

Viability Kernel Based Control Approach for a Flight Simulator Model ^{*} ^{**}

Alexander Gerdt¹[0000-0003-3516-2838], Nikolai Botkin²[0000-0003-2724-4817],
Johannes Diepolder¹[0000-0002-5962-6126], Varvara Turova²[0000-0002-6613-3311],
and Florian Holzapfel¹[0000-0003-4733-9186]

¹ Institute of Flight System Dynamics, Technische Universität München,
Garching bei München, Germany
{alexander.gerdt, johannes.diepolder, florian.holzapfel}@tum.de

² Mathematical Faculty, Chair of Mathematical Modelling,
Technische Universität München, Garching bei München, Germany
turova@ma.tum.de

Abstract. An approach for the application of differential game theory to control a realistic flight simulator model is presented. In the context of aircraft control safe operation must be ensured particularly under external disturbances (e.g. wind). The application of viability theory enables a determination of envelopes (viability kernels) in which safe operation is guaranteed. Here, a state-feedback control law can be derived based on a viability kernel which keeps the dynamic system within a set of safe states. So far, our solver implementation on a supercomputer allows us to compute viability kernels for general nonlinear dynamic systems in up to seven state dimensions. Unfortunately, the mathematical model of the flight simulator consists of about one hundred differential equations. Therefore, the following procedure is used to enable the application of the viability kernel based control. First, a reduced model of the flight simulator model is derived for the calculation of the viability kernel in up to seven state dimensions. Then, the optimal controls are determined through the evaluation of the precomputed viability kernel using the reduced model. In order to apply the optimal controls to the flight simulator model a Nonlinear Dynamic Inversion (NDI) control architecture is used. This NDI controller contains two cascaded control loops with modified reference models of relative degree one. Numerical experiments in a cruise flight condition using different wind disturbances suggest that the proposed control procedure keeps the considered states of the flight simulator model within the viability kernel.

Keywords: Aircraft control · Differential games · Viability kernel · Flight simulator · Reduced model · Nonlinear dynamic inversion.

^{*} This work has been supported by the DFG grant TU427/2-2, HO4190/8-2, and the LRZ supercomputing grant pr74lu.

^{**} Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1 Introduction

Wind is a common and unforeseeable disturbance, which has a strong effect on the aerodynamic forces, and thus, flight dynamics. Therefore, a robust performance of flight control systems regarding wind disturbances is of paramount importance to ensure safe aircraft operation. In this paper, the application of viability theory is investigated for this purpose in which the viability kernel takes a central role. This viability kernel is the largest subset of the state constraints in which a system can remain arbitrarily long for all admissible disturbances if an appropriate feedback control is used [4]. It is remarkable that an optimal feedback control which ensures safe aircraft operation can be constructed if the viability kernel is known. The viability approach has already been successfully applied in a similar context to a simplified aircraft model [8].

The notion of viability kernel is clarified in [1] for control systems and in [3, 4] for conflict control problems. It should be emphasized that the notion of viability kernel is more appropriate for control systems (without disturbances). In the case of differential games, the terms *discriminating* and *leadership* kernels are more suitable, see e.g. [6]. The discriminating kernel corresponds to the case where the first player (pilot) can exactly measure current wind components to use counter feedback strategies. In contrast, the leadership kernel assumes that the second player (wind) knows the current controls of the pilot and uses feedback counter-strategies, which is rather realistic in the context of computing guaranteeing controls. If the saddle point condition (2) holds, then discriminating and leadership kernels coincide. It is shown that for the application considered in this paper the saddle point condition (2) holds, and, therefore, we will keep the term *viability kernel* also in the case of differential games.

The numerical computation of viability kernels, using a highly parallelized implementation on dozens of compute nodes for a supercomputer, is currently possible, with reasonable effort, for dynamic systems containing up to seven state variables. However, as the realistic flight simulator model considered in this study consists of about one hundred state variables it is unrealistic to directly apply the viability kernel based control. Therefore, our approach considers the computation of the viability kernel only for a reduced model. Clearly, this reduced model should, on the one hand, allow us to compute the viability kernel (i.e. not contain more than seven states) and, on the other hand, reflect the behavior (considered dynamics) of the flight simulator model as well as possible. For a formulation of such a model, we use a first-order reference model (RM) prescribing the attitude dynamics, which yields an eight-state self-contained model together with the altitude, translational, and thrust dynamics. Assuming one state variable to be constant, a seven-dimensional model is obtained, which meets the requirements regarding the numerical computation of the viability kernel. This allows us to construct a feedback strategy that keeps all seven state variables of the reduced model inside the viability kernel. It is shown that applying the same feedback strategy to the flight simulator model, which uses the same RM for the attitude loop, makes it possible to keep the states of the flight simulator model within the viability kernel.

The paper is organized as follows: Section 2 describes the numerical method for computing viability kernels regarding conflict control problems. The flight simulator model is briefly described in Section 3. The reduced model is outlined in Section 4 and Section 5 presents the control architecture based on Nonlinear Dynamic Inversion (NDI). Finally, Section 6 demonstrates the application to simulated cruise flight trajectories of a flight simulator model for different wind disturbances. A concluding discussion is given in Section 7.

2 Computation of the Viability Kernel

In the following, a numerical method for the computation of viability kernels will be outlined. Details regarding the theoretical background and implementation can be found in [3] and [4]. See also [12] for methods and techniques of differential game theory.

Consider a general state constrained conflict control problem with the dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v}), \quad (1)$$

where $\mathbf{x} = [x_1, \dots, x_n]' \in R^n$ represents the states, and $\mathbf{u} = [u_1, \dots, u_{n_p}]' \in P \subset R^{n_p}$ and $\mathbf{v} = [v_1, \dots, v_{n_q}]' \in Q \subset R^{n_q}$ stand for controls of the first and second player, respectively. Here and in what follows, the symbol “'” denotes transposition.

Assume, that the Isaacs saddle point condition holds:

$$\min_{u \in P} \max_{v \in Q} \ell' \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = \max_{v \in Q} \min_{u \in P} \ell' \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v}), \quad \ell, \mathbf{x} \in R^n. \quad (2)$$

This saddle point condition is fulfilled for right-hand sides with additively separable controls

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v}) = \mathbf{f}_u(\mathbf{x}, \mathbf{u}) + \mathbf{f}_v(\mathbf{x}, \mathbf{v}) \quad (3)$$

which is the case for our application (see Section 4).

The objective of the first player (aircraft commands) is to stay within the state constraint, whereas the objective of the second player (wind) is the opposite. The state constraints and the bounds on the control variables of the first and second players are defined as:

$$G_0 : x_{i,Viab}^{lb} \leq x_i \leq x_{i,Viab}^{ub}, \quad i = 1, \dots, n, \quad (4)$$

$$P : u_{i,Viab}^{lb} \leq u_i \leq u_{i,Viab}^{ub}, \quad i = 1, \dots, n_p, \quad (5)$$

$$Q : v_{i,Viab}^{lb} \leq v_i \leq v_{i,Viab}^{ub}, \quad i = 1, \dots, n_q. \quad (6)$$

The viability kernel represents the largest subset of the state constraint in which the system trajectory can be kept arbitrarily long if the first player employs an appropriate state feedback law $\mathbf{u}(\mathbf{x})$.

Assume that the state constraint, G_0 , is included into the family of sets

$$G_\lambda = \{\mathbf{x} \in R^n, g(\mathbf{x}) \leq \lambda\}, \quad (7)$$

where g is a suitable continuous function. The viability kernels, $Viab(G_\lambda)$, of the state constraints (7) can be represented as level sets of an appropriate function V :

$$Viab(G_\lambda) = \{\mathbf{x} \in R^n, V(\mathbf{x}) \leq \lambda\}. \quad (8)$$

The required function V can be found as a grid approximation of a limiting solution ($t \rightarrow -\infty$) of an appropriate Hamilton-Jacobi equation, which arises from the conflict control problem (1), see [2].

The numerical solution requires a discretization in space, with some step sizes $h := (h_1, \dots, h_n)$, and in time, with a step length $\delta > 0$. The grid scheme

$$\mathcal{V}_{\ell+1}^h = \max \left\{ \Pi[\mathcal{V}_\ell^h; \delta, h], g^h \right\}, \quad \mathcal{V}_0^h = g^h, \quad (9)$$

with g^h being the grid restriction of g , yields a sequence \mathcal{V}_ℓ^h , $\ell = 0, 1, \dots$, that monotonically and point-wise converges (see [3] and references [1] and [12] cited there) to a grid function \mathcal{V}^h , which is an approximation of the function V introduced in (8). The operator Π in (9) is defined as

$$\Pi[\phi; \delta, h](\mathbf{x}) = \phi(\mathbf{x}) + \delta \min_{\mathbf{u} \in P} \max_{\mathbf{v} \in Q} \sum_{i=1}^n (p_i^r f_i^+ + p_i^l f_i^-), \quad (10)$$

with

$$\begin{aligned} f_i^+ &= \max\{f_i, 0\}, \quad f_i^- = \min\{f_i, 0\}, \\ p_i^r &= \frac{\phi(x_1, \dots, x_i + h_i, \dots, x_n) - \phi(x_1, \dots, x_i, \dots, x_n)}{h_i}, \\ p_i^l &= \frac{\phi(x_1, \dots, x_i, \dots, x_n) - \phi(x_1, \dots, x_i - h_i, \dots, x_n)}{h_i}, \end{aligned} \quad (11)$$

where f_i is the i -th component of \mathbf{f} from (1). The implementation of such a method is feasible for up to seven dimensions, see [4] and [8], because of computer memory and performance requirements. For this implementation we have to use a highly parallelized solver tailored to a large computer grid such as the SuperMUC system at the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities.

