{\alpha}_{cmd}(k+i) = K_{\alpha/n_z}(\gamma^i\mu(k)+\nu(k)) + \alpha_{trim} \qquad (11c)$$
$$\bar{n}_{z,cmd}(k+i) = (\gamma^i\mu(k)+\nu(k)) + \cos\theta \qquad (11d)$$
$$\alpha_{min} - s_\alpha(k+i) \le \alpha(k+i) + \varepsilon_\alpha \le \alpha_{max} + s_\alpha(k+i) \quad (11e)$$
$$n_{z,min} - s_{n_z}(k+i) \le n_z(k+i) + \varepsilon_{n_z} \le n_{z,max} + s_{n_z}(k+i)$$
$$(11f)$$
$$s_\alpha(k+i) \ge 0 \qquad (11g)$$
$$s_{n_z}(k+i) \ge 0 \qquad (11h)$$

where the constraints are for $i = 1, \ldots, N$. We have used the Euler forward discretization for the response model, $G_m(q)$, which gives the following relation between the time steps of the predicted outputs

$$\alpha(k+1) = a\alpha(k) + b\alpha(k-1) + c\bar{\alpha}_{cmd}(k-1)$$

with $a = 2(1 - T_s\zeta\omega_0)$, $b = 2\zeta\omega_0 T_s - 1 - w_0^2 T_s^2$, and $c = w_0^2 T_s^2$, and similar for the load factor model. To simplify the implementation we select the same sampling time for the command governor as for the nominal controller, $T_s = 1/60$ s. The input to the command governor is the measured angle of attack and load factor at the current time step and the previous time step as well as the current delta load factor command. The selection of prediction horizon is a tradeoff between performance of the command governor and size of optimization problem to be solved. The step response time of the nominal controller, in well tuned design points, is approximately 2 seconds and it would

be reasonable to include at least half of that, i.e., $N = 50$ samples, in the prediction horizon. However, from simulations we experienced no noticeable increase in performance of the command governor for prediction horizons above approximately 40 samples and hence that was chosen as prediction horizon. Several different objective function weights were tested. The β-weights were changed with a factor of ten from small weights up to weights larger than the slack weight. In general we also wanted a larger weight on the deviation from the pilots reference input, i.e., β_2, than on the transient behavior of the reference, i.e., β_1. In Figure 8 the load factor step response of the closed loop system is shown for several different β-tunings. From this we can conclude that too small weights will cause a very bad response. This is due to the shortcomings in the prediction model combined with an excessive weight on the contraint violations. If the weights are selected larger than one, i.e., larger than the slack weight, the command governor will not have any effect on the closed loop system. The ideal
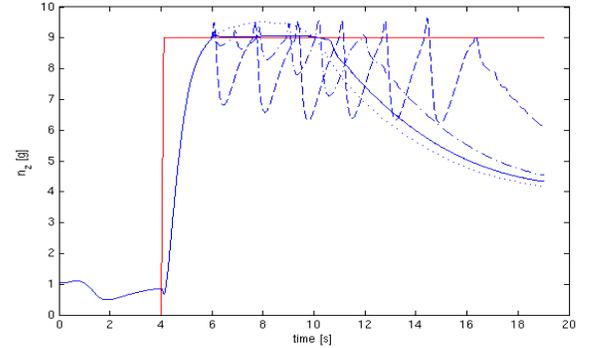

Figure 8. The normal load factor response of the closed loop system for different weight combinations in the command governor objective function. The solid blue line is the weights $\beta_1 = 0.01$, $\beta_2 = 0.1$. The dashed and dash-dotted lines are $\beta_1 = 0.001$, $\beta_2 = 0.01$ and $\beta_1 = 0.01$, $\beta_2 = 0.001$ respectively and the dotted line are the tuning with $\beta_1 = 10$, $\beta_2 = 100$. The red line is the pilot commanded load factor.

weights from the simulations have been selected to

$$\beta_1 = 0.01, \quad \beta_2 = 0.1$$

The formulation (10) and (11) is a standard quadratic program and it has been implemented using the QP solver qpOASES [46]. Extensive research has been performed to develop fast, real-time solvers for quadratic programs [47–49] enabling real-time implementation in aircraft computer systems. Additionally if we take a closer look at the problem (10) and (11) we can see that it in fact only has the two free variables, μ and ν, to optimize. All other variables are implicitly given by these two. This mean that we can perform a simple bisection search in the two variables to solve the problem in micro seconds.