The state feedback controls $\mathbf{u}(\mathbf{x})$ and $\mathbf{v}(\mathbf{x})$ of the first and second players, respectively, can be computed as solutions of the following minimax and maximin problems:

$$\mathbf{u}(\mathbf{x}) \rightarrow \min_{\mathbf{u} \in P} \max_{\mathbf{v} \in Q} \mathcal{L}^h[\mathcal{V}_\ell^h](\mathbf{x} + \tau \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v})), \quad (12)$$

$$\mathbf{v}(\mathbf{x}) \rightarrow \max_{\mathbf{v} \in Q} \min_{\mathbf{u} \in P} \mathcal{L}^h [\mathcal{V}_\ell^h] (\mathbf{x} + \tau \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v})), \quad (13)$$

where \mathcal{L}^h is an interpolation operator, and τ is a small extrapolation step length. It should be noted that $\tau > \delta$ for stability. In practice, $\tau \approx 10 \delta$.

3 Flight Simulator

The control scheme outlined in Section 2 is implemented on a flight simulator model at the Institute of Flight System Dynamics of the Technical University of Munich. The flight simulator model represents a modern transport aircraft with realistic dynamics, which includes rigid body motion, high fidelity aerodynamics, and engine industry data [11]. Furthermore, the simulator model has second-order transfer functions for the dynamics of elevators, rudders, ailerons, and other actuators. In total, the number of state variables is about one hundred. Additionally, inexact measurements of the states through noisy sensor models are considered for the feedback control using an Extended Kalman Filter (EKF).

Based on this mathematical flight simulator model, a model of forces for gravity, aerodynamics, and thrust, as well as the gains of reference models for attitude dynamics, and the rotational dynamics are derived for the reduced model. Moreover, in order to realize the viability kernel based control approach, the flight simulator model is extended by a control architecture for transforming the optimal attitude commands to actuator deflections of the flight simulator model.

4 Reduced Model

The main requirement for the reduced model is that a maximum number of seven states can be used for the dynamics in order to calculate the viability kernel. Considering the altitude, thrust, translation, and attitude of the aircraft, a reduced model with eight states can be derived. The corresponding state vector \mathbf{x} of the reduced model comprises the altitude h , the kinematic velocity V_K , the kinematic climb angle γ_K , the kinematic course angle χ_K , and the states of the attitude reference model (32), i.e. the kinematic angle of attack $\alpha_{K,RM}$, the kinematic sideslip angle $\beta_{K,RM}$, the kinematic bank angle $\mu_{K,RM}$, and the thrust level δ_T . Thus,

$$\mathbf{x} = [h, V_K, \gamma_K, \chi_K, \alpha_{K,RM}, \beta_{K,RM}, \mu_{K,RM}, \delta_T]'. \quad (14)$$

For the conflict control problem under consideration the first player \mathbf{u} utilizes the attitude and thrust commands:

$$\mathbf{u} = [\alpha_{K,c}, \beta_{K,c}, \mu_{K,c}, \delta_{T,c}]'. \quad (15)$$

The opposing player \mathbf{v} controls wind velocity components in the body fixed frame (B), i.e.:

$$\mathbf{v} = [(u_W)_B, (v_W)_B, (w_W)_B]'. \quad (16)$$

Observe that the wind velocities are directly used as the disturbance inputs to the reduced model and no states for the wind dynamics are introduced in the model. The reasoning behind this modeling choice is the following: First, the additional states for the wind model would further augment the state vector, rendering the computation of the viability kernel infeasible. Second, all states of the reduced model need to be measured for the viability kernel based control. Thus, if the wind states are included in the set of states of the reduced model an accurate measurement of the current wind velocity is required for the controller implementation in the flight simulator which for realistic applications is typically difficult to obtain. Obviously, this modeling choice is highly conservative as it allows the wind to change its velocity instantaneously. However, as shown for the illustrative example in Section 6, even if the viability kernel is computed for maximum (optimal) wind velocities considerably below the wind velocities which typically occur in aircraft operation, the viability based controller is able to withstand much higher (suboptimal) wind velocities in realistic simulations.

The dynamic equations for this model are detailed in the following. Simplifying assumptions for the derivation of this model are:

- Only gravity, aerodynamic, and engine forces in the cruise flight condition are considered.
- Constant gravity and mass are assumed.
- The effect of control surface deflections on the aerodynamic forces is neglected.
- The same power setting for left and right wing engine is used.
- A flat and non-rotating earth is supposed.
- The wind velocity is the only disturbance.

Moreover, recall that for the calculation of the viability kernel and the evaluation of the viability kernel based control, we require a model with at most seven states. In order to arrive at this number of states, we further reduce the model by setting the angle of side-slip command $\beta_{K,c}$ to zero which directly implies $\beta_{K,RM} = 0^\circ$. Thus, this state may be removed from the model and we arrive at the desired number of seven states.

The altitude propagation for the reduced model is obtained from the following relation:

$$\dot{h} = \sin(\gamma_K)V_K. \quad (17)$$

The translational dynamics, assuming flat and non-rotating earth, can be determined using the aircraft mass m and total force $(\mathbf{F}_T)_K = [X_T, Y_T, Z_T]'_K$ acting on the aircraft. In the kinematic frame (K), the corresponding equations read:

$$\begin{bmatrix} \dot{V}_K \\ \dot{\chi}_K \\ \dot{\gamma}_K \end{bmatrix} = \frac{1}{mV_K} \begin{bmatrix} V_K (X_T)_K \\ \frac{1}{\cos(\gamma_K)} (Y_T)_K \\ -(Z_T)_K \end{bmatrix}. \quad (18)$$

The total force $(\mathbf{F}_T)_K$ comprises the aerodynamic force $(\mathbf{F}_A)_K$, the propulsion force $(\mathbf{F}_P)_K$, and the gravitation force $(\mathbf{F}_G)_K$, i.e.:

$$(\mathbf{F}_T)_K = (\mathbf{F}_A)_K + (\mathbf{F}_P)_K + (\mathbf{F}_G)_K. \quad (19)$$

In (19), the aerodynamic force $(\mathbf{F}_A)_K$ is defined as:

$$(\mathbf{F}_A)_K = \mathbf{R}_{KA} (\mathbf{F}_A)_A, \quad (20)$$

where \mathbf{R}_{KA} is the transformation matrix between the aerodynamic frame (A) and the kinematic frame (K). The aerodynamic force model for $(\mathbf{F}_A)_A$ is derived from the mathematical model of the flight simulator and, besides the tabulated aerodynamic coefficients, depends on quantities such as the aerodynamic angle of attack α_A , the aerodynamic sideslip angle β_A , the air density ρ , the wing reference area S , the Mach number $Ma = V_A/a$ with the speed of sound $a = \sqrt{\kappa R T_{stat}}$ and the ratio of specific heat $\kappa = 1.4$, the specific gas constant $R = 287.05 \text{ J}/(\text{kg} \cdot \text{K})$, and the static temperature of air T_{stat} . The atmospheric quantities are determined based on the international standard atmosphere (ISA) model according to DIN ISO2533. The following equations are valid up to an altitude of 11000 m:

$$T_{stat} = T_s + \gamma_{Tr} H_G, \quad (21)$$

$$\rho = \rho_s \left(1 + \frac{\gamma_{Tr}}{T_s} H_G \right)^{\frac{1}{\eta_{Tr}-1}}, \quad (22)$$

$$p_{stat} = p_s \left(1 + \frac{\gamma_{Tr}}{T_s} H_G \right)^{\frac{\eta_{Tr}}{\eta_{Tr}-1}}. \quad (23)$$

In these equations, $T_s = 288.15 \text{ K}$ is the reference temperature, $\rho_s = 1.225 \text{ kg}/\text{m}^3$ the reference density, $p_s = 1.01325 \text{ N}/\text{m}^2$ the reference pressure, $\gamma_{Tr} = -6.510^{-3} \text{ K}/\text{m}$ the temperature gradient of the troposphere, $\eta_{Tr} = 1.235$ the exponent of the troposphere, and H_G the geopotential altitude calculated as

$$H_G = \frac{r_E h}{r_E + h}, \quad (24)$$

with the earth radius $r_E = 6356766 \text{ m}$. The aerodynamic quantities such as the aerodynamic velocity V_A , the aerodynamic angle of attack α_A , and the aerodynamic angle of sideslip β_A are derived from the vectorial wind relation denoted in the body-fixed frame (B)

$$(\mathbf{V}_A)_B = \mathbf{R}_{BK} (\mathbf{V}_K)_K - (\mathbf{V}_W)_B, \quad (25)$$

with the wind velocity vector $(\mathbf{V}_W)_B = \mathbf{v}$ (see equation (16)) and the transformation matrix \mathbf{R}_{BK} between the kinematic frame (K) and the body-fixed frame (B). From the components of $(\mathbf{V}_A)_B = [(u_A)_B, (v_A)_B, (w_A)_B]^T$, we can compute the aerodynamic quantities as follows:

$$V_A = \sqrt{(u_A)_B^2 + (v_A)_B^2 + (w_A)_B^2}, \quad (26)$$

$$\alpha_A = \arctan \left[\frac{(w_A)_B}{(u_A)_B} \right], \quad (27)$$

$$\beta_A = \arctan \left[\frac{(v_A)_B}{\sqrt{(u_A)_B^2 + (w_A)_B^2}} \right]. \quad (28)$$

Regarding the modeling of the forces, the propulsion force in body-fixed frame $(\mathbf{F}_P)_B$ derived from the mathematical model of the flight simulator provides with a transformation matrix $\mathbf{R}_{KB} = [\mathbf{R}_{BK}]'$ the following equation:

$$(\mathbf{F}_P)_K = \mathbf{R}_{KB} (\mathbf{F}_P)_B. \quad (29)$$

The propulsion force depends on the thrust δ_T , the aerodynamic angle of attack α_A , the aerodynamic sideslip angle β_A , the Mach number Ma , the static temperature T_{stat} and the static pressure p_{stat} of air. For the gravitational forces, we assume a constant gravitational acceleration vector $(\mathbf{g})_O$ and a constant mass m of the aircraft. Using the transformation matrix $\mathbf{R}_{KO} = [\mathbf{R}_{OK}]'$ we obtain:

$$(\mathbf{F}_G)_K = \mathbf{R}_{KO} (\mathbf{F}_G)_O = \mathbf{R}_{KO} m (\mathbf{g})_O. \quad (30)$$

Finally, it should be mentioned that the dynamics regarding the thrust state δ_T depends on the thrust command $\delta_{T,c}$, the thrust state itself, and the atmospheric quantities Ma , T_{stat} as well as p_{stat} . Note that the Mach number depends on the aerodynamic velocity which represents the disturbance in our conflict control problem. However, the dynamic model of the thrust can be written in the form

$$\dot{\delta}_T = f_{p,u}(\mathbf{x}, \delta_{T,c}) + f_{p,v}(\mathbf{x}, \mathbf{v}), \quad (31)$$

meaning that the right-hand side is additively separable regarding the thrust command and the disturbances (cf. (3)).

One of the key concepts for the reduced model is the use of a first-order reference model (32) for the description of attitude dynamics. This reference model essentially defines an interface between the reduced model and the closed-loop flight simulator model as the same reference model is used in the NDI controller of the flight simulator described in the following Section 5. For the attitude states in the reference model we use the kinematic angle of attack $\alpha_{K,RM}$, the kinematic sideslip angle $\beta_{K,RM}$, and the kinematic bank angle $\mu_{K,RM}$. It is important to mention that besides this choice of the reference model states also Euler angles Φ , Θ , and Ψ have been considered for the attitude loop. However, the performance of this modeling alternative showed considerably inferior results. A possible explanation may be a too conservative design regarding the performance of the reference model. Due to the rather high computational burden associated with the calculation of the viability kernel an extensive study for the determination of the exact root cause was outside the scope of this study. As such, the following reference dynamics are used for the attitude:

$$\begin{bmatrix} \dot{\alpha}_{K,RM} \\ \dot{\beta}_{K,RM} \\ \dot{\mu}_{K,RM} \end{bmatrix} = \begin{bmatrix} K_\alpha(h, V_K) (\alpha_{K,c} - \alpha_{K,RM}) \\ K_\beta(h, V_K) (\beta_{K,c} - \beta_{K,RM}) \\ K_\mu(h, V_K) (\mu_{K,c} - \mu_{K,RM}) \end{bmatrix} \quad (32)$$

The commands $\alpha_{K,c}$, $\beta_{K,c}$, and $\mu_{K,c}$ represent the controls of the reduced model and the gains $K_\alpha(h, V_K)$, $K_\beta(h, V_K)$, and $K_\mu(h, V_K)$ depend on the altitude h as well as the kinematic velocity V_K (see Fig. 1).

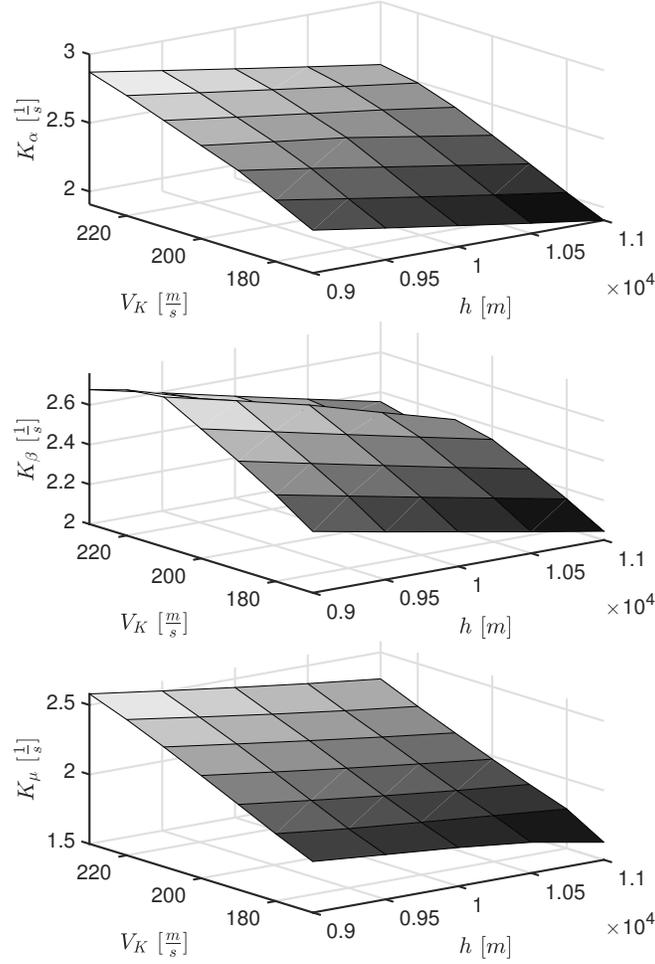


Fig. 1. Gain coefficients $K_\alpha(h, V_K)$, $K_\beta(h, V_K)$, and $K_\mu(h, V_K)$ for the attitude dynamics (32) and (34).

It is important to mention that it would be preferable to formulate the dependencies of these gains on aerodynamic quantities, i.e. the aerodynamic velocity V_A or similar quantities such as the dynamic pressure. Here, we deliberately do not follow this approach and only consider the kinematic velocity V_K for scheduling purposes in order to decouple the wind velocities (disturbance, second player) from the aircraft commands (controls, first player) in the reference

model. This allows us to separate the dynamics in the form of (3). As the same form holds as well for the thrust control (cf. (31)) the saddle point condition (2) is automatically fulfilled for the reduced model under consideration.

At this point it should be mentioned that for the translation of the optimal attitude commands calculated in the viability kernel based control to the corresponding actuator deflections the flight simulator model is extended by a NDI control architecture. This controller features first-order reference models for the attitude and rotation dynamics. It is particularly noteworthy that the reference model for the middle loop in the NDI controller shares the same structure and gains as the reference dynamics (32) but is modified by hedging signals and error controllers. The inner loop for the rotation dynamics shares a similar structure with its gains scheduled over the same quantities (V_K and h). Details regarding the controller implementation are provided in Section 5.

The gain coefficients presented in Fig. 1 for the attitude reference model (32) and the rotation reference model in the innermost loop of the NDI controller described in the following Section are determined for a trim grid over different altitudes and kinematic velocities. For this procedure a time-scale separation factor of ten between the attitude and the rotation loop is used. A detailed description regarding the calculation of these gains can be found in [9]. Intermediate values of the gain coefficients are obtained based on a multi-linear interpolation.

5 Control Architecture

The main task of the control architecture described in the following is to translate the optimal controls obtained from the viability kernel (cf. (12)) to surface deflection increments for the flight simulator model. For this purpose, two cascaded control loops with modified reference models of relative degree one (34) and (42), and a NDI in each loop are applied. The basic idea of NDI is to define an appropriate nonlinear feedback law that linearizes the plant. This can be achieved by computing the Lie-Derivative [13] of the output equations until the control input appears explicitly. Inversion of the resulting equation yields the nonlinear control feedback law. Using this approach, smooth reference trajectories can be followed by the plant. In the control concept presented here, two modified reference models of relative degree one (34) and (42) are used for the attitude dynamics (middle loop) and the rotation dynamics (inner loop). Further details regarding this control architecture can be found in [5] and [10]. The considered reference models are modified by hedging signals and PI error controllers. The attitude reference

$$\mathbf{r}_{\alpha\beta\mu} = \begin{bmatrix} \alpha_{K,c} \\ \beta_{K,c} \\ \mu_{K,c} \end{bmatrix}, \quad (33)$$

found from the reduced model using the feedback (12), is first propagated through the attitude equation

$$\boldsymbol{\nu}_{RM,\alpha\beta\mu} = \begin{bmatrix} \nu_{RM,\alpha} \\ \nu_{RM,\beta} \\ \nu_{RM,\mu} \end{bmatrix} = \begin{bmatrix} K_\alpha(h, V_K) (\alpha_{K,c} - \hat{\alpha}_{K,RM}) \\ K_\beta(h, V_K) (\beta_{K,c} - \hat{\beta}_{K,RM}) \\ K_\mu(h, V_K) (\mu_{K,c} - \hat{\mu}_{K,RM}) \end{bmatrix}. \quad (34)$$

with the states of the modified reference model $\hat{\alpha}_{K,RM}$, $\hat{\beta}_{K,RM}$, and $\hat{\mu}_{K,RM}$, as well as the same gains $K_\alpha(h, V_K)$, $K_\beta(h, V_K)$ and $K_\mu(h, V_K)$ used for the attitude dynamics (32). The state derivatives of the modified reference model are then obtained as:

$$\begin{bmatrix} \dot{\hat{\alpha}}_{K,RM} \\ \dot{\hat{\beta}}_{K,RM} \\ \dot{\hat{\mu}}_{K,RM} \end{bmatrix} = \boldsymbol{\nu}_{RM,\alpha\beta\mu} - \boldsymbol{\nu}_{h,\alpha\beta\mu}, \quad (35)$$

with the hedging signal $\boldsymbol{\nu}_{h,\alpha\beta\mu}$

$$\boldsymbol{\nu}_{h,\alpha\beta\mu} = \begin{bmatrix} \nu_{h,\alpha} \\ \nu_{h,\beta} \\ \nu_{h,\mu} \end{bmatrix} = \begin{bmatrix} \dot{\hat{\alpha}}_{K,RM,e} - \dot{\alpha}_K \\ \dot{\hat{\beta}}_{K,RM,e} - \dot{\beta}_K \\ \dot{\hat{\mu}}_{K,RM,e} - \dot{\mu}_K \end{bmatrix}, \quad (36)$$

defined as the expected reaction deficit between the pseudo commands $\dot{\hat{\alpha}}_{K,RM,e}$, $\dot{\hat{\beta}}_{K,RM,e}$, $\dot{\hat{\mu}}_{K,RM,e}$ and the expected system reactions $\dot{\alpha}_K$, $\dot{\beta}_K$ and $\dot{\mu}_K$. Note that $\dot{\alpha}_K$, $\dot{\beta}_K$, and $\dot{\mu}_K$ are the time derivatives of the corresponding states of the flight simulator model. The pseudo command of the attitude dynamics (middle loop) is defined as

$$\boldsymbol{\nu}_m = \begin{bmatrix} \dot{\hat{\alpha}}_{K,RM,e} \\ \dot{\hat{\beta}}_{K,RM,e} \\ \dot{\hat{\mu}}_{K,RM,e} \end{bmatrix} = \boldsymbol{\nu}_{RM,\alpha\beta\mu} + \boldsymbol{\nu}_{e,\alpha\beta\mu}, \quad (37)$$

where $\boldsymbol{\nu}_{e,\alpha\beta\mu} = [\nu_{e,\alpha}, \nu_{e,\beta}, \nu_{e,\mu}]'$ is obtained from the error controller

$$\boldsymbol{\nu}_{e,\alpha\beta\mu} = \begin{bmatrix} K_{e,\alpha}^P (\hat{\alpha}_{K,RM} - \alpha_K) + K_{e,\alpha}^I \int (\hat{\alpha}_{K,RM} - \alpha_K) dt \\ K_{e,\beta}^P (\hat{\beta}_{K,RM} - \beta_K) + K_{e,\beta}^I \int (\hat{\beta}_{K,RM} - \beta_K) dt \\ K_{e,\mu}^P (\hat{\mu}_{K,RM} - \mu_K) + K_{e,\mu}^I \int (\hat{\mu}_{K,RM} - \mu_K) dt \end{bmatrix} \quad (38)$$

consisting of proportional ($K_{e,\alpha}^P$, $K_{e,\beta}^P$, $K_{e,\mu}^P$) and integral ($K_{e,\alpha}^I$, $K_{e,\beta}^I$, $K_{e,\mu}^I$) parts.

In this way, equations (33)–(38) describe the modified reference model for the attitude dynamics (middle loop). The described reference model structure is illustrated in Fig. 2 for the kinematic angle of attack.

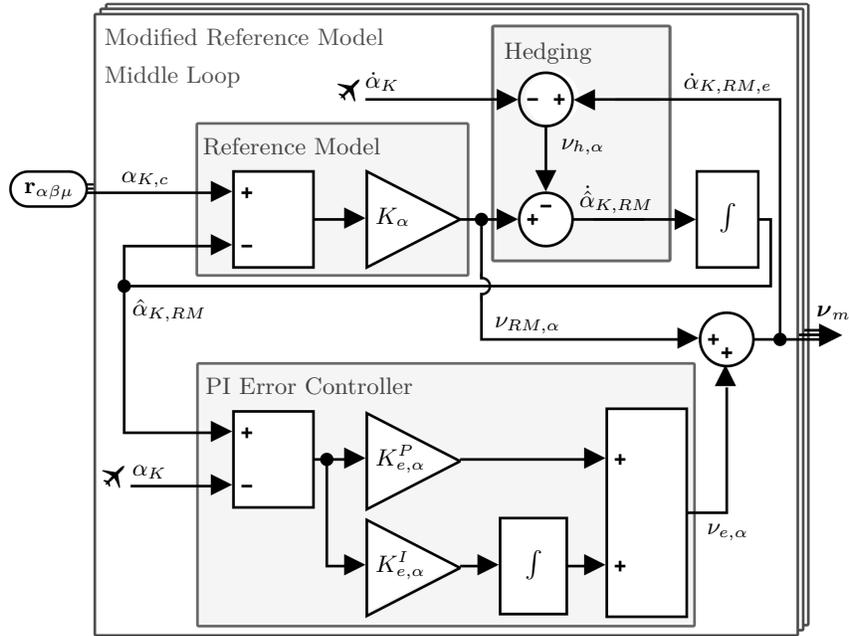


Fig. 2. Structure of the modified reference model for the middle loop.

Continuing to the inner loop, the reference command of the rotational dynamics is computed from

$$\mathbf{r}_{pqr} = \begin{bmatrix} p_c \\ q_c \\ r_c \end{bmatrix}_B = \mathbf{R}_{BK} [(\boldsymbol{\omega}^{OK})_K + (\boldsymbol{\omega}_c^{KB})_K], \quad (39)$$

with explicit inversions

$$(\boldsymbol{\omega}^{OK})_K = \begin{bmatrix} -\dot{\chi}_K \sin(\gamma_K) \\ \dot{\gamma}_K \\ \dot{\chi}_K \cos(\gamma_K) \end{bmatrix}_K \quad (40)$$

and

$$(\boldsymbol{\omega}_c^{KB})_K = \begin{bmatrix} \dot{\mu}_{K,RM,e} + \dot{\alpha}_{K,RM,e} \sin(\beta_K) \\ \dot{\alpha}_{K,RM,e} \cos(\beta_K) \cos(\mu_K) + \dot{\beta}_{K,RM,e} \sin(\mu_K) \\ \dot{\alpha}_{K,RM,e} \cos(\beta_K) \sin(\mu_K) - \dot{\beta}_{K,RM,e} \cos(\mu_K) \end{bmatrix}_K. \quad (41)$$

From the commands p_c , q_c , and r_c using

$$\boldsymbol{\nu}_{RM,pqr} = \begin{bmatrix} \nu_{RM,p} \\ \nu_{RM,q} \\ \nu_{RM,r} \end{bmatrix} = \begin{bmatrix} K_p(h, V_K) (p_c - \hat{p}_{RM}) \\ K_q(h, V_K) (q_c - \hat{q}_{RM}) \\ K_r(h, V_K) (r_c - \hat{r}_{RM}) \end{bmatrix}, \quad (42)$$

with the states of the modified reference model \hat{p}_{RM} , \hat{q}_{RM} , and \hat{r}_{RM} , the gain coefficients $K_p(h, V_K)$, $K_q(h, V_K)$, and $K_r(h, V_K)$ depicted in Fig. 3 and the

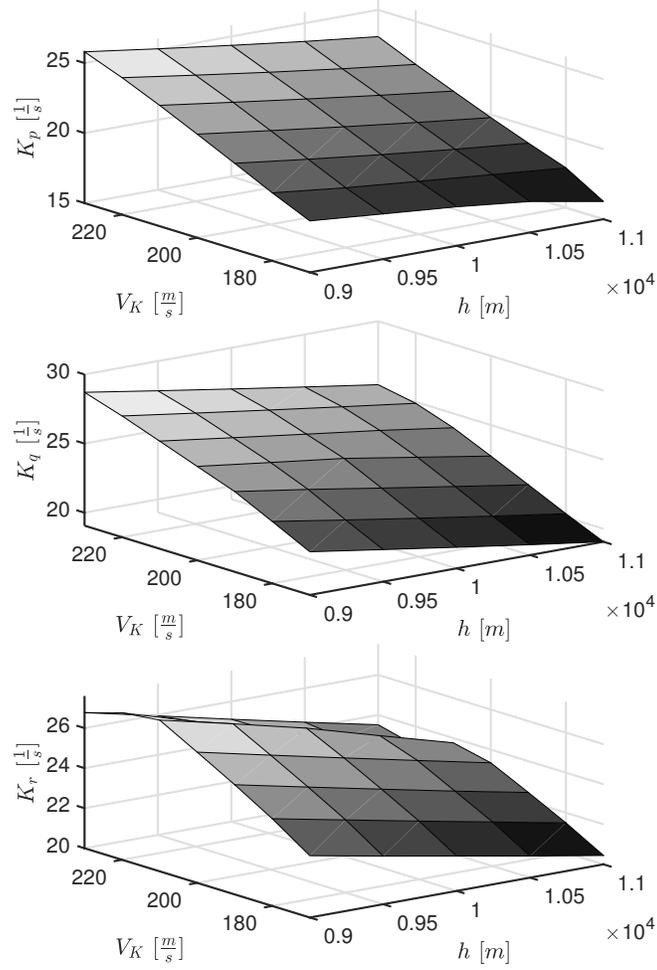


Fig. 3. Gain coefficients $K_p(h, V_K)$, $K_q(h, V_K)$, and $K_r(h, V_K)$ for the modified reference model of the rotational dynamics (42).