To illustrate the achieved performance of the command governor implementation in ARES we have made a series of *bleed-off turns* at different mach and altitude points. A bleed-off turn is a maximum turn in which the speed reduces continuously throughout the maneuver, designed to test maneuver load limiting systems. If the maneuver is initiated at speeds above corner speed then the load factor limit will be the most limiting constraint and as the speed reduces the angle of attack limit will become the active constraint.

In Figure 9 we have plotted the angle of attack and load factor responses from a bleed-off turn initiated at Mach 0.75 and altitude 1000 m.

The upper two figures show the angle of attack and load factor response and the lower two figures show the commanded load factor and angle of attack as calculated from the pilot stick input. The green thick lines show the responses with the command governor and the blue thin lines the responses without the command governor. In this simulation the command governor achieves a distinct reduction in the maneuver limit overshoot, from about 0.6 g to 0.1 g overshoot in load factor and 1.2 degrees to 0.2 degrees in angle of attack.
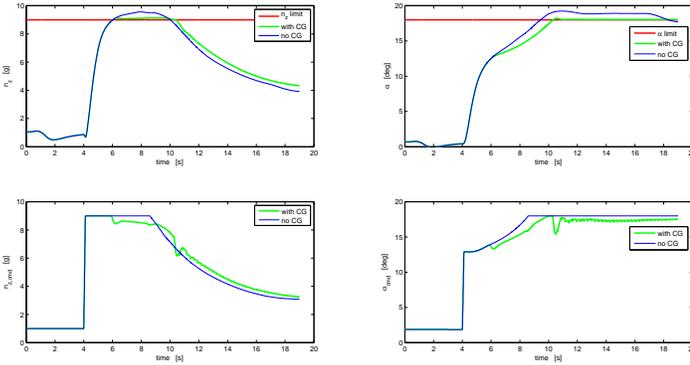
Figure 9. An ARES simulation of a bleed-off turn from Mach 0.75 and altitude 1000 m.

Figure 11. An ARES simulation of a bleed-off turn from Mach 0.6 and altitude 6000 m.

In the lower parts of the figure we can see that at the beginning of the maneuver the command governor does not interfere with the pilots command, but as soon as the command governor predicts an overshoot of the maneuver limit it modifies the commanded load factor change, $\Delta n_{z,cmd}$, which is used to calculate the commands, $n_{z,cmd}$ and α_{cmd}. By more closely examining the figures one can see that the command governor reacts fairly late, approximately 0.1 s., before the overshoot, compared to the prediction horizon, which is 0.67 s. long. We have tried several different ways to have the command governor react earlier but without success. Our conclusion is, as discussed in the previous section, that this is a result of an imperfect model knowledge of the closed loop system.

At lower speeds, when the nominal maneuver limit overshoot is even bigger, the short reaction time of the command governor result in quite aggressive reference adjustment, see Figure 10. This aggressive adjustment causes a small oscillation in the angle of attack response. This is most likely a result of the tuning of the command governor which is constant throughout the flight envelope. In a final implementation it is suggested to schedule the objective function weights, β_i, as a function of speed and altitude, just as is done in the design of the nominal controller.
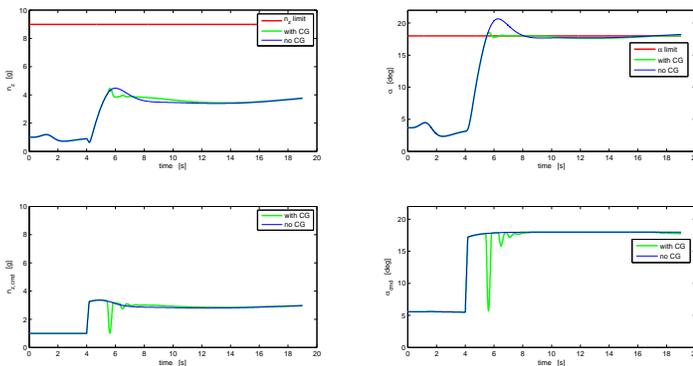
Figure 10. An ARES simulation of a bleed-off turn from Mach 0.4 and altitude 1000 m.

Also in this low speed case there is a significant reduction in the overshoot. It should be noted here that the two illustrated maneuvers in figures 9 and 10 are performed in envelope points that are not design points of nominal controller. This means that the true closed loop response is not as close to the desired closed loop response as it is at the design points. This means that the overshoot will be bigger here and at the same time will the command governor have more difficulties predicting it, i.e., we are studying the hard cases here.

If we instead investigate the response in one of the envelope points where the nominal controller has been tuned we can see that the command governor makes only minimal adjustments to the reference command and there are in principle no difference between the responses, see Figure 11.
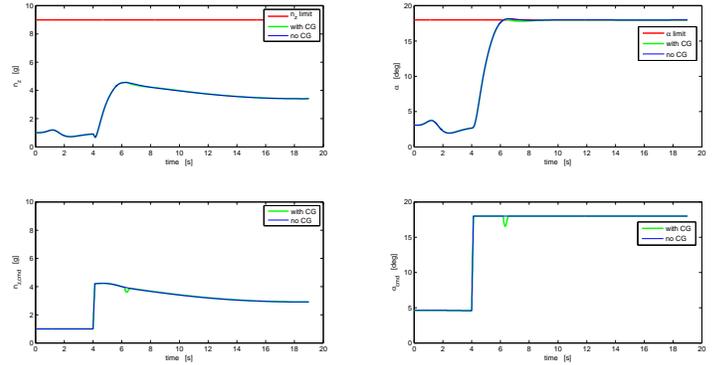
V. Conclusion

In this paper we have investigated reference and command governors as a method of imposing maneuver limiting in fighter aircraft control systems. The proposed command governor structure has been implemented and verified in the ARES (Aircraft Rigid Body Engineering Simulation) environment at Saab Aeronautics, which is the simulation environment used in the Jas 39 Gripen fighter aircraft development program.

The realism of the simulation environment gives proof of concept since it is as close as possible to real flight testing, in this case with the exception of implementing on target hardware. Due to the complexity of the problem with nonlinear dynamics that varies throughout the flight envelope the use of standard reference governor formulations with an output admissible set is not suitable. In this paper we have instead used a more model predictive control like scheme with a finite prediction horizon and point wise in time output constraints.

The implemented command governor gives a substantial reduction in the angle of attack and load factor limit overshoot. It improves the design in envelope points where the nominal controller due to model errors or tuning is insufficient. However in these areas the used prediction model is not as accurate as desired. Methods to overcome the model errors in the prediction model showed to be insufficient and future work consist of analyzing the model error and developing a technique that has a better potential of capturing the relevant dynamics.

As a design method the command governor as shown to be easy to implement and tune. The feature that the command governor can be used as an add-on functionality to existing control system architecture makes it a very attractive choice for future development.

Acknowledgments

References

[1] Fabio Andrade de Almeida. Offline Model Predictive Control Based on Weighted Projection over Polytopes. *Journal of Applied Mathematics*, 2015:1–14, 2015. ISSN 1110-757X. doi: 10.1155/2015/741348.

[2] Fabio Andrade de Almeida and D. Leissling. Fault-Tolerant Flight Control System Using Model Predictive Control. In *2009 Brazilian Symposium on Aerospace Eng. & Applications*, 2009.

[3] David Amsallem, Sunil Deolalikar, Fazzel Gurrola, and Charbel Farhat. Model Predictive Control under Coupled Fluid-Structure Constraints Using a Database of Reduced-Order Models on a Tablet. In *21st AIAA Computational Fluid Dynamics Conference*. AIAA Paper 2013-2588, 2013. doi: 10.2514/6.2013-2588.

[4] Raktim Bhattacharya, Gary J Balas, M. Alpay Kaya, and Andy Packard. Nonlinear Receding Horizon Control of an F-16 Aircraft. *Journal of Guidance, Control, and Dynamics*, 25(5):924–931, sep 2002. ISSN 0731-5090. doi: 10.2514/2.4965.