hedging signal $\nu_{h,pqr}$, we obtain the modified rotation reference model:

$$\begin{bmatrix} \dot{\hat{p}}_{RM} \\ \dot{\hat{q}}_{RM} \\ \dot{\hat{r}}_{RM} \end{bmatrix} = \nu_{RM,pqr} - \nu_{h,pqr}. \quad (43)$$

Herein, the hedging signal, $\boldsymbol{\nu}_{h,pqr} = [\nu_{h,p}, \nu_{h,q}, \nu_{h,r}]'$, is defined as the expected reaction deficit between the pseudo commands $\dot{p}_{RM,e}$, $\dot{q}_{RM,e}$, $\dot{r}_{RM,e}$ and the expected reactions \dot{p} , \dot{q} and \dot{r} of the system, i.e.

$$\boldsymbol{\nu}_{h,pqr} = \begin{bmatrix} \dot{p}_{RM,e} - \dot{p} \\ \dot{q}_{RM,e} - \dot{q} \\ \dot{r}_{RM,e} - \dot{r} \end{bmatrix} \quad (44)$$

Note that, the hedging signal in the rotation dynamics accounts for the actuator dynamics, which are not included in the inversion. The pseudo command for the rotation dynamics is defined as

$$\boldsymbol{\nu}_i = \begin{bmatrix} \dot{p}_{RM,e} \\ \dot{q}_{RM,e} \\ \dot{r}_{RM,e} \end{bmatrix} = \boldsymbol{\nu}_{RM,pqr} + \boldsymbol{\nu}_{e,pqr}, \quad (45)$$

where the error controller signal for the rotation dynamics $\boldsymbol{\nu}_{e,pqr} = [\nu_{e,p}, \nu_{e,q}, \nu_{e,r}]'$ is defined as

$$\boldsymbol{\nu}_{e,pqr} = \begin{bmatrix} K_{e,p}^P (\hat{p}_{RM} - p) + K_{e,p}^I \int (\hat{p}_{RM} - p) dt \\ K_{e,q}^P (\hat{q}_{RM} - q) + K_{e,q}^I \int (\hat{q}_{RM} - q) dt \\ K_{e,r}^P (\hat{r}_{RM} - r) + K_{e,r}^I \int (\hat{r}_{RM} - r) dt \end{bmatrix} \quad (46)$$

consisting of proportional ($K_{e,p}^P$, $K_{e,q}^P$, $K_{e,r}^P$) and integral ($K_{e,p}^I$, $K_{e,q}^I$, $K_{e,r}^I$) parts as for the middle loop. Note that, the modified reference model of the rotation loop share the same structure as the modified reference model of the attitude loop. This structure is visualized in Fig. 4 for the pitch rate.

The actuator command increment vector

$$\delta \mathbf{u}_r = \begin{bmatrix} \delta \xi \\ \delta \eta \\ \delta \zeta \end{bmatrix} \quad (47)$$

containing the aileron increment $\delta \xi$, elevator increment $\delta \eta$, and rudder increment $\delta \zeta$ is obtained from the inversion of the rotational dynamics. For this purpose, the control effectiveness matrix \mathbf{B} collecting the derivatives of the total moments $(\mathbf{M}_T)_B = [L, M, N]'$ with respect to the actuator positions ξ , η and ζ is defined as:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial L}{\partial \xi} & \frac{\partial L}{\partial \eta} & \frac{\partial L}{\partial \zeta} \\ \frac{\partial M}{\partial \xi} & \frac{\partial M}{\partial \eta} & \frac{\partial M}{\partial \zeta} \\ \frac{\partial N}{\partial \xi} & \frac{\partial N}{\partial \eta} & \frac{\partial N}{\partial \zeta} \end{bmatrix}. \quad (48)$$

The desired moments $(\mathbf{M}_T)_B$ are computed from the angular body rates $(\boldsymbol{\omega}_K^{OB})_B$, the inertia tensor $(\mathbf{I})_{BB}$, and the pseudo command of the rotation dynamics (inner loop) $\boldsymbol{\nu}_i$ from (45):

$$(\mathbf{M}_T)_B = (\boldsymbol{\omega}_K^{OB})_B \times (\mathbf{I})_{BB} (\boldsymbol{\omega}_K^{OB})_B + (\mathbf{I})_{BB} \boldsymbol{\nu}_i. \quad (49)$$

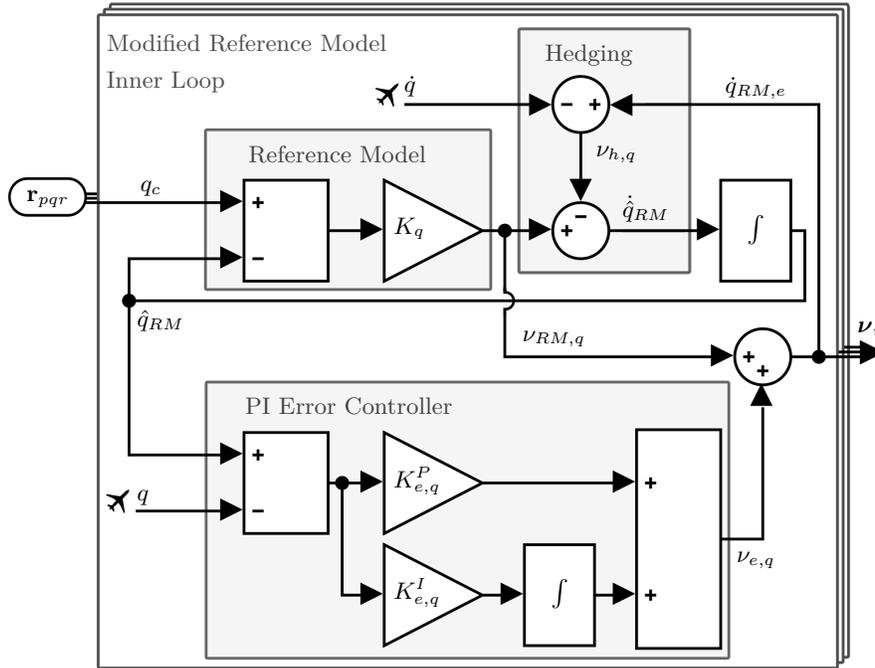


Fig. 4. Structure of the modified reference model for the inner loop.

Subtracting the estimated moment $(\tilde{\mathbf{M}}_T)_B$ from the desired moment (49) corresponds to the product of the control effectiveness (48) and the actuator command increments (47):

$$\mathbf{B}\delta\mathbf{u}_r = (\mathbf{M}_T)_B - (\tilde{\mathbf{M}}_T)_B. \quad (50)$$

Finally, we get the actuator command increments $\delta\mathbf{u}_r$ by solving the control allocation problem (50). Thus, the actuator controls \mathbf{u}_r result from adding the actuator increments to the current actuator surface deflections $\mathbf{x}_r = [\xi, \eta, \zeta]'$. The actuator controls and the thrust control $\delta T_{,c}$ yield the controls \mathbf{u}_{fs} for the flight simulator model. The information flow in the control architecture from the viability kernel based control to the control surface and thrust command are illustrated in Fig. 5.

Moreover, step responses for the same trim conditions used in the simulation in Section 6 are shown in Fig. 6. For the step responses presented here perfect measurements of states are assumed.

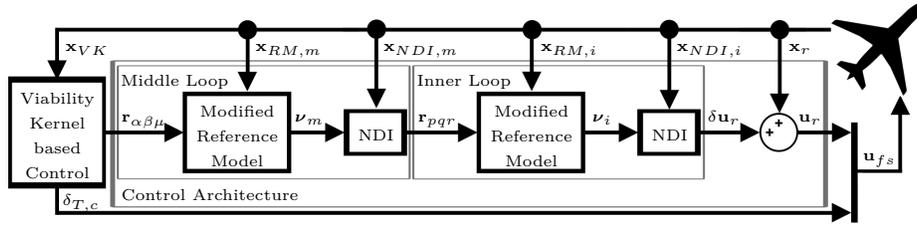


Fig. 5. Schematic overview of the control implementation in the flight simulator model.

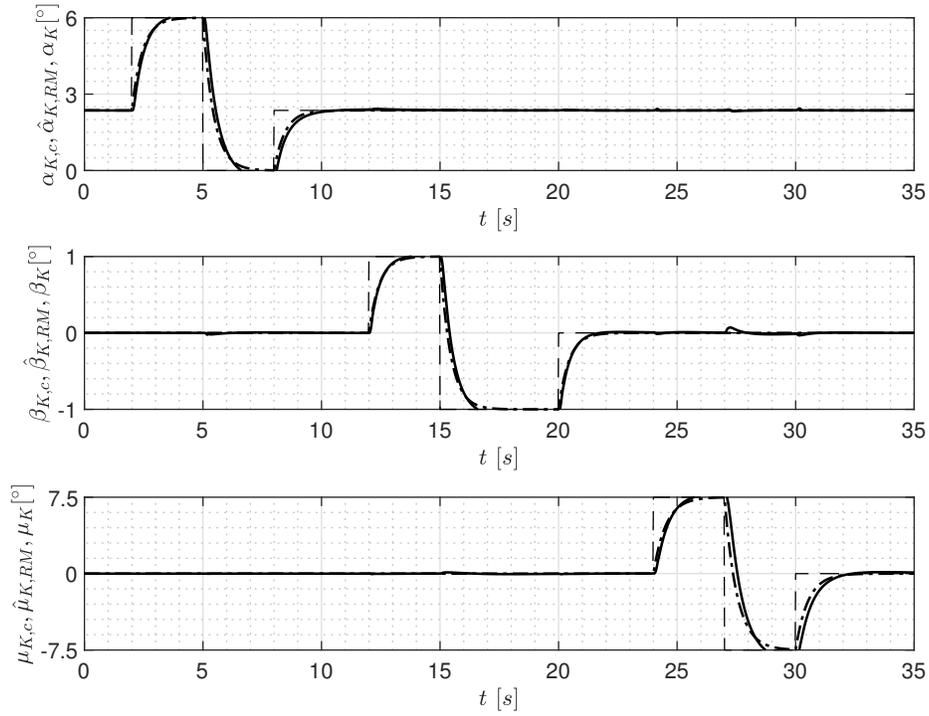


Fig. 6. Step response of the modified reference model $\hat{\alpha}_{K,RM}$, $\hat{\beta}_{K,RM}$, and $\hat{\mu}_{K,RM}$ without hedging signal, (dash-dotted line) and the following behavior of the flight simulator states α_K , β_K , and μ_K (solid line) in response to the step controls $\alpha_{K,c}$, $\beta_{K,c}$, and $\mu_{K,c}$ (dashed line).