[5] William B Dunbar, Mark B Milam, Ryan Franz, and Richard M Murray. Model Predictive Control of a Thrust-vectored Flight Control Experiment. *IFAC Proceedings Volumes*, 35(1):355–360, 2002. ISSN 14746670. doi: 10.3182/20020721-6-ES-1901.00965.

[6] Derek W Ebdon. *Model Predictive Control of Aerospace Systems*. PhD thesis, Air force insitute of technology, 1996.

[7] Wouter Falkena, Clark Borst, Q.P. Chu, and J.A. Mulder. Investigation of Practical Flight Envelope Protection Systems for Small Aircraft. *Journal of Guidance, Control, and Dynamics*, 34(4):976–988, jul 2011. ISSN 0731-5090. doi: 10.2514/1.53000.

[8] Hiroaki Fukushima, Ryosuke Saito, Fumitoshi Matsuno, Yasushi Hada, Kuniaki Kawabata, and Hajime Asama. Model Predictive Control of an Autonomous Blimp with Input and Output Constraints. In *2006 IEEE International Conference on Control Applications*, pages 2184–2189. IEEE, oct 2006. ISBN 0-7803-9796-7. doi: 10.1109/CCA.2006.286205.

[9] Peter W Gibbens and Eran D B Medagoda. Efficient Model Predictive Control Algorithm for Aircraft. *Journal of Guidance, Control, and Dynamics*, 34(6):1909–1915, 2011. doi: 10.2514/1.52162.

[10] H.-G. Giesseler, M Kopf, P Varutti, T Faulwasser, and Rolf Findeisen. Model Predictive Control for Gust Load Alleviation. *IFAC Proceedings Volumes*, 45(17):27–32, 2012. ISSN 14746670. doi: 10.3182/20120823-5-NL-3013.00049.

[11] Sebastien Gros, Rien Quirynen, and Moritz Diehl. Aircraft control based on fast non-linear MPC & multiple-shooting. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, number 1, pages 1142–1147. IEEE, dec 2012. ISBN 978-1-4673-2066-5. doi: 10.1109/CDC.2012.6426439.

[12] Kristoffer Gryte and Thor I Fossen. Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing. In *AIAA Guidance, Navigation, and Control Conference*. AIAA Paper 2016-0512, 2016. ISBN 978-1-62410-388-9. doi: 10.2514/6.2016-0512.

[13] E N Hartley. Predictive Control with Parameter Adaptation to Achieve α -Protection in the RECONFIGURE Benchmark in the Presence of Icing. *IFAC-PapersOnLine*, 48(314):172–177, 2015. ISSN 24058963. doi: 10.1016/j.ifacol.2015.11.279.

[14] E N Hartley and J M Maciejowski. A longitudinal flight control law based on robust MPC and H2 methods to accommodate sensor loss in the RECONFIGURE benchmark. Technical report, 2015. URL www.sciencedirect.com.

[15] E N Hartley, J M Maciejowski, and K-V Ling. Performance evaluation of multiplexed model predictive control for a large airliner in nominal and contingency scenarios. In *2012 American Control Conference (ACC)*, pages 1199–1204. IEEE, jun 2012. ISBN 978-1-4577-1096-4. doi: 10.1109/ACC.2012.6314656.

[16] M.M. Kale and A.J. Chipperfield. Stabilized MPC formulations for robust reconfigurable flight control. *Control Engineering Practice*, 13(6):771–788, jun 2005. ISSN 09670661. doi: 10.1016/j.conengprac.2004.09.001.

[17] Tamás Keviczky and Gary J. Balas. Receding horizon control of an F-16 aircraft: A comparative study. *Control Engineering Practice*, 14(9):1023–1033, sep 2006. ISSN 09670661. doi: 10.1016/j.conengprac.2005.06.003.

[18] Tamás Keviczky and G.J. Balas. Flight test of a receding horizon controller for autonomous UAV guidance. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 3518–3523. IEEE, 2005. ISBN 0-7803-9098-9. doi: 10.1109/ACC.2005.1470518.