Figure 6 suggests a good following behavior of the modified reference models for the attitude dynamics by the flight simulator model, even without using the hedging signal. Note that using the hedging signal, the step responses of the modified reference models coincides with the behavior of the flight simulator. This also holds for larger step responses as can be seen in Fig. 7.

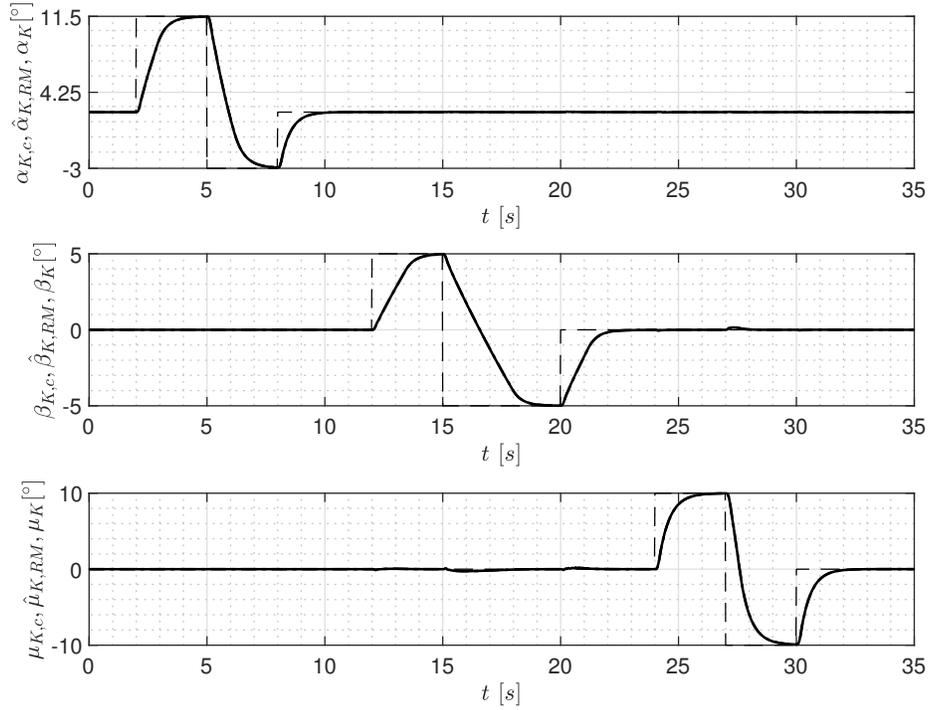


Fig. 7. Larger step response of the modified reference model $\hat{\alpha}_{K,RM}$, $\hat{\beta}_{K,RM}$, and $\hat{\mu}_{K,RM}$ with hedging signal, (dash-dotted line) and the following behavior of the flight simulator states α_K , β_K , and μ_K (solid line) in response to the step controls $\alpha_{K,c}$, $\beta_{K,c}$, and $\mu_{K,c}$ (dashed line).

6 Calculation and Simulation Results

In this section, the calculation of the viability kernel, simulation aspects, and numerical results of the flight simulations in the cruise flight condition are presented.

6.1 Calculation of the Viability Kernel

The calculation was performed on a grid with $9 \cdot 10^7$ nodes according to Section 2. A resolution of 30 nodes both for the altitude and kinematic velocity, as well as ten nodes for the other states were used.

Recall, that due to the computer resource limitation, the kinematic sideslip angle is neglected (set to zero) which reduces the dimension of the reduced model to seven states. The state constraints are chosen according to Table 1 and the bounds imposed on control and disturbance variables are presented in Tables 2 and 3.

Table 1. State constraints for the calculation of the viability kernel.

i	x_i	$x_{i,Viab}^{lb}$	$x_{i,Viab}^{ub}$	Unit
1	h	9955	10045	m
2	V_K	170	230	$\frac{m}{s}$
3	γ_K	-5.2	5.2	$^\circ$
4	χ_K	-1.5	1.5	$^\circ$
5	$\alpha_{K,RM}$	-3.9	14.9	$^\circ$
6	$\beta_{K,RM}$	0	0	$^\circ$
7	$\mu_{K,RM}$	-13	13	$^\circ$
8	δ_T	70	110	$\%$

Table 2. Control bounds for the calculation of the viability kernel.

i	u_i	$u_{i,Viab}^{lb}$	$u_{i,Viab}^{ub}$	Unit
1	$\alpha_{K,c}$	-3	11.5	$^\circ$
2	$\beta_{K,c}$	0	0	$^\circ$
3	$\mu_{K,c}$	-10	10	$^\circ$
4	$\delta_{T,c}$	80	100	$\%$

Table 3. Disturbance bounds for the calculation of the viability kernel.

i	v_i	$v_{i,Viab}^{lb}$	$v_{i,Viab}^{ub}$	Unit
1	$(u_W)_B$	-3	3	$\frac{m}{s}$
2	$(v_W)_B$	-3	3	$\frac{m}{s}$
3	$(w_W)_B$	-3	3	$\frac{m}{s}$

Note that, the control bounds for the calculation of viability kernel (Table 2) are not the same as for the simulation controls (compare Table 4) because the viability kernel disappears in the case of the same bounds. The calculation of the viability kernel was performed with a step length $\delta = 0.01$ s until the functions produced by the formula (9) converge to a precision of 10^{-6} .

A visualization of the seven-dimensional viability kernel is not useful for checking whether some point belongs to it. Instead of that the limiting value function produced by (9) can be used. If the value function is non-positive at a point, this point lies in the viability kernel, and vice versa.

6.2 Simulation Aspects

For the initial values of the simulation, we use a trim condition of the flight simulator which is obtained through the optimal values for the altitude and the kinematic velocity from the viability kernel. Depending on these values we determine the relative angle of attack, elevator deflection, and thrust level by trimming the model of the flight simulator. Thus, it is ensured that the simulation starts from a trim condition inside the viability kernel.

It is noteworthy that the simulation shows a sensitive behavior with respect to the control variables and the length of the extrapolation step τ (see (12)). If the controls are too large, the reduced model does not accurately reflect the aircraft dynamics. If the controls are too small, the disturbance cannot be sufficiently compensated. The low resolution of the control can be partly compensated by the extrapolation step length τ . If the time step is too small, the faster dynamics are weighted more, which may lead to an unfavorable control if the model deviates. If the time step is too large, the predictive shift can aim near to or beyond the boundary of the viability kernel leading to a higher cost function value. Thus, the bounds on the control variables and the predictive simulation time step τ (extrapolation step) need to be selected carefully for the application under consideration.

In our simulation, we evaluate the min max-operator in (12) using control values $\mathbf{u} = [u_1, \dots, u_4]'$ according to Table 4 and disturbance values according to Table 3. Note that \underline{u}_i , $i = 1, \dots, 4$ represent the lower thresholds of the control, and \bar{u}_i , $i = 1, \dots, 4$ the upper thresholds. Moreover \tilde{u}_i , $i = 1, \dots, 4$ are the current values of the corresponding flight simulator model states, i.e. \tilde{u}_1 corresponds to the kinematic angle of attack α_K , \tilde{u}_2 to the kinematic angle of sideslip β_K , \tilde{u}_3 to the kinematic bank angle μ_K , and \tilde{u}_4 to the thrust state δ_T . During the whole simulation, the control variable of the kinematic sideslip angle $\beta_{K,c}$ is kept at zero.

It is noteworthy that the use of current states as control variables (see Table 4) shows a positive influence on the simulation results. Computational experience suggests that, on the one side, this strategy is less likely to lead to fast control chattering as in many cases the current state is preferred over large corrective actions. On the other side, the low number of controls to be evaluated (min, max, current state), compared to an otherwise potentially fine resolution of the control, has a positive effect on the simulation time.

Table 4. Control values for simulations.

i	u_i	\underline{u}_i	\tilde{u}_i	\bar{u}_i	Unit
1	$\alpha_{K,c}$	0	α_K	6	°
2	$\beta_{K,c}$	0	0	0	°
3	$\mu_{K,c}$	-7.5	μ_K	7.5	°
4	$\delta_{T,c}$	80	δ_T	100	%

In our experiments, we achieved good results using the extrapolation step $\tau = 0.2$ s in (12). In addition to the case of optimal wind and the case without wind, we considered a suboptimal wind generated by the Dryden turbulence model [7]. The simulation was performed for 100 s using the Euler forward method with the step size $\delta_s = 0.0001$ s. The optimal control variables are determined with the step size $\delta_c = 0.02$ s. Moreover, noisy measurement from the sensor models are assumed and the measured quantities for the control architecture are estimated using an EKF implementation.