[19] Tamas Keviczky and Gary J. Balas. Software-Enabled Receding Horizon Control for Autonomous Unmanned Aerial Vehicle Guidance. *Journal of Guidance, Control, and Dynamics*, 29(3):680–694, 2006. ISSN 0731-5090. doi: 10.2514/1.15562.

[20] D K Kufoalor and T A Johansen. Reconfigurable fault tolerant flight control based on Nonlinear Model Predictive Control. In *2013 American Control Conference*, pages 5128–5133. IEEE, jun 2013. ISBN 978-1-4799-0178-4. doi: 10.1109/ACC.2013.6580635.

[21] Thomas Lombaerts, Gertjan Looye, Joost Ellerbroek, and Mitchell Rodriguez Martin. Design and Piloted Simulator Evaluation of Adaptive Safe Flight Envelope Protection Algorithm. In *AIAA Guidance Navigation and Control Conference*. AIAA Paper 2016-0093, 2016. doi: 10.2514/6.2016-0093.

[22] Jan M Maciejowski and Colin N Jones. MPC Fault-tolerant Flight Control Case Study: Flight 1862. In *IFAC Safeprocess Conference*, 2003.

[23] Christopher Petersen, Morgan Baldwin, and Ilya Kolmanovsky. Model Predictive Control Guidance with Extended Command Governor Inner-Loop Flight Control for Hypersonic Vehicles. In *AIAA Guidance, Navigation, and Control (GNC) Conference*. AIAA Paper 2013-5028, 2013. ISBN 978-1-62410-224-0. doi: 10.2514/6.2013-5028.

[24] Arthur Richards, William Stewart, and Alex Wilkinson. Auto-coding Implementation of Model Predictive Control with Application to Flight Control. *European Control Conference*, pages 150–155, 2009.

[25] C. Shearer and S. Heise. Constrained model predictive control of a nonlinear aerospace system. In *Guidance, Navigation, and Control Conference and Exhibit*. AIAA Paper 98-4235, 1998. doi: 10.2514/6.1998-4235.

[26] Marc Steinberg. A comparison of intelligent, adaptive, and nonlinear flight control laws. In *Guidance, Navigation, and Control Conference and Exhibit*, volume 24, page 11. AIAA Paper 99-4044, 1999. doi: 10.2514/6.1999-4044.

[27] Yinan Wang, Andrew Wynn, and Rafael Palacios. Model-Predictive Control of Flexible Aircraft Dynamics using Nonlinear Reduced-Order Models. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number January, pages 1–11, 2016. ISBN 978-1-62410-392-6. doi: 10.2514/6.2016-0711.

[28] Zhi-Jun Yang, Xiao-Hui Qi, and Gan-Lin Shan. Simulation of Flight Control Laws Design Using Model Predictive Controllers. In *IEEE International Conference on Mechatronics and Automation*, pages 4213–4218, 2009.

[29] Alicia Zinnecker, Andrea Serrani, Michael Bolender, and David Doman. Combined Reference Governor and Anti-Windup Design for Constrained Hypersonic Vehicles Models. In *AIAA Guidance, Navigation, and Control Conference*. AIAA Paper 2009-6283, 2009. ISBN 978-1-60086-978-5. doi: 10.2514/6.2009-6283.

[30] Ilya Kolmanovsky and Nazli E Kahveci. Constrained Control of UAVs Using Adaptive Anti-windup Compensation and Reference Governors. In *Proceedings of SAE AeroTech Congress and Exhibition*, 2009.

[31] Davide Martino. Flight control with amplitude and rate constraints: A command governor approach. In *American Control Conference*, pages 1788–1793, 2008. ISBN 9781424420797. doi: 10.1109/ACC.2008.4586751.

[32] Hui Ye, Mou Chen, and Qingxian Wu. Flight Envelope Protection Control Based on Reference Governor Method in High Angle of Attack Maneuver. *Mathematical Problems in Engineering*, 2015:1–15, 2015. ISSN 1024-123X. doi: 10.1155/2015/254975.