6.3 Simulation Results

The flight simulator model was initialized in a cruise flight condition with the kinematic angle of attack $\alpha_{K,trim} = 2.36^\circ$, elevator deflection $\eta_{trim} = -1.13^\circ$, and thrust level $\delta_{T,trim} = 84,36\%$ at the kinematic velocity $V_K = 186$ m/s and altitude of $h = 9980$ m.

Figures 10–13 show eight states of the flight simulator model (the same as in the reduced model) for the optimal, suboptimal, and without wind disturbances, see Fig. 8. Recall that only seven states of the viability kernel are used to calculate optimal controls and disturbances. The corresponding control variables are depicted in Fig. 14. Figure 9 illustrates the value function along the trajectories. Here it should be mentioned that the negative values indicate that the trajectories remain within the viability kernel during the whole simulation time. It is seen that the trajectories corresponding to lower values of the value function are obtained for the case without and suboptimal (Dryden) wind.

In addition, simulation results for the Dryden suboptimal wind with an amplitude of 12 m/s are shown in Fig. 15. For this case, Figure 16 shows the value function along the trajectory. Since the value function remains negative, the whole trajectory lies inside the viability kernel in this case. It should be noted that 12 m/s is four times larger than the wind disturbance bounds used for the construction of the viability kernel.

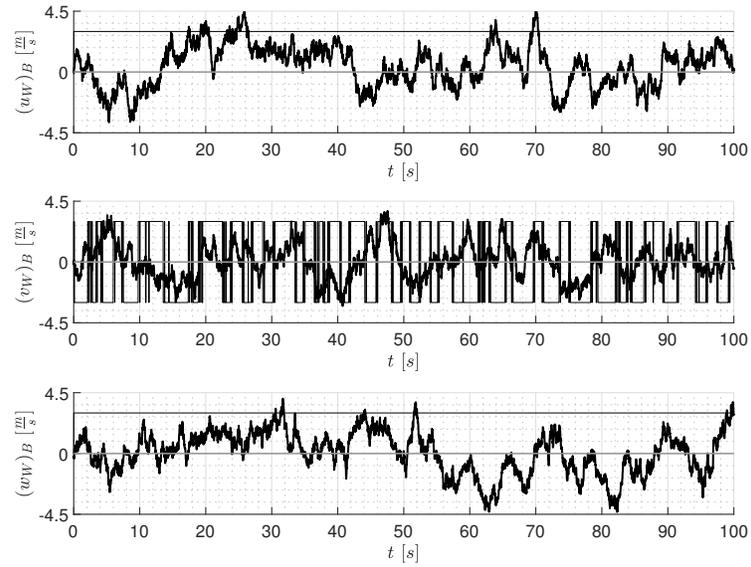


Fig. 8. Wind disturbance $(u_W)_B$, $(v_W)_B$ and $(w_W)_B$ for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

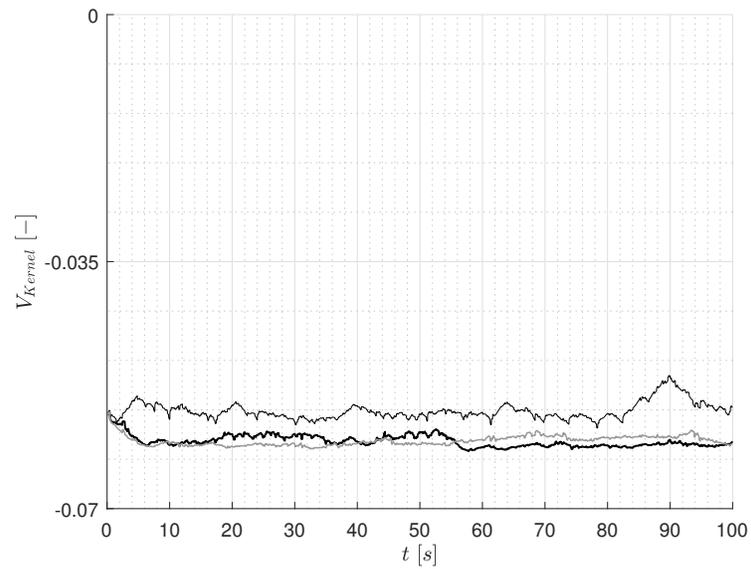


Fig. 9. Value function for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

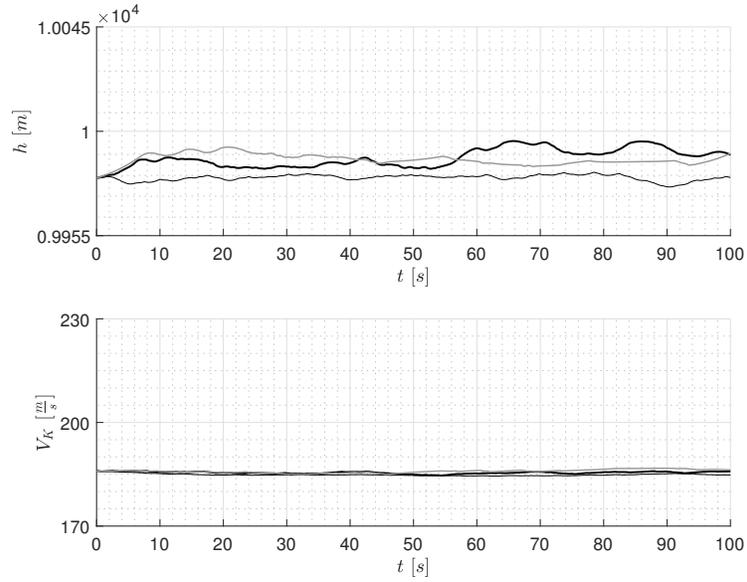


Fig. 10. Trajectory for the flight simulator states h and V_K for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

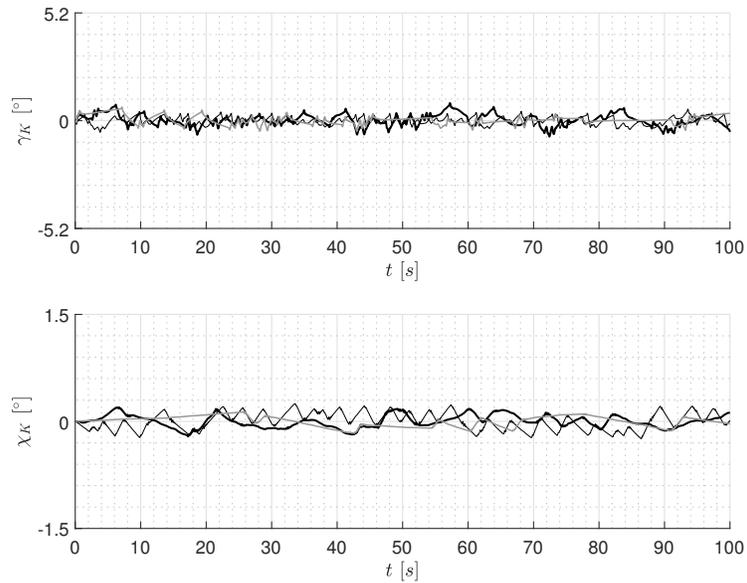


Fig. 11. Trajectory for the flight simulator states γ_K and χ_K for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

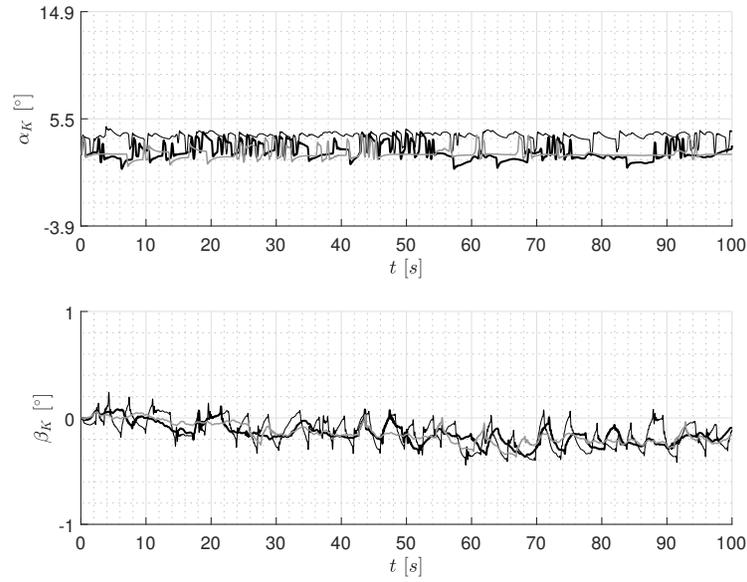


Fig. 12. Trajectory for the flight simulator states α_K and β_K for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