[33] Domenico Famularo, Davide Martino, and Massimiliano Mattei. Constrained Control Strategies to Improve Safety and Comfort on Aircraft. *Journal of Guidance, Control, and Dynamics*, 31(6):1782–1792, 2008. ISSN 0731-5090. doi: 10.2514/1.34426.

[34] Uros Kalabic, Yash Chitalia, Julia Buckland, and Ilya Kolmanovsky. Prioritization Schemes for Reference and Command Governors. In *European Control Conference*, pages 2734–2739, 2013. ISBN 9783952417348.

[35] Qinghua Li, Uros V. Kalabic, and Ilya V. Kolmanovsky. Fast reference governors for second-order linear systems with constraints and an input time-delay. *Automatica*, 50(2):641–645, 2014. ISSN 00051098. doi: 10.1016/j.automatica.2013.11.007.

[36] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, 2 edition, 2003. ISBN 0-471-37145-9.

[37] Lars Forssell and Ulrik Nilsson. ADMIRE The Aero-Data Model In a Research Environment Version 4.0, Model Description. Technical Report December, Swedish Defence Research Agency, 2005.

[38] Elmer G. Gilbert and Kok Tin Tan. Maximal output admissible sets and the nonlinear control of linear discrete-time systems with state and control constraints. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[39] E.G. Gilbert, I. Kolmanovsky, and K.T. Tan. Nonlinear control of discrete-time linear systems with state and control constraints: a reference governor with global convergence properties. *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, 1:144–149, 1994. doi: 10.1109/CDC.1994.411031.

[40] Elmer G Gilbert, Ilya Kolmanovsky, and K.T. Tan. Discrete-time reference governsors and the nonlinear control of systems with state and control constraints. *International Journal of Robust and Nonlinear Control*, 5 (1995):487–504, 1995.

[41] Ilya Kolmanovsky and Elmer G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998. ISSN 1024-123X. doi: 10.1155/S1024123X98000866.

[42] Alberto Bemporad, Alessandro Casavola, and Edoardo Mosca. Nonlinear control of constrained linear systems via predictive reference management. *Automatic Control, IEEE Transactions on*, 42(3):340–349, 1997. ISSN 00189286. doi: 10.1109/9.557577.

[43] Elmer G. Gilbert and Chong Jin Ong. Constrained linear systems with hard constraints and disturbances: An extended command governor with large domain of attraction. *Automatica*, 47(2):334–340, 2011. ISSN 00051098. doi: 10.1016/j.automatica.2010.10.016.

[44] Uros Kalabic, Ilya Kolmanovsky, Julia Buckland, and Elmer Gilbert. Reference and extended command governors for control of turbocharged gasoline engines based on linear models. In *Proceedings of the IEEE International Conference on Control Applications*, pages 319–325, 2011. ISBN 9781457710629. doi: 10.1109/CCA.2011.6044457.

[45] Ilya Kolmanovsky, Emanuele Garone, and Stefano Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. *2014 American Control Conference*, pages 226–241, jun 2014. doi: 10.1109/ACC.2014.6859176.

[46] Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock, and Moritz Diehl. qpOASES: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4): 327–363, dec 2014. ISSN 1867-2949. doi: 10.1007/s12532-014-0071-1.

[47] Juan L Jerez, Paul J Goulart, Stefan Richter, George A Constantinides, Eric C Kerrigan, and Manfred Morari. Embedded Online Optimization for Model Predictive Control at Megahertz Rates. *IEEE Transactions on Automatic Control*, 59(12):3238–3251, 2014.

[48] Stefan Richter, Colin Neil Jones, and Manfred Morari. Computational Complexity Certification for Real-Time MPC With Input Constraints Based on the Fast Gradient Method. *IEEE Transactions on Automatic Control*, 57(6):1391–1403, 2012.

[49] Colin N Jones, Alexander Domahidi, Manfred Morari, Stefan Richter, Fabian Ullmann, and Melanie Zeilinger. Fast Predictive Control : Real-time Computation and Certification. In *4th IFAC Nonlinear model predictive control conference*. IFAC, 2012. ISBN 9783902823076. doi: 10.3182/20120823-5-NL-3013.00075.