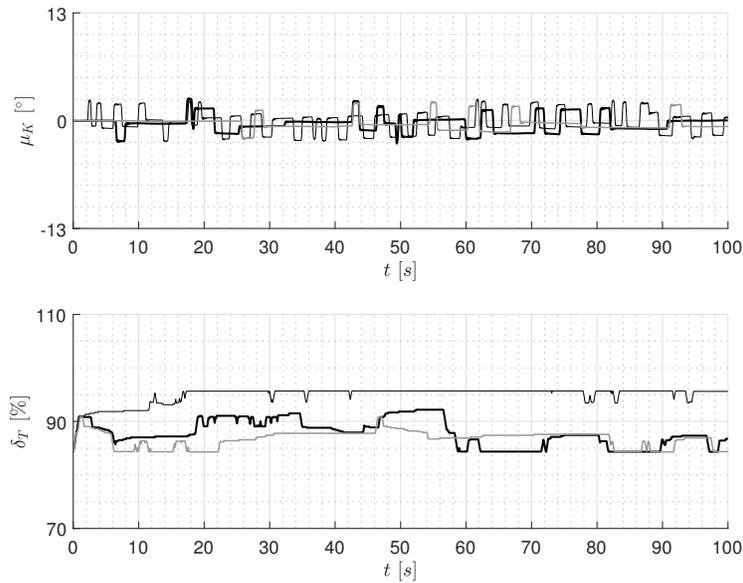


Fig. 13. Trajectory for the flight simulator states μ_K and δ_T for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

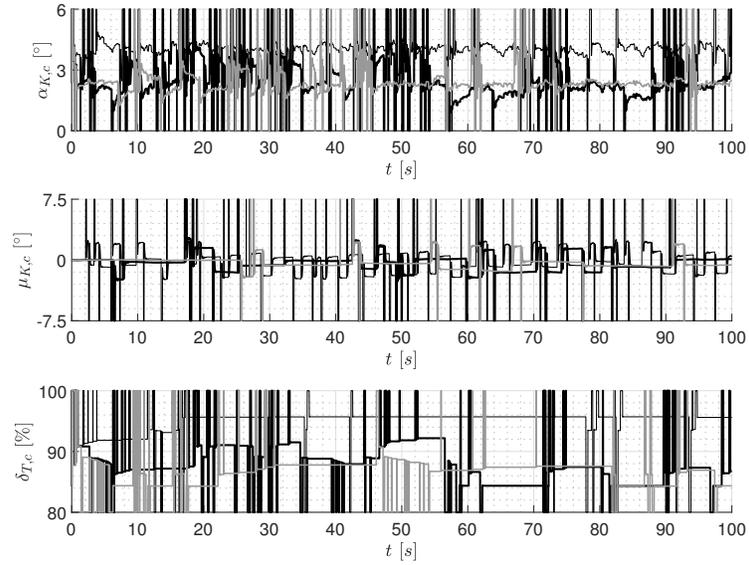


Fig. 14. Flight simulator controls $\alpha_{K,c}$, $\mu_{K,c}$ and $\delta_{T,c}$ for optimal (thin black line), suboptimal (thick black line), and no wind (grey line) disturbance.

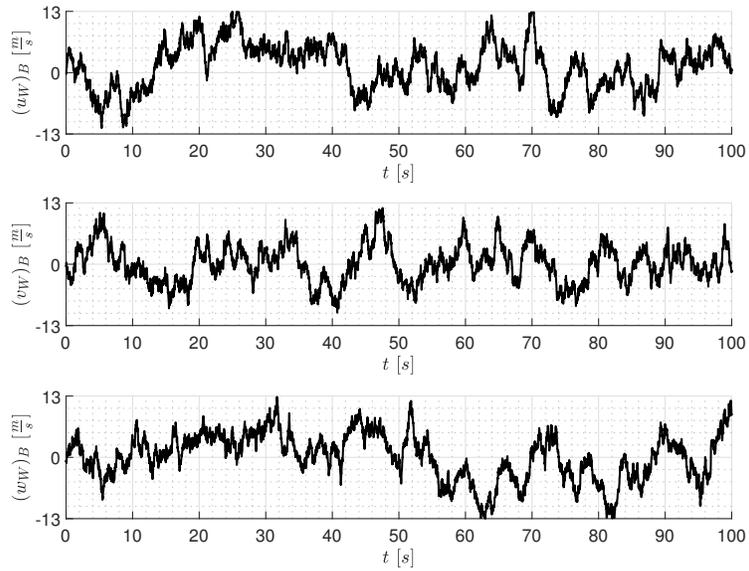


Fig. 15. Dryden wind disturbances $(u_W)_B$, $(v_W)_B$ and $(w_W)_B$ for suboptimal wind with an amplitude of 12 m/s.

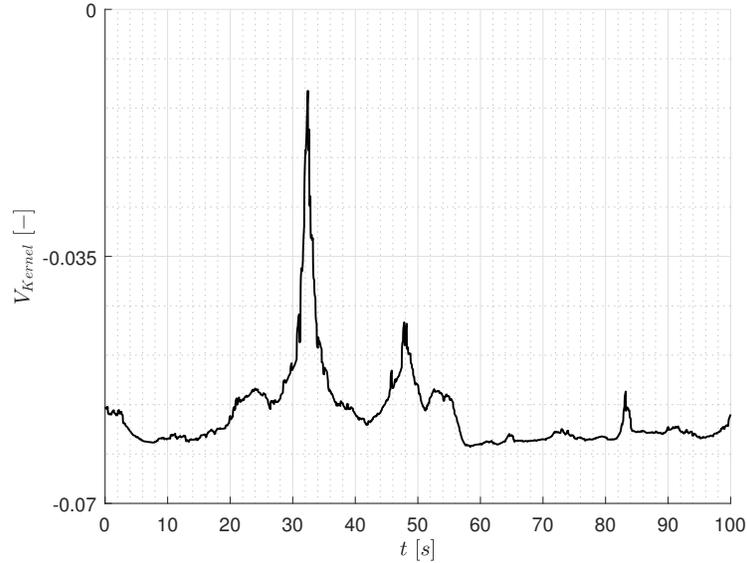


Fig. 16. Value function for suboptimal wind disturbance of 12 m/s.

7 Conclusions and Future Perspective

The current investigation shows that the model of a flight simulator with about a hundred state variables can be controlled by applying viability theory on a reduced problem having few (seven) states. The reduced model enables the computation of the viability kernels, which allows designing a feedback control for keeping the state vector of the reduced model inside the viability kernel. This feedback control, using a control architecture based on NDI, can be applied to the flight simulation model in such a way that the seven state variables (the same as in the reduced model) remain in the viability kernel.

It should be stressed that the reduced model has to reflect the flight simulator dynamics as well as possible. If the rates produced by the reduced model (32) are too high, the flight simulator dynamics can not follow it. If the rates are too low, the viability kernel can not exist. As such, the design of an appropriate reduced model is a key ingredient in the control approach presented in this paper.

For future research the idea to include unmodeled parts (such as the control surface deflections in the aerodynamic forces) in the reduced model as disturbances seems appealing.

Acknowledgment

This work was supported by the DFG grant HO4190/8-2 and TU427/2-2. Computer resources for this project have been provided by the Gauss Centre for Supercomputing/Leibniz Supercomputing Centre under the grant: pr74lu.

References

1. Aubin, J.P.: Viability theory. Systems and control, Birkhäuser, Boston (1991)
2. Botkin, N.D., Hoffmann, K.H., Mayer, N., Turova, V.L.: Approximation schemes for solving disturbed control problems with non-terminal time and state constraints. *Analysis* **31**(4), 355 – 379 (2011). <https://doi.org/10.1524/anly.2011.1122>
3. Botkin, N.D., Turova, V.L., Diepolder, J., Bittner, M., Holzapfel, F.: Aircraft control during cruise flight in windshear conditions: viability approach. *Dynamic Games and Applications* **7**(4), 594–608 (2017). <https://doi.org/10.1007/s13235-017-0215-9>
4. Botkin, N.D., Turova, V.L., Diepolder, J., Holzapfel, F.: Computation of viability kernels on grid computers for aircraft control in windshear. *Advances in Science, Technology and Engineering Systems Journal* **3**, 502–510 (2018). <https://doi.org/10.25046/aj030161>
5. Bugajski, D.J., Enns, D.F., Elgersma, M.R.: A dynamic inversion based control law with application to the high angle-of-attack research vehicle. In: *Guidance, Navigation and Control Conference. Guidance, Navigation, and Control and Co-located Conferences*, pp. 826–839. American Institute of Aeronautics and Astronautics (1990). <https://doi.org/10.2514/6.1990-3407>
6. Cardaliaguet, P.: A differential game with two players and one target. *SIAM Journal on Control and Optimization* **34**(4), 1441–1460 (1996). <https://doi.org/10.1137/S036301299427223X>
7. Chalk, C.R., Neal, T.P., Harris, T.M., Pritchard, F.E., Woodcock, R.J.: Background information and user guide for Mil-F-8785B (ASG), 'Military specification-flying qualities of piloted airplanes', AFFDL-TR, vol. 70-72. Air Force Flight Dynamics Laboratory Air Force Systems Command, Wright-Patterson AFB, Ohio (1969)
8. Diepolder, J., Piprek, P., Botkin, N.D., Turova, V.L., Holzapfel, F.: A robust aircraft control approach in the presence of wind using viability theory. In: *Australian and New Zealand Control Conference (ANZCC)*, pp. 155–160 (2017). <https://doi.org/10.1109/ANZCC.2017.8298503>
9. Gerdt, A.: *Integration und Testen einer robusten Regelungsstruktur an einem Forschungsflugsimulator*. Masterarbeit, Technische Universität München, München (2018)
10. Holzapfel, F.: *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*. Dissertation, Technische Universität München, München (2004)
11. Research Flight Simulator, <https://www.fsd.mw.tum.de/infrastructure/simulators>. Last accessed 29 Sep 2020
12. Krasovskii, N.N., Subbotin, A.I.: *Game-theoretical control problems*. Springer, New York (1988)
13. Slotine, J.J.E., Li, W.: *Applied nonlinear control*. Prentice Education Taiwan Ltd, Taipei, international ed. (2005